Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

І. Я. Спекторський, В. М. Статкевич

Формальні мови та автомати

Затверджено Вченою Радою КПІ ім. Ігоря Сікорського
як підручник для студентів,
які навчаються за спеціальністю 124 «Системний аналіз»

Київ
КПІ ім. Ігоря Сікорського
2019

Рецензенти: *Калюжний О. О.* д-р фіз.-мат. наук, пров. наук. співроб.,
Інститут математики НАН України
Покутний О. О. д-р фіз.-мат. наук, ст. наук. співроб.,
Інститут математики НАН України

Відповідальний
редактор: *Подколзін Г. Б., канд. фіз.-мат. наук, доц.*

Гриф надано Вченою Радою КПІ ім. Ігоря Сікорського (протокол № 7 від 24.06.2019 р.)

Навчальне видання

Спекторський Ігор Якович, канд. фіз.-мат. наук, доц.
Статкевич Віталій Михайлович, канд. фіз.-мат. наук

Формальні мови та автомати

Формальні мови та автомати : підруч. для студ. спец. 124 «Системний аналіз» / І. Я. Спекторський, В. М. Статкевич ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2019. – 167 с.

Наведено основні теоретичні відомості з традиційних для цієї дисципліни розділів: формальні мови та граматики, машина Тьюрінга, скінченні автомати та автомати з магазинною пам'яттю, лінійно обмежені машини Тьюрінга. Означення та конструкції надано у формальному вигляді, для теорем, лем і тверджень надано повні доведення; виняток становлять декілька простих тверджень, доведення яких запропоновано як вправи, а також декілька тверджень, повне доведення яких, на думку авторів, суттєво виходить за межі цього підручника. Теоретичний матеріал ілюстровано численними прикладами. Для самостійного виконання надано багато вправ, до яких за потреби надано вказівки.

Для студентів ІПСА КПІ ім. Ігоря Сікорського, які навчаються за спеціальністю 124 «Системний аналіз». Підручник також може бути корисним для студентів математичних факультетів вищих навчальних закладів, зокрема, для студентів ІПСА КПІ ім. Ігоря Сікорського, які навчаються за спеціальністю 122 «Комп'ютерні науки».

Зміст

Вступ

Теорія формальних мов та автоматів активно розвивається з 1950-х років. Вихідною точкою у становленні теорії стали роботи американського вченого, професора Н. Хомського (див. [1–3]). Головним завданням теорії було розроблення математичного апарату для опису та аналізу природних і штучних мов. У наш час теорія формальних мов та автоматів надає потужні засоби математичного моделювання, які активно застосовуються, зокрема, в синтаксичному аналізі, перекладі та у розробленні компіляторів (див., наприклад, [4]).

У цьому підручнику головний акцент зроблено на двох підходах до опису формальної мови – за допомогою породжувальних граматик та через абстрактні автомати, що допускають (розпізнають) певну формальну мову. Матеріал у підручнику викладено згідно з ієрархією формальних мов, запропонованою Н. Хомським, яка розподіляє породжувальні граматики на чотири типи (рівні ієрархії). Кожний клас формальних мов, породжених граматиками певного типу, згідно з ієрархією Хомського відповідає певному класу абстрактних автоматів (табл. В.1).

Таблиця В.1

Тип граматики	Клас формальних мов	Клас абстрактних автоматів
0	Напіврозв'язні (рекурсивно-перераховні)	Машини Тьюрінга
1	Контекстно-залежні	Лінійно обмежені машини Тьюрінга
2	Контекстно-вільні	Магазинні (стекові) автомати
3	Регулярні	Скінченні автомати

У першому розділі підручника введено поняття формальної мови та формальної (породжувальної) граматики, визначено основні операції над формальними мовами та описано ієрархію Хомського.

У другому розділі розглянуто напіврозв'язні (рекурсивно-перераховні) формальні мови, які породжуються будь-якими формальними граматиками без обмежень та допускаються найбільш потужним типом абстрактних автоматів – машинами Тьюрінга. У третьому розділі описано найвужчий згідно з ієрархією Хомського клас регулярних мов, які допускаються (розпізнаються) скінченними автоматами. У четвертому розділі підручника розглянуто клас контекстно-вільних мов, які допускаються (розпізнаються) автоматами зі стековою пам'яттю – у вітчизняній літературі ці пристрої називають магазинними автоматами. У п'ятому розділі коротко розглянуто контекстно-залежні мови, які допускаються машинами Тьюрінга з обмеженою стрічкою – лінійно обмеженими машинами Тьюрінга.

Кінець доведення теорем, лем і т. ін. позначено символом «□». Означення та теореми проілюстровано прикладами. Прості твердження та твердження, що можуть бути доведені за аналогією, запропоновано як вправи. Наприкінці кожного розділу надано запитання та завдання для самоконтролю. Як збірник задач може бути корисним, наприклад, [5, 6].

Вважаємо, що читач опанував базові розділи дискретної математики – алгебру висловлень, алгебру множин, теорію відношень та теорію графів (див., наприклад, [7, 8]).

Розділ 1

Формальні мови та граматики. Ієрархія Хомського

1.1. Алфавіти, слова, конкатенація слів

Алфавітом називають довільну непорожню скінченну множину. Елемент алфавіту називають *символом* або *літерою*. *Словом* над алфавітом A називають довільну скінченну, можливо порожню, послідовність символів із A. Символи у слові можуть повторюватись. Загальну кількість символів у слові α, з урахуванням повторень, називають *довжиною слова* α і позначають через $|\alpha|$. Слово довжини 0 називають *порожнім* і позначають через ε.

Приклад 1.1. Нехай A – множина великих і малих літер українського алфавіту (всього 66 символів). Тоді послідовність $\alpha =$ фіІґфф – слово над A, і $|\alpha| = 6$.

Множину всіх слів над алфавітом A позначають через A^*. Очевидно, що $\varepsilon \in A^*$ для будь-якого алфавіту A. Зазначимо, що самі символи алфавіту A є словами довжини 1, тобто входять як до множини A, так і до множини A^*. (Звичайно, $\varepsilon \notin A$, оскільки $|\varepsilon| = 0 \neq 1$.)

На множині A^* визначають бінарну операцію *конкатенації*: якщо $\alpha = a_{i_1}a_{i_2}\ldots a_{i_m}$, $\beta = a_{j_1}a_{j_2}\ldots a_{j_n}$, де всі $a_k \in A$, то конкатенація $\alpha \cdot \beta = a_{i_1}a_{i_2}\ldots a_{i_m}a_{j_1}a_{j_2}\ldots a_{j_n}$.

Приклад 1.2. Якщо $\alpha = abb$, $\beta = ba$, то $\alpha \cdot \beta = abbba$, $\beta \cdot \alpha = baabb$.

Приклад 1.2 показує, що конкатенація некомутативна. Однак, як легко зрозуміти, конкатенація асоціативна, тобто $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$. Також очевидно, що ε – нейтральний елемент відносно конкатенації: $\alpha \cdot \varepsilon = \varepsilon \cdot \alpha = \alpha$ для довільного $\alpha \in A^*$. Таким чином, алгебрична структура $\langle A^*, \cdot \rangle$ – некомутативна півгрупа з нейтральним елементом, тобто моноїд (детально про алгебричні структури див., наприклад, [7]).

Зазначимо, що, хоча структура $\langle A^*, \cdot \rangle$ ніколи не є групою (жодне непорожнє слово не має оберненого), виконується типовий для груп закон скорочення:

$$(\alpha \cdot x = \alpha \cdot y) \Rightarrow (x = y);\ (x \cdot \beta = y \cdot \beta) \Rightarrow (x = y), \quad \alpha, \beta, x, y \in A^*.$$

У записі виразів з конкатенацією слів символ «$\cdot$» часто опускають, тобто замість $\alpha \cdot \beta$ пишуть $\alpha\beta$. Дужки у записі конкатенації трьох або більше слів, з огляду на асоціативність, також можна опускати, тобто замість $(w_1 \cdot w_2) \cdot w_3$ чи $w_1 \cdot (w_2 \cdot w_3)$ пишуть $w_1 w_2 w_3$.

Надалі використовуватимемо природне в теорії півгруп позначення

$$\alpha^n = \underbrace{\alpha \cdot \alpha \cdot \cdots \cdot \alpha}_{n},\ \alpha^0 = \varepsilon,$$

де $\alpha \in A^*$, $n \in \mathbb{N}$. Так, замість $abbaaa$ можна писати ab^2a^3 (але не a^4b^2 чи b^2a^4, оскільки конкатенація некомутативна), а запис $abababa$ можна скорочено подати як $(ab)^3a$ або $a(ba)^3$.

Слово $\alpha \in A^*$ називають *підсловом* слова $\beta \in A^*$, якщо існують такі $u, v \in A^*$, що $\beta = u\alpha v$. Очевидно, що для певних $\alpha, \beta \in A^*$ можуть існувати різні пари $u, v \in A^*$, такі, що $\beta = u\alpha v$; у таких випадках говорять про різні входження α до β. Кількість входжень α до β позначають через $|\beta|_\alpha$.

Приклад 1.3. 1. Слово ab є підсловом слова abb, оскільки $abb = \varepsilon \cdot ab \cdot b$, і $|abb|_{ab} = 1$.

2. Слово ba не є підсловом слова abb, тобто $|abb|_{ba} = 0$.

3. Слово b є підсловом слова $abcb$, і $|abcb|_b = 2$: $abcb = a \cdot b \cdot cb = abc \cdot b \cdot \varepsilon$.

4. Слово bb є підсловом слова $abbbb$, і $|abbbb|_{bb} = 3$: $abbbb = a \cdot bb \cdot bb = ab \cdot bb \cdot b = abb \cdot bb \cdot \varepsilon$.

5. Будь-яке слово w містить $|w| + 1$ входжень порожнього слова.

1.2. Формальні мови. Операції над формальними мовами

1.2.1. Поняття формальної мови

Надалі вважатимемо, що заданий алфавіт A розбито на дві непорожні підмножини V і T, що не перетинаються: $A = V \cup T$, $V \cap T = \varnothing$. Множину V називають алфавітом *нетермінальних* символів (*нетермінальним* алфавітом), T – алфавітом *термінальних* символів (*термінальним* алфавітом). Якщо не вказано інше, для позначення нетермінальних символів використовуватимемо великі літери англійського алфавіту з індексами або без (A, C_3, $W_{6,2}$ тощо), для позначення термінальних символів – маленькі літери англійського алфавіту з індексами або без (a, c_5, $y_{2,1,5}$ тощо).

Означення 1.1. Формальною мовою (або просто мовою) над алфавітом T називають будь-яку підмножину L множини T^*: $L \subset T^*$.

Приклад 1.4. 1. $L = \{a^{2n}cb \colon n \in \mathbb{N}\}$ – формальна мова над алфавітом $T = \{a, b, c\}$.

2. $L = \{\alpha \in \{0, 1, 2\}^* : |\alpha|_{10} = 0\}$ – формальна мова над алфавітом $T = \{0, 1, 2\}$ (містить слова, у яких за 1 не слідує 0).

3. $L = T$ – формальна мова над T, оскільки $T \subset T^*$.

4. $\varnothing$ та T^* – формальні мови над T, оскільки $\varnothing \subset T^*$ та $T^* \subset T^*$.

1.2.2. Операції над формальними мовами

1. Теоретико-множинні операції. Нехай $L_1, L_2 \subset T^*$. Тоді $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \setminus L_2$ – відповідно об’єднання, перетин та різниця мов L_1 та L_2 як множин; доповнення $\overline{L}$ до мови $L \subset T^*$ визначено відносно T^*, тобто $\overline{L} = T^* \setminus L$.

2. Конкатенація мов. *Конкатенацією* формальних мов $L_1, L_2 \subset T^*$ називають формальну мову $L_1 \cdot L_2 = \{\alpha \cdot \beta \colon \alpha \in L_1, \beta \in L_2\}$ (конкатенація $L_1 \cdot L_2$ містить усі можливі пари конкатенацій слів із мов L_1 і L_2). Як і конкатенація слів, конкатенація мов асоціативна, але, в загальному випадку, некомутативна.

Символ «$\cdot$» у записі конкатенації мов часто опускають, тобто замість $L_1 \cdot L_2$ пишуть L_1L_2. Дужки у записі конкатенації трьох або більше мов, з огляду на асоціативність, також можна опускати, тобто замість $(L_1 \cdot L_2) \cdot L_3$ чи $L_1 \cdot (L_2 \cdot L_3)$ пишуть $L_1 L_2 L_3$.

Приклад 1.5. 1. $\{ab, abc, bc\}\{c, \varepsilon\} = \{abc, abcc, bcc, ab, bc\}$, однак $\{c, \varepsilon\}\{ab, abc, bc\} = \{cab, ab, cabc, abc, cbc, bc\}$.

2. $\{c^2\}\{(ab)^n \colon n \in \mathbb{N}\}\{c\} = \{c^2(ab)^n c \colon n \in \mathbb{N}\}$.

3. $L\{\varepsilon\} = \{\varepsilon\}L = L$ для будь-якої мови L.

4. $L \cdot \varnothing = \varnothing \cdot L = \varnothing$ для будь-якої мови L.

Наведений приклад показує, що слід розрізняти мови $\{\varepsilon\}$ (містить одне слово – порожнє) та $\varnothing$ (є порожньою множиною, тобто взагалі не містить жодного слова).

Як і у випадку конкатенації слів, для конкатенації мов використовуватимемо природне визначення степеня: $L^n = \underbrace{L \cdot L \cdot \dots \cdot L}_{n}$, $L^0 = \{\varepsilon\}$, де $L \subset T^*$, $n \in \mathbb{N}$.

Зауваження 1.1. L^n складається з усіх конкатенацій n слів із L:

$$L^n = \{w_1 w_2 \dots w_n \colon w_1, w_2, \dots, w_n \in L\};$$

випадок $n = 0$ відповідає мові, що містить лише порожнє слово: $L^0 = \{\varepsilon\}$.

Зауваження 1.2. Визначаючи бінарні операції $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 \setminus L_2$ та $L_1 \cdot L_2$, можна відмовитись від умови $L_1, L_2 \subset T^*$, тобто можна вважати, що мови L_1 та L_2 задані над різними алфавітами. Якщо $L_1 \subset T_1^*$, $L_2 \subset T_2^*$, достатньо ввести термінальний алфавіт $T = T_1 \cup T_2$ і вважати, що мови L_1 та L_2 задані над T, оскільки $L_1, L_2 \subset T^*$.

3. Замикання (зірочка) Кліні.[1] *Замиканням* (*зірочкою*) *Кліні* формальної мови L називають формальну мову

$$L^* = L^0 \cup L^1 \cup L^2 \cup \dots \cup L^n \cup \cdots . \tag{1.1}$$

Зауваження 1.3. L^* складається з усіх можливих конкатенацій скінченної кількості слів із L:

$$L^* = \{w_1 w_2 \dots w_n \colon w_1, w_2, \dots, w_n \in L, n \geqslant 0\},$$

де випадок $n = 0$ відповідає порожньому слову: $\varepsilon \in L^*$.

[1] Кліні (Клейні) Стівен Коул (1909–1994) – видатний американський вчений; роботи Кліні спільно з роботами А. Черча, К. Геделя і А. Тьюрінга дали початок теорії обчислюваності; саме Кліні запропонував регулярні вирази як засіб вивчення автоматних (регулярних) мов.

Зазначимо, що множина всіх слів над алфавітом термінальних символів T є замиканням Кліні множини T як формальної мови, що узгоджується з уведеним раніше позначенням T^*.

Приклад 1.6. 1. $\{ab\}^* = \{(ab)^n : n \geqslant 0\}$.
2. $\{ab, c\}^* = \{(ab)^{n_1} c^{m_1} \ldots (ab)^{n_k} c^{m_k} : n_1, \ldots, n_k, m_1, \ldots, m_k \geqslant 0,\ k \geqslant 0\}$.
3. $\{aa, ab, ba, bb\}^* = \{w \in \{a, b\}^* : |w|$ – парне$\}$.
4. $\{a, ab\}^* = \{w \in \{a, b\}^* : b$ у складі w може слідувати тільки за $a\}$.
5. $\varnothing^* = \{\varepsilon\}$.

Із замиканням Кліні пов'язана ще одна операція над мовою L:

$$L^+ = L^1 \cup L^2 \cup \cdots \cup L^n \cup \cdots . \tag{1.2}$$

Зокрема, T^+ складається із всіх непорожніх слів над алфавітом T.

Приклад 1.7. 1. $\{ab\}^+ = \{(ab)^n : n \geqslant 1\}$.
2. $\varnothing^+ = \varnothing$.
3. $\{\varepsilon\}^+ = \{\varepsilon\}$.

Для мов L^* та L^+ безпосередньо із (1.1) та (1.2) випливають такі співвідношення:

$$L^* = L^+ \cup \{\varepsilon\},\ L^+ = L \cdot L^* = L^* \cdot L.$$

Вправа 1.1. Довести, що $(L^+ = L^*) \Leftrightarrow (\varepsilon \in L)$.

Операцію «$^+$», як і «*», можна застосовувати до довільної множини слів над алфавітом A, який містить не тільки термінальні символи; зокрема, A^+ складається із всіх непорожніх слів над алфавітом A. Звичайно, множини A^* і A^+ не є формальними мовами, оскільки A містить принаймні один нетермінальний символ ($V \neq \varnothing$).

4. Обертання мови. *Обертанням (дзеркальним відображенням)* слова $w = a_1 a_2 \cdots a_n \in T^*$ $(a_1, a_2, \ldots, a_n \in T)$ називають слово $w^R = a_n \ldots a_2 a_1$, записане у зворотному порядку. Окрім того, $\varepsilon^R = \varepsilon$. *Обертанням (дзеркальним відображенням)* формальної мови $L \subset T^*$ називають формальну мову $L^R = \{w^R : w \in L\}$.

Приклад 1.8. $\{a^n b^m : n, m \geqslant 0\}^R = \{b^m a^n : n, m \geqslant 0\}$.

1.3. Поняття формальної граматики

Нехай задано алфавіт $A = V \cup T$, $V \cap T = \varnothing$, де V і T – алфавіти нетермінальних і термінальних символів відповідно.

Означення 1.2. Продукцією або правилом підстановки над алфавітом A називають довільну пару $(\alpha, \beta) \in (A^* \times A^*)$; продукцію (α, β) записують у вигляді $\alpha \to \beta$.

Кажуть, що слово $\gamma_2 \in A^*$ можна отримати із слова $\gamma_1 \in A^*$ продукцією $p = (\alpha \to \beta)$, якщо $\gamma_1 = u\alpha v$, $\gamma_2 = u\beta v$ для деяких $u, v \in A^*$. Інакше кажучи, γ_2 отримано із γ_1 заміною деякого входження α до γ_1 на β. Якщо γ_1 містить декілька входжень підслова α, то заміні на β підлягає будь-яке (але одне) входження; у цьому випадку із γ_1 продукцією p можна отримати, як правило, декілька різних слів. Якщо γ_1 не містить підслова α, жодного слова із γ_1 продукцією p отримати неможливо; у цьому випадку говорять, що продукцію p неможливо застосувати до слова γ_1.

Приклад 1.9. Зафіксуємо продукцію $ab \to b$. Тоді:

1) із слова bab можна отримати одне слово bb;

2) із слова $babbab$ можна отримати два слова: $bbbab$ та $babbb$;

3) із слова bba заданою продукцією не можна отримати жодного слова, оскільки bba не містить підслова ab.

Приклад 1.10. Зафіксуємо продукцію $b \to bb$. Тоді:

1) із слова ba можна отримати одне слово bba;

2) із слова $aabab$ можна отримати два слова: $aabbab$ та $aababb$;

3) із слова abb можна отримати одне слово $abbb$ (два входження b до abb після заміни на bb дають однаковий результат);

4) із слова aa заданою продукцією не можна отримати жодного слова.

Продукцію вигляду $\alpha \to \varepsilon$, $\alpha \in (V \cup T)^*$ називають ε-*продукцією.* Застосування такої продукції до слова γ приводить до видалення із слова γ одного із входжень підслова α.

Приклад 1.11. Продукцією $ba \to \varepsilon$ із слова $bacba$ можна отримати два слова: cba та bac; із слова abb цією продукцією не можна отримати жодного слова.

Продукцію вигляду $\varepsilon \to \alpha$ можна застосувати до будь-якого слова γ, оскільки γ містить принаймні одне (насправді $|\gamma| + 1 > 0$) входження підслова ε.

Приклад 1.12. Продукцією $\varepsilon \to b$ із слова ac можна отримати три слова: bac, abc та acb; із слова bc цією продукцією можна отримати тільки два різних слова – bbc та bcb, оскільки перші два входження ε до bc після заміни на b дають одне й те саме слово bbc.

Означення 1.3. Формальною граматикою (або просто граматикою) називають четвірку $G = \langle V, T, P, S\rangle$, де V і T – алфавіти нетермінальних і термінальних символів відповідно, P – непорожня скінченна множина продукцій над алфавітом $A = V \cup T$, $S \in V$ – фіксований нетермінальний символ, який називають джерелом.

Щоб записати кілька продукцій з однаковими лівими частинами, тобто щоб записати n продукцій вигляду $\alpha \to \beta_1$, $\alpha \to \beta_2$, ..., $\alpha \to \beta_n$, часто використовують позначення $\alpha \to \beta_1|\beta_2|\ldots|\beta_n$.

Для граматики $G = \langle V, T, P, S\rangle$ пишуть $\alpha \underset{G}{\Rightarrow} \beta$, якщо слово $\beta \in A^*$ можна отримати із слова $\alpha \in A^*$ деякою продукцією $p \in P$. Отже, з кожною граматикою G пов'язане бінарне відношення «$\underset{G}{\Rightarrow}$» на A^*, яке складається із таких пар $(\alpha, \beta) \in (A^* \times A^*)$, що $\alpha \underset{G}{\Rightarrow} \beta$. Якщо із контексту зрозуміло, про яку граматику йдеться, замість $\alpha \underset{G}{\Rightarrow} \beta$ пишуть $\alpha \Rightarrow \beta$.

Означення 1.4. Транзитивно-рефлексивним замиканням бінарного відношення $R \subset (X \times X)$ називають мінімальне за вкладенням «$\subset$» відношення $R^* \subset (X \times X)$, яке містить R і є транзитивним та рефлексивним.

Із визначення зрозуміло, що для відношення $R \subset (X \times X)$ транзитивно-рефлексивне замикання $R^* \subset (X \times X)$ складається із таких пар $(a_0, a_n) \in (X \times X)$, що

$$a_0Ra_1, a_1Ra_2, \ldots, a_{n-1}Ra_n$$

для деяких $a_1, \ldots, a_{n-1} \in X$, $n \geqslant 0$. Зокрема, для заданої граматики $G = \langle V, T, P, S\rangle$ транзитивно-рефлексивне замикання «$\overset{*}{\underset{G}{\Rightarrow}}$» відношення «$\underset{G}{\Rightarrow}$» складається із таких пар слів (α, β), що β можна отримати (вивести, породити) із α скінченною кількістю застосувань продукцій із множини P. Конкретну послідовність слів $\alpha_1, \ldots, \alpha_n$, таку, що

$$\alpha \underset{G}{\Rightarrow} \alpha_1 \underset{G}{\Rightarrow} \alpha_2 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} \alpha_n \underset{G}{\Rightarrow} \beta,$$

називають *виведенням* β із α у граматиці G. Наголосимо, що всі пари (α, α) $(\alpha \in A^*)$ належать відношенню «$\overset{*}{\underset{G}{\Rightarrow}}$», тобто $\alpha \overset{*}{\underset{G}{\Rightarrow}} \alpha$; можна вважати, що $\beta = \alpha$ виведене із α за 0 застосувань продукцій із P.

Детальніше про різні замикання бінарного відношення див. [7, 8].

Означення 1.5. Формальною мовою $L[G]$, породженою граматикою $G = \langle V, T, P, S \rangle$, називають множину слів над термінальним алфавітом, які можна отримати (породити) із джерела скінченною кількістю застосувань продукцій із множини P:

$$L[G] = \{w \in T^* : S \underset{G}{\overset{*}{\Rightarrow}} w\}.$$

Казатимемо, що слово $w \in T^*$ породжене граматикою G, якщо $S \underset{G}{\overset{*}{\Rightarrow}} w$.

Означення 1.6. Граматики G_1 і G_2 називають еквівалентними, якщо вони породжують ту саму мову:

$$(G_1 \sim G_2) \Leftrightarrow (L[G_1] = L[G_2]).$$

Приклад 1.13. 1. Розглянемо $G_1 = \langle \{S\}, \{a, b\}, \{S \to abS|\varepsilon\}, S \rangle$. Із джерела S застосуванням продукції $p_1 = (S \to abS)$ можна отримати слово abS, застосуванням продукції $p_2 = (S \to \varepsilon)$ – слово ε. Із слова abS продукціями p_1 та p_2 можна отримати відповідно $(ab)^2S$ та ab. Продовжуючи процес, отримуємо із S слова $(ab)^nS$ та $(ab)^n$ для всіх $n \geqslant 0$. На рис. 1.1 показано схему виведення; поруч із символами виведення «$\Rightarrow$» вказано продукції, застосовані на конкретному кроці.

$$\begin{tikzcd} S \arrow[r, Rightarrow, "p_1"] \arrow[d, Rightarrow, "p_2"] & abS \arrow[r, Rightarrow, "p_1"] \arrow[d, Rightarrow, "p_2"] & (ab)^2S \arrow[r, Rightarrow, "p_1"] \arrow[d, Rightarrow, "p_2"] & \cdots \arrow[r, Rightarrow, "p_1"] & (ab)^nS \arrow[r, Rightarrow, "p_1"] \arrow[d, Rightarrow, "p_2"] & \cdots \\ \varepsilon & ab & (ab)^2 & & (ab)^n & \end{tikzcd}$$

Рис. 1.1

Отже, формальна мова, породжена граматикою G_1, містить слова $(ab)^n \in \{a, b\}^*$ $(n \geqslant 0)$. Однак слова $(ab)^nS$ $(n \geqslant 0)$, хоч і виводяться із джерела S продукціями p_1 та p_2, не входять у мову $L[G_1]$, оскільки містять нетермінальний символ S. Остаточно отримуємо:

$$L[G_1] = \{(ab)^n : n \geqslant 0\}.$$

2. Розглянемо $G_2 = \langle \{S, A\}, \{a, b\}, \{S \to aA|\varepsilon,\ A \to bS\}, S \rangle$. Із джерела S застосуванням продукцій $q_1 = (S \to aA)$ та $q_2 = (A \to bS)$ можна

отримати слова $(ab)^n aA$ та $(ab)^n S$ $(n \geqslant 0)$; далі із слів $(ab)^n S$ $(n \geqslant 0)$ продукцією $q_3 = (S \to \varepsilon)$ можна отримати слова $(ab)^n$ $(n \geqslant 0)$ (рис. 1.2).

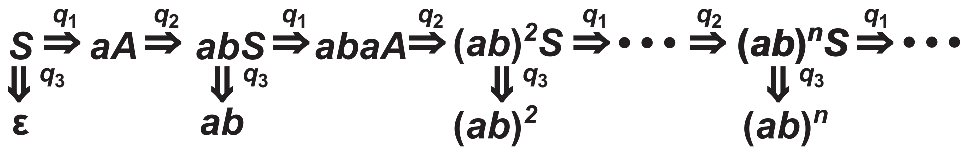

Рис. 1.2

Очевидно, слова $(ab)^n aA$ та $(ab)^n S$ $(n \geqslant 0)$ не входять до мови $L[G_2]$, оскільки містять нетермінальні символи A та S. Остаточно отримуємо:

$$L[G_2] = \{(ab)^n \colon n \geqslant 0\}.$$

Бачимо, що $G_2 \sim G_1$, оскільки обидві граматики породжують ту саму мову $\{(ab)^n \colon n \geqslant 0\}$.

3. Розглянемо $G_3 = \langle \{S\}, \{a, b, c\}, \{S \to aSb|c\}, S \rangle$. Із джерела S продукцією $S \to aSb$ можна отримати слово aSb, із якого цією ж продукцією можна отримати слово a^2Sb^2. Продовжуючи процес, отримуємо із S продукцією $S \to aSb$ всі слова a^nSb^n $(n \geqslant 0)$. Застосувавши до цих слів продукцію $S \to c$, отримуємо слова, що складають мову $L[G_3]$:

$$L[G_3] = \{a^n cb^n \colon n \geqslant 0\}.$$

4. Побудуємо граматику, що породжує мову $\{(ab)^n, a^n cb^n \colon n \geqslant 0\}$. Спочатку зазначимо, що

$$\{(ab)^n, a^n cb^n \colon n \geqslant 0\} = L[G_1] \cup L[G_3],$$

вважаючи, що обидві мови $L[G_1]$ та $L[G_3]$ задані над спільним алфавітом $\{a, b, c\}$ (див. зауваження 1.2). Перепишемо граматики G_1 та G_3 так, щоб вони мали спільний термінальний алфавіт та не мали спільних нетермінальних символів:

$$G_1' = \langle \{S_1\}, \{a, b, c\}, \{S_1 \to abS_1|\varepsilon\}, S_1 \rangle \sim G_1,$$
$$G_3' = \langle \{S_3\}, \{a, b, c\}, \{S_3 \to aS_3b|c\}, S_3 \rangle \sim G_3.$$

До множини продукцій шуканої граматики, що породжує формальну мову $L[G_1'] \cup L[G_3']$, включимо всі продукції граматик G_1' та G_3', а також дві продукції $S \to S_1 | S_3$, отримуючи граматику

$$\langle \{S, S_1, S_3\}, \{a, b, c\}, \{S_1 \to abS_1 | \varepsilon,\ S_3 \to aS_3b | c,\ S \to S_1 | S_3\}, S \rangle.$$

5. Побудуємо граматику, що породжує мову $\{(ab)^n a^m cb^m : n, m \geqslant 0\}$. Спочатку зазначимо, що

$$\{(ab)^n a^m cb^m : n, m \geqslant 0\} = L[G_1'] \cdot L[G_3'],$$

де G_1' та G_3' – граматики, побудовані у попередньому пункті цього прикладу. До множини продукцій шуканої граматики, що породжує мову $L[G_1'] \cdot L[G_3']$, включимо всі продукції граматик G_1' та G_3', а також продукцію $S \to S_1 S_3$, отримуючи граматику

$$\langle \{S, S_1, S_3\}, \{a, b, c\}, \{S_1 \to abS_1 | \varepsilon,\ S_3 \to aS_3b | c,\ S \to S_1 S_3\}, S \rangle.$$

6. Побудуємо граматику, що породжує мову

$$\{a^{n_1} cb^{n_1} a^{n_2} cb^{n_2} \ldots a^{n_k} cb^{n_k} : n_1, n_2, \ldots, n_k \geqslant 0,\ k \geqslant 0\}.$$

Зазначимо, що $\{a^{n_1} cb^{n_1} \ldots a^{n_k} cb^{n_k} : n_1, \ldots, n_k \geqslant 0,\ k \geqslant 0\} = (L[G_3])^*$. Для побудови граматики, що породжує мову $(L[G_3])^*$, додамо до множини нетермінальних символів граматики G_3 новий символ S_1, який оголосимо джерелом; до множини продукцій додамо дві продукції $S_1 \to S_1 S | \varepsilon$. Таким чином, отримуємо граматику

$$\langle \{S, S_1\}, \{a, b, c\}, \{S \to aSb | c,\ S_1 \to S_1 S | \varepsilon\}, S_1 \rangle,$$

яка, вочевидь, породжує шукану мову $(L[G_3])^*$.

Зауваження 1.4. У загальному випадку для побудови граматик, які породжують об'єднання мов $L_1 \cup L_2$, конкатенацію мов $L_1 L_2$ та замикання Кліні L_1^* мови L_1, можна побудувати граматики, які породжують мови L_1 та L_2, а потім скористатися прийомом, викладеним у прикладі 1.13.

1.4. Ієрархія Хомського[1]

Різні граматики можуть породжувати ту саму формальну мову; нагадаємо, що такі граматики згідно з означенням 1.6 називають еквівалентними. Так, еквівалентними є граматики $\langle\{S\},\{a,b\},\{S\to abS|\varepsilon\},S\rangle$ та $\langle\{S,A\},\{a,b\},\{S \to aA|\varepsilon,\ A \to bS\},S\rangle$, бо вони породжують ту саму формальну мову $\{(ab)^n : n \geqslant 0\}$ (див. приклад 1.13). Цю мову породжують також безліч інших граматик – наприклад, граматика $\langle\{S,A\},\{a,b\},\{S\to Ab|\varepsilon,\ A\to Sa\},S\rangle$. Очевидно, що не існує однозначної відповіді на запитання, яка з еквівалентних граматик краща – все залежить від конкретної практичної задачі. У наш час найбільш поширеною класифікацією формальних граматик є *ієрархія Хомського* (див. [1–3]), яка ґрунтується виключно на вигляді продукцій і не бере до уваги кількість продукцій та нетермінальних символів.

Згідно з ієрархією Хомського виділяють чотири класи, або типи, граматик; ці типи традиційно нумерують числами від 0 до 3.

Граматики типу 0. Це загальний тип: будь-яку формальну граматику відносять до типу 0.

Зауваження 1.5. У визначенні граматики типу 0, тобто у загальному визначенні формальної граматики $\langle V,T,P,S\rangle$, часто забороняють появу порожнього слова у лівих частинах продукцій, накладаючи умову $P\subset(A^+\times A^*)$, $A=V\cup T$ (див., наприклад, [4–9]). Це обмеження не звужує клас мов, породжених формальними граматиками, оскільки кожну продукцію $\varepsilon\to\alpha$ ($\alpha\in A^*$) можна замінити продукціями $\xi\to\xi\alpha$ та $\xi\to\alpha\xi$ ($\xi\in A$).

Граматики типу 1. Формальну граматику $\langle V,T,P,S\rangle$ називають *контекстно-залежною* (КЗ-граматикою), або граматикою типу 1, якщо множина P містить лише продукції вигляду $\gamma_1A\gamma_2\to\gamma_1\alpha\gamma_2$, де $A\in V$; $\alpha\in(V\cup T)^+$; $\gamma_1,\gamma_2\in(V\cup T)^*$. Наголосимо, що КЗ-граматики не містять ε-продукцій.

Зауваження 1.6. У визначенні граматики типу 1 іноді дозволяють появу однієї ε-продукції вигляду $S\to\varepsilon$, де S – джерело, вимагаючи, щоб символ S не містився у правій частині жодної продукції. Очевидно, що

[1] Хомський Ноам Абрагам (народ. у 1928 р.) – видатний американський вчений; запропонував апарат формальних граматик для вивчення природних мов, заснувавши генеративний напрям у лінгвістиці.

за такої умови продукція $S \to \varepsilon$ може зустрітися у виведенні не більше одного разу, а отже забезпечить виведення лише одного слова – ε (див., наприклад, [14]).

Граматики типу 2. Формальну граматику $\langle V, T, P, S\rangle$ називають *контекстно-вільною* (КВ-граматикою), або граматикою типу 2, якщо P містить лише продукції вигляду $A \to \alpha$, де $A \in V$, $\alpha \in (V \cup T)^*$.

Граматики типу 3. Формальну граматику $\langle V, T, P, S\rangle$ називають *регулярною*, або граматикою типу 3, якщо множина P містить лише продукції вигляду $A \to aB$, $A \to a$ та $A \to \varepsilon$, де $A, B \in V$, $a \in T$.

Зауваження 1.7. У сучасній літературі часто наводять більш широке визначення граматик типу 3, дозволяючи продукції вигляду $A \to uB$ та $A \to u$, де $A, B \in V$, $u \in T^*$, V і T – відповідно нетермінальний та термінальний алфавіти. Такі граматики називають *праволінійними* (див. [9]). Легко зрозуміти, що праволінійні граматики породжують той самий клас мов, що й регулярні граматики, оскільки кожну продукцію $A \to a_1a_2\ldots a_nB$ та $A \to a_1a_2\ldots a_n$ можна замінити n продукціями вигляду $A \to a_1B_1$, $B_1 \to a_2B_2$, $\ldots$, $B_{n-2} \to a_{n-1}B_{n-1}$, $B_{n-1} \to a_nB$ чи $B_{n-1} \to a_n$ (замість $B_{n-1} \to a_nB$), вводячи нові нетермінальні символи B_1, B_2, $\ldots$, B_{n-1}.

Бачимо, що для визначених класів формальних граматик справджуються такі вкладення:

- кожна граматика типу 3 (регулярна граматика) є граматикою типу 2 (КВ-граматикою);
- кожна граматика типу 2 (КВ-граматика) без ε-продукцій є граматикою типу 1 (КЗ-граматикою);
- кожна формальна граматика є граматикою типу 0.

Приклад 1.14. Формальна граматика $\langle\{S, S_1, S_2\}, \{a, b\}, \{S \to S_1S_2, S_1 \to aS_1|\varepsilon, S_2 \to bS_2|\varepsilon\}, S\rangle$ контекстно-вільна, але не регулярна – продукція $S \to S_1S_2$ не є дозволеного для регулярних граматик вигляду. Однак еквівалентна граматика $\langle\{S, A\}, \{a, b\}, \{S \to aS|bA|\varepsilon, A \to bA|\varepsilon\}, S\rangle$ регулярна.

Приклад 1.15. Формальна граматика $\langle\{S, A, B\}, \{a, b, c\}, \{S \to aAa, aAa \to abBca, bBc \to baAac|baac\}, S\rangle$ контекстно-залежна, але не контекстно-вільна – жодна продукція, окрім першої, не є дозволеного для КВ-граматик вигляду. Зауважимо, що мова, породжена цією граматикою, є контекстно-вільною.

Вправа 1.2. Для граматики з прикладу 1.15 вказати мову, яку вона породжує, та еквівалентну КВ-граматику.

У прикладі 1.13 граматика п. 1 праволінійна, але не регулярна, п. 2 – регулярна, граматики пп. 3–6 – контекстно-вільні.

Відповідно до ієрархії граматик визначають класи формальних мов.

Означення 1.7. Формальну мову називають регулярною, або типу 3 (відповідно контекстно-вільною, або типу 2, та контекстно-залежною, або типу 1), якщо її породжує деяка регулярна (відповідно контекстно-вільна, контекстно-залежна) граматика. Формальну мову, яку породжує довільна граматика, відносять до типу 0.

Приклад 1.16. Формальна мова $\{(ab)^n : n \geqslant 0\}$ регулярна, бо її породжує регулярна граматика $\langle\{S, A\}, \{a, b\}, \{S \to aA|\varepsilon,\ A \to bS\}, S\rangle$. Зазначимо, що цю ж мову породжують безліч інших граматик, серед яких є нерегулярні (див. приклад 1.13); однак ця мова згідно з означенням 1.7 є регулярною, оскільки існує регулярна граматика, яка її породжує.

Приклад 1.17. Побудуємо регулярну формальну граматику, яка породжує мову $\{a^{2n}b^{3k+1} : n \geqslant 0, k \geqslant 0\}$. Із джерела S застосуванням продукцій $S \to aS_1$, $S_1 \to aS$ можна отримати всі слова вигляду $a^{2n}S$, $a^{2n+1}S_1$ $(n \geqslant 0)$. Аналогічними міркуваннями додаємо продукції для породження символів b, остаточно отримуючи граматику

$$\langle\{S, A, B_1, B_2, B_3\}, \{a, b\}, \{S \to aA|bB_1,\ A \to aS,$$
$$B_1 \to bB_2|\varepsilon, B_2 \to bB_3,\ B_3 \to bB_1\}, S\rangle.$$

Приклад 1.18. Формальна мова $\{a^nb^n : n \geqslant 0\}$ контекстно-вільна, бо її породжує КВ-граматика $\langle\{S\}, \{a, b\}, \{S \to aSb|\varepsilon\}, S\rangle$. Однак ця мова нерегулярна, тобто її не породжує жодна регулярна граматика; цей неочевидний факт буде доведено різними способами у підрозд. 3.5.3 (приклад 3.28) та 3.8.3 (приклад 3.47).

Приклад 1.19. Формальна мова $\{a^nb^nc^n : n \geqslant 1\}$ не тільки нерегулярна, а навіть не контекстно-вільна, що буде доведено у підрозд. 4.6 (приклад 4.24). Однак ця мова контекстно-залежна (приклад 5.1).

Вправа 1.3. Будь-яка скінченна формальна мова є регулярною.

Вказівка. Використати прийом, застосований у зауваженні 1.7.

Зауваження 1.8. Зазначимо, що існують класи формальних мов (граматик), які не збігаються з класами ієрархії Хомського. Наприклад, детерміновані контекстно-вільні мови є вужчим класом, ніж клас КВ-мов, але ширшим за клас регулярних мов. Класи граматик $LL(k)$ та $LR(k)$, які використовуються в теорії синтаксичного аналізу та компіляції, є власними підмножинами класу КВ-граматик та власними надмножинами класу регулярних граматик.

Запитання та завдання для самоконтролю

1. Навести означення алфавіту та формальної мови.
2. Перевірити рівності на $\{a, b, c\}^*$: $a^2b^2c^2 = (abc)^2$, $a(bca)^3b = (abc)^3ab$.
3. Навести означення теоретико-множинних операцій об'єднання, перетину, різниці та доповнення формальних мов.
4. Навести означення операцій конкатенації, замикання (зірочки) Кліні та обертання формальних мов.
5. Чи справджується рівність $L = L^n$ для мови $L = \varnothing$? Для яких мов справджується рівність $L^+ = L^*$?
6. Навести означення продукції та формальної граматики.
7. Навести означення ε-продукції.
8. Навести означення мови, породженої формальною граматикою. Які граматики називають еквівалентними?
9. Навести всі типи формальних граматик та формальних мов згідно з ієрархією Хомського, вказати відповідні приклади.
10. Чи може КВ-граматика або КЗ-граматика породжувати регулярну мову? Якщо так, вказати відповідні приклади.

Розділ 2

Машини Тьюрінга і формальні граматики

2.1. Визначення машини Тьюрінга

2.1.1. Неформальний опис

Машина Тьюрінга[1] – абстрактний пристрій, який можна уявити як нескінченну в обидва боки стрічку, розбиту на комірки, та курсор, який в кожний момент часу вказує на певну комірку і може пересуватися вздовж стрічки. Кожна комірка стрічки може містити один символ або не містити жодного; якщо комірка не містить жодного символа, то вважають, що вона містить *порожній символ* Λ. Комірку, на яку вказує курсор, називають *поточною*; символ, що його містить поточна комірка, також називають *поточним*. Множина X тих символів, які можуть містити комірки стрічки, складає *робочий алфавіт*, або просто алфавіт машини Тьюрінга; ця множина є скінченною і непорожньою.

Машина Тьюрінга працює покроково, за дискретними моментами часу $t = 0, 1, 2, \ldots$. На початку роботи, тобто у момент $t = 0$, на стрічку машини подають вхідне слово скінченної довжини, яке складається із символів *зовнішнього алфавіту* $T \subset X$, $T \neq \varnothing$; вважають, що $\Lambda \notin T$. Під час роботи машини у кожний момент t може змінюватись лише поточ-

[1]Тьюрінг Алан (1912–1954) – англійський вчений, один із перших запропонував визначати поняття алгоритму через абстрактні машини.

ний символ. Отже, на стрічці у будь-який момент часу лише скінченна кількість комірок може містити непорожні символи (рис. 2.1).

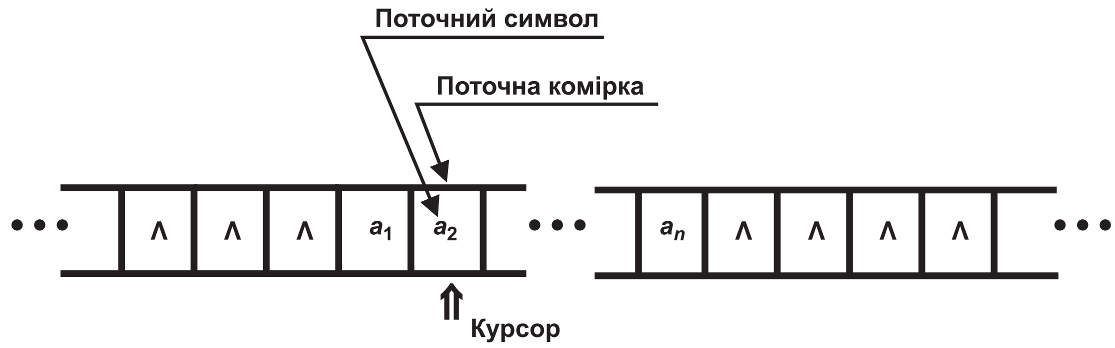

Рис. 2.1

Зауваження 2.1. Якщо не вказано інше, курсор у момент $t = 0$ вказує на перший зліва символ вхідного слова, тобто на крайній лівий непорожній символ.

У кожний момент часу машина Тьюрінга перебуває в одному із своїх *станів*; стан машини у конкретний момент t називають *поточним*. Множина Q станів машини Тьюрінга є скінченною і непорожньою. У множині станів Q виділяють дві підмножини: множину *початкових станів* $I \subset Q$ і множину *заключних станів* $F \subset Q$. На початку роботи машина перебуває в одному із початкових станів $q_0 \in I$ (у випадку $I = \varnothing$ машина взагалі не починає роботу).

Дії машини Тьюрінга на кожному кроці визначає скінченна множина *переходів*, або *команд*. Кожний перехід машини є впорядкованим набором вигляду $((q_1, \alpha_1), (q_2, \alpha_2, d)) \in (((Q \setminus F) \times X) \times (Q \times X \times \{l, r, s\}))$. Набори (q_1, α_1) та (q_2, α_2, d) називатимемо відповідно лівою та правою частинами команди. На кожному кроці машина шукає команду $((q_1, \alpha_1), (q_2, \alpha_2, d))$, таку, щоб елементи $q_1 \in Q \setminus F$ та $\alpha_1 \in X$ були відповідно поточним станом та поточним символом, і виконує такі дії:

1) записує у поточну комірку новий символ $\alpha_2 \in X$ (можливий випадок $\alpha_2 = \alpha_1$);

2) переміщує курсор на одну комірку чи залишає його на місці: переміщує ліворуч, якщо $d = l$; переміщує праворуч, якщо $d = r$; залишає на місці, якщо $d = s$;

3) змінює поточний стан на стан $q_2 \in Q$ (можливий випадок $q_2 = q_1$).

Якщо будь-які два переходи $((q_1, \alpha_1), (q_2, \alpha_2, d))$ та $((\widetilde{q}_1, \widetilde{\alpha}_1), (\widetilde{q}_2, \widetilde{\alpha}_2, \widetilde{d}))$ різняться лівими частинами, тобто $q_1 \neq \widetilde{q}_1$ або $\alpha_1 \neq \widetilde{\alpha}_1$, дії машини на кожному кроці визначені однозначно або не визначені взагалі (якщо немає переходу, в якому ліва частина відповідає поточному стану та поточному символу). Таку машину Тьюрінга, яка, до того ж, має лише один початковий стан, називають *детермінованою*.

У загальному випадку деякі переходи можуть збігатися за лівою частиною, тобто мати вигляд $((q_1, \alpha_1), (q_2, \alpha_2, d))$ та $((q_1, \alpha_1), (\widetilde{q}_2, \widetilde{\alpha}_2, \widetilde{d}))$; тоді дії машини неоднозначні у випадку поточного стану q_1 та поточного символа α_1. Очевидно, що дії машини неоднозначні й за наявності декількох початкових станів $q_0^1, q_0^2, \ldots, q_0^m$, оскільки робота може початися з будь-якого стану q_0^i $(i = 1, 2, \ldots, m)$. Машину Тьюрінга без обмежень щодо детермінованості називають *недетермінованою*.

Детермінована машина Тьюрінга завершує роботу в двох випадках:

1) поточний стан є одним із заключних (нормальне завершення);

2) поточний стан не заключний, і немає команди, ліва частина якої відповідає поточному стану та поточному символу (аварійне завершення).

Зазначимо, що на певних вхідних словах машина Тьюрінга може взагалі ніколи не закінчити роботу.

Приклад 2.1. Розглянемо машину Тьюрінга з робочим алфавітом $X = \{a, b, c, \Lambda\}$, зовнішнім алфавітом $T = \{a, b, c\}$ і множиною станів $Q = \{q_0, q_1\}$, серед яких один початковий стан q_0 та один заключний стан q_1, і системою команд

$$((q_0, a), (q_0, a, r)), ((q_0, \Lambda), (q_0, c, r)), ((q_0, b), (q_1, c, s)).$$

Зазначимо, що ця машина детермінована, оскільки має лише один початковий стан і не містить команд з однаковою лівою частиною.

Вважаємо, що вхідне слово має вигляд $w \in \{a, b, c\}^*$; курсор, згідно зі стандартною домовленістю (див. зауваження 2.1), вказує на перший зліва символ вхідного слова.

Якщо вхідне слово починається з символа a, то робота почнеться з виконання команди $((q_0, a), (q_0, a, r))$: поточний символ та поточний стан не зміняться, курсор зсунеться вправо на одну комірку. Очевидно, що машина виконуватиме команду $((q_0, a), (q_0, a, r))$ доти, поки курсор не «пройде» всі символи a на початку слова. Далі можливі три різні випадки, які й розглянемо.

1. Нехай $w = a^n b\alpha$ ($n \geqslant 0$, $\alpha \in \{a, b, c\}^*$), тобто на початку вхідного слова після n літер a слідує b. Тоді, щойно поточною стане перша зліва літера b, буде виконана команда $((q_0, b), (q_1, c, s))$: у поточну комірку буде записано c, курсор залишиться на місці й машина зупиниться, перейшовши у заключний стан q_1. На рис. 2.2 показано роботу цієї машини на вхідному слові $aabac$ (тут і далі біля курсора показано поточний стан).

Рис. 2.2

2. Нехай $w = a^n c\alpha$ ($n \geqslant 0$, $\alpha \in \{a, b, c\}^*$), тобто на початку вхідного слова після n літер a слідує c. Тоді, щойно поточною стане перша зліва літера c, машина аварійно зупиниться, оскільки серед команд розглядуваної машини немає жодної з лівою частиною (q_0, c) (символ c взагалі не міститься у лівій частині жодного переходу заданої машини). На рис. 2.3 показано роботу цієї машини на вхідному слові $aacab$.

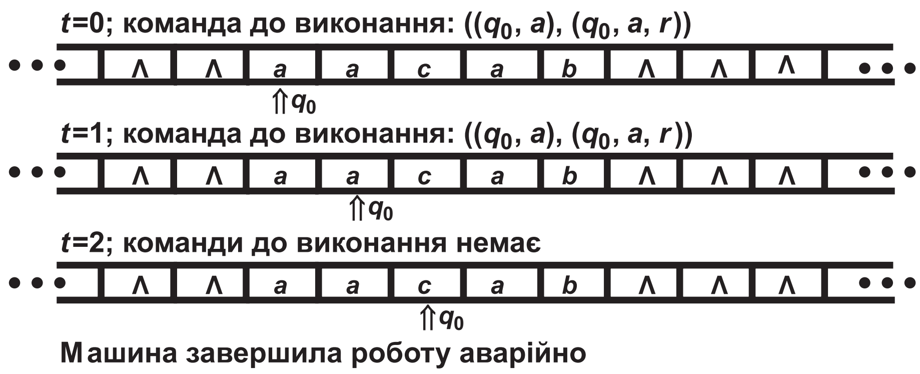

Рис. 2.3

3. Нехай $w = a^n$ ($n \geqslant 0$), тобто вхідне слово не містить літер b та c. Тоді курсор після останньої літери a опиниться на порожньому символі, що приведе до виконання команди $((q_0, \Lambda), (q_0, c, r))$. Таким чином, у першу порожню комірку праворуч від вхідного слова буде записано символ c, машина залишиться у стані q_0, і курсор зсунеться вправо на одну комірку. Далі ситуація не зміниться, оскільки праворуч від курсора тепер лише порожні комірки: машина постійно буде виконувати команду $((q_0, \Lambda), (q_0, c, r))$, записуючи на кожному кроці у поточну порожню комірку літеру c і пересуваючи курсор все далі вправо. На рис. 2.4 показано роботу цієї машини на вхідному слові aa.

Рис. 2.4

2.1.2. Формальний опис

Означення 2.1. Машиною Тьюрінга називають впорядкований набір $\langle Q, X, T, \Lambda, \Delta, I, F\rangle$, де Q, X та T – скінченні непорожні множини, $\Lambda \in X \setminus T$, $\Delta \subset (((Q \setminus F) \times X) \times (Q \times X \times \{l, r, s\}))$, $I \subset Q$, $F \subset Q$. Множину Q називають множиною станів, X – робочим алфавітом, T – зовнішнім алфавітом, Λ – порожнім символом, Δ – множиною переходів (команд), I – множиною початкових станів, F – множиною заключних станів.

Означення 2.2. Машину Тьюрінга $\langle Q, X, T, \Lambda, \Delta, I, F\rangle$ називають детермінованою, якщо виконуються дві умови:

1) для будь-якої пари $(q_1, \alpha_1) \in ((Q \setminus F) \times X)$ існує не більше однієї трійки $(q_2, \alpha_2, d) \in (Q \times X \times \{l, r, s\})$, такої, що $((q_1, \alpha_1), (q_2, \alpha_2, d)) \in \Delta$;

2) машина має лише один початковий стан, тобто $I = \{q_0\}$, $q_0 \in Q$.

Означення 2.3. Конфігурацією машини Тьюрінга $\langle Q, X, T, \Lambda, \Delta, I, F\rangle$ називають довільний набір $(u, q, \alpha, v) \in (X^* \times Q \times X \times X^*)$. Для фіксованої конфігурації (u, q, α, v) елементи q і α називають поточним станом та поточним символом.

На множині всіх конфігурацій машини $M = \langle Q, X, T, \Lambda, \Delta, I, F\rangle$ визначимо бінарне відношення «$\vdash_M$», яке називають *тактом роботи.*

1. Нехай $((q_1, \alpha_1), (q_2, \alpha_2, s)) \in \Delta$. Тоді $(u, q_1, \alpha_1, v) \vdash_M (u, q_2, \alpha_2, v)$ для всіх $u \in X^*$, $v \in X^*$.

2. Нехай $((q_1, \alpha_1), (q_2, \alpha_2, l)) \in \Delta$. Тоді $(u\beta, q_1, \alpha_1, v) \vdash_M (u, q_2, \beta, \alpha_2 v)$ та $(\varepsilon, q_1, \alpha_1, v) \vdash_M (\varepsilon, q_2, \Lambda, \alpha_2 v)$ для $u \in X^*$, $v \in X^*$, $\beta \in X$.

3. Нехай $((q_1, \alpha_1), (q_2, \alpha_2, r)) \in \Delta$. Тоді $(u, q_1, \alpha_1, \beta v) \vdash_M (u\alpha_2, q_2, \beta, v)$ та $(u, q_1, \alpha_1, \varepsilon) \vdash_M (u\alpha_2, q_2, \Lambda, \varepsilon)$ для $u \in X^*$, $v \in X^*$, $\beta \in X$.

Відношення «$\models^*_M$» – транзитивно-рефлексивне замикання відношення «$\vdash_M$» (див. означення 1.4), – складається із таких пар конфігурацій (c_1, c_2), що c_2 можна отримати із c_1 скінченною кількістю тактів «$\vdash_M$». Зазначимо, що $c \models^*_M c$ для будь-якої конфігурації c, оскільки відношення «$\models^*_M$» рефлексивне.

Для конфігурації машини Тьюрінга $\langle Q, X, T, \Lambda, \Delta, I, F\rangle$ використовуватимемо також альтернативне позначення $(u, q, w) \in (X^* \times Q \times X^*)$, яке не передбачає відокремлення поточного символа:

$$(u, q, w) = \begin{cases} (u, q, \alpha, v), \text{ якщо } w = \alpha v,\ \alpha \in X,\ v \in X^*, \\ (u, q, \Lambda, \varepsilon), \text{ якщо } w = \varepsilon. \end{cases}$$

Якщо $M = \langle Q, X, T, \Lambda, \Delta, \{q_0\}, F\rangle$ – детермінована машина Тьюрінга, і для $w \in T^*$, $\widetilde{w} \in T^*$ існують такі $q \in F$, $k \geqslant 0$, $m \geqslant 0$, що $(\varepsilon, q_0, w) \models^*_M (\Lambda^k, q, \widetilde{w}\Lambda^m)$, то кажуть, що M переводить (перетворює) вхідне слово w у вихідне слово $\widetilde{w}$.

Надалі, якщо із контексту зрозуміло, що йдеться саме про машину Тьюрінга M, замість «$\vdash_M$» та «$\models^*_M$» писатимемо відповідно «$\vdash$» та «$\models^*$».

Приклад 2.2. Опишемо роботу машини Тьюрінга з прикладу 2.1 для вхідних слів $aabac$, $aacab$ та aa через відношення «$\vdash$»:

$$(\varepsilon, q_0, a, abac) \vdash (a, q_0, a, bac) \vdash (aa, q_0, b, ac) \vdash (aa, q_1, c, ac);$$
$$(\varepsilon, q_0, a, acab) \vdash (a, q_0, a, cab) \vdash (aa, q_0, c, ab);$$
$$(\varepsilon, q_0, a, a) \vdash (a, q_0, a, \varepsilon) \vdash (aa, q_0, \Lambda, \varepsilon) \vdash (aac, q_0, \Lambda, \varepsilon) \vdash (aacc, q_0, \Lambda, \varepsilon) \vdash \cdots .$$

На слові $aabac$ машина закінчує роботу нормально, перейшовши у заключний стан q_1; на слові $aacab$ машина закінчує роботу аварійно, оскільки немає переходу з лівою частиною (q_0, c); на слові aa машина не закінчує роботу, здійснюючи переходи по конфігураціях $(aac^n, q_0, \Lambda, \varepsilon)$, $n \geqslant 0$.

Приклад 2.3. Розглянемо машину Тьюрінга $\langle Q, X, T, \Lambda, \Delta, I, F\rangle$, яка здійснює копіювання слова $w \in \{a, b\}^*$. Нехай $Q = \{q_0, q_1, q_2, q_3, q_4, q_a, q_b\}$, $X = \{a, b, \widetilde{a}, \widetilde{b}, A, B, \Lambda\}$, $T = \{a, b\}$, $I = \{q_0\}$, $F = \{q_4\}$, множина Δ містить такі переходи (номери проставлено для зручності посилань):

$$((q_0, a), \overset{1}{(q_a, \widetilde{a}, r)}), ((q_0, b), \overset{2}{(q_b, \widetilde{b}, r)}), ((q_a, a), \overset{3}{(q_a, a, r)}), ((q_a, b), \overset{4}{(q_a, b, r)}),$$
$$((q_a, A), \overset{5}{(q_a, A, r)}), ((q_a, B), \overset{6}{(q_a, B, r)}), ((q_b, a), \overset{7}{(q_b, a, r)}), ((q_b, b), \overset{8}{(q_b, b, r)}),$$
$$((q_b, A), \overset{9}{(q_b, A, r)}), ((q_b, B), \overset{10}{(q_b, B, r)}), ((q_a, \Lambda), \overset{11}{(q_1, A, l)}), ((q_b, \Lambda), \overset{12}{(q_1, B, l)}),$$
$$((q_1, A), \overset{13}{(q_1, A, l)}), ((q_1, B), \overset{14}{(q_1, B, l)}), ((q_1, a), \overset{15}{(q_1, a, l)}), ((q_1, b), \overset{16}{(q_1, b, l)}),$$
$$((q_1, \widetilde{a}), \overset{17}{(q_0, a, r)}), ((q_1, \widetilde{b}), \overset{18}{(q_0, b, r)}), ((q_0, A), \overset{19}{(q_2, a, r)}), ((q_0, B), \overset{20}{(q_2, b, r)}),$$
$$((q_2, A), \overset{21}{(q_2, a, r)}), ((q_2, B), \overset{22}{(q_2, b, r)}), ((q_2, \Lambda), \overset{23}{(q_3, \Lambda, l)}), ((q_3, a), \overset{24}{(q_3, a, l)}),$$
$$((q_3, b), \overset{25}{(q_3, b, l)}), ((q_3, \Lambda), \overset{26}{(q_4, \Lambda, r)}), ((q_0, \Lambda), \overset{27}{(q_4, \Lambda, s)}).$$

Покажемо роботу цієї машини на вхідному слові $w = aab$ (над символом «$\vdash$» всюди проставлено номер команди, що виконується):

$$(\varepsilon, q_0, a, ab) \overset{1}{\vdash} (\widetilde{a}, q_a, a, b) \overset{3}{\vdash} (\widetilde{a}a, q_a, b, \varepsilon) \overset{4}{\vdash} (\widetilde{a}ab, q_a, \Lambda, \varepsilon) \overset{11}{\vdash} (\widetilde{a}a, q_1, b, A) \overset{16}{\vdash}$$
$$\overset{16}{\vdash} (\widetilde{a}, q_1, a, bA) \overset{15}{\vdash} (\varepsilon, q_1, \widetilde{a}, abA) \overset{17}{\vdash} (a, q_0, a, bA) \overset{1}{\vdash} (a\widetilde{a}, q_a, b, A) \overset{4}{\vdash} (a\widetilde{a}b, q_a, A, \varepsilon) \overset{5}{\vdash}$$
$$\overset{5}{\vdash} (a\widetilde{a}bA, q_a, \Lambda, \varepsilon) \overset{11}{\vdash} (a\widetilde{a}b, q_1, A, A) \overset{13}{\vdash} (a\widetilde{a}, q_1, b, AA) \overset{16}{\vdash} (a, q_1, \widetilde{a}, bAA) \overset{17}{\vdash}$$
$$\overset{17}{\vdash} (aa, q_0, b, AA) \overset{2}{\vdash} (aa\widetilde{b}, q_b, A, A) \overset{9}{\vdash} (aa\widetilde{b}A, q_b, A, \varepsilon) \overset{9}{\vdash} (aa\widetilde{b}AA, q_b, \Lambda, \varepsilon) \overset{12}{\vdash}$$

$$\overset{12}{\vdash} (aa\widetilde{b}A, q_1, A, B) \overset{13}{\vdash} (aa\widetilde{b}, q_1, A, AB) \overset{13}{\vdash} (aa, q_1, \widetilde{b}, AAB) \overset{18}{\vdash} (aab, q_0, A, AB) \overset{19}{\vdash}$$
$$\overset{19}{\vdash} (aaba, q_2, A, B) \overset{21}{\vdash} (aabaa, q_2, B, \varepsilon) \overset{22}{\vdash} (aabaab, q_2, \Lambda, \varepsilon) \overset{23}{\vdash} (aabaa, q_3, b, \Lambda) \overset{25}{\vdash}$$
$$\overset{25}{\vdash} (aaba, q_3, a, b\Lambda) \overset{24}{\vdash} (aab, q_3, a, ab\Lambda) \overset{24}{\vdash} (aa, q_3, b, aab\Lambda) \overset{25}{\vdash} (a, q_3, a, baab\Lambda) \overset{24}{\vdash}$$
$$\overset{24}{\vdash} (\varepsilon, q_3, a, abaab\Lambda) \overset{24}{\vdash} (\varepsilon, q_3, \Lambda, aabaab\Lambda) \overset{26}{\vdash} (\Lambda, q_4, a, abaab\Lambda),$$

і машина завершує роботу переходом у заключний стан q_4.

Отже, $(\varepsilon, q_0, aab) \overset{*}{\vdash} (\Lambda, q_4, aabaab\Lambda)$. Легко зрозуміти, що для довільного $w \in \{a, b\}^+$ машина копіює непорожнє вхідне слово w, перетворюючи його у слово ww: $(\varepsilon, q_0, w) \overset{*}{\vdash} (\Lambda, q_4, ww\Lambda)$. Коректність роботи у випадку порожнього вхідного слова забезпечує остання команда:

$$(\varepsilon, q_0, \Lambda, \varepsilon) \overset{27}{\vdash} (\varepsilon, q_4, \Lambda, \varepsilon).$$

Вправа 2.1. Побудувати машину Тьюрінга, яка здійснює копіювання слова $w \in T^*$, де $T = \{a_1, a_2, \ldots, a_n\}$, $n \in \mathbb{N}$.

2.2. Машина Тьюрінга як розпізнавач слів

Означення 2.4. Кажуть, що детермінована машина Тьюрінга $M = \langle Q, X, T, \Lambda, \Delta, \{q_0\}, \{q_a, q_r\}\rangle$ розв'язує (розпізнає) формальну мову $L \subset T^*$ зі станом допуску q_a, якщо:

1) для кожного слова $w \in L$ знайдуться такі $k \geqslant 0$, $m \geqslant 0$, що $(\varepsilon, q_0, w) \overset{*}{\vdash}_M (\Lambda^k, q_a, \Lambda^m)$;

2) для кожного слова $w \notin L$ знайдуться такі $k \geqslant 0$, $m \geqslant 0$, що $(\varepsilon, q_0, w) \overset{*}{\vdash}_M (\Lambda^k, q_r, \Lambda^m)$. Формальну мову, яку розв'язує принаймні одна машина Тьюрінга, називають розв'язною, або рекурсивною.

Зауваження 2.2. Якщо машина Тьюрінга $\langle Q, X, T, \Lambda, \Delta, \{q_0\}, \{q_a, q_r\}\rangle$ розв'язує формальну мову $L \subset T^*$ зі станом допуску q_a, то ця ж машина розв'язує мову $\overline{L}$ зі станом допуску q_r.

Приклад 2.4. Розглянемо машину Тьюрінга $\langle Q, X, T, \Lambda, \Delta, I, F\rangle$, яка розв'язує мову $L = \{a^n b^n c^n : n \geqslant 0\}$. Нехай $X = \{a, b, c, *, \Lambda\}$, $T = \{a, b, c\}$, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_a, q_r\}$, $I = \{q_0\}$, $F = \{q_a, q_r\}$ зі станом допуску q_a, множина Δ містить такі переходи:

$$((q_0, a), (q_1, *, r)), ((q_1, a), (q_1, a, r)), ((q_1, b), (q_2, *, r)), ((q_1, *), (q_1, *, r)),$$

$$((q_2,b),(q_2,b,r)),((q_2,c),(q_3,*,l)),((q_2,*),(q_2,*,r)),((q_3,b),(q_3,b,l)),$$
$$((q_3,*),(q_3,*,l)),((q_3,a),(q_1,*,r)),((q_3,\Lambda),(q_4,\Lambda,r)),((q_4,*),(q_4,\Lambda,r)),$$
$$((q_4,\Lambda),(q_a,\Lambda,s)),((q_0,\Lambda),(q_a,\Lambda,s)),((q_0,b),(q_5,*,l)),((q_0,c),(q_5,*,l)),$$
$$((q_1,c),(q_5,*,l)),((q_1,\Lambda),(q_5,\Lambda,l)),((q_2,a),(q_5,*,l)),((q_2,\Lambda),(q_5,\Lambda,l)),$$
$$((q_4,a),(q_5,*,l)),((q_4,b),(q_5,*,l)),((q_4,c),(q_5,*,l)),((q_5,a),(q_5,*,l)),$$
$$((q_5,b),(q_5,*,l)),((q_5,c),(q_5,*,l)),((q_5,*),(q_5,*,l)),((q_5,\Lambda),(q_6,\Lambda,r)),$$
$$((q_6,a),(q_6,\Lambda,r)),((q_6,b),(q_6,\Lambda,r)),((q_6,c),(q_6,\Lambda,r)),((q_6,*),(q_6,\Lambda,r)),$$
$$((q_6,\Lambda),(q_r,\Lambda,s)).$$

Легко перевірити, що ця машина розв'язує мову $L=\{a^nb^nc^n\colon n\geqslant 0\}$. Справді, $(\varepsilon,q_0,a^nb^nc^n)\overset{*}{\vdash}(\Lambda^{3n+1},q_a,\Lambda)$ для $n\in\mathbb{N}$, та $(\varepsilon,q_0,\Lambda)\vdash(\varepsilon,q_a,\Lambda)$; якщо $w\notin L$, то $(\varepsilon,q_0,w)\overset{*}{\vdash}(\Lambda^{3n+1},q_r,\Lambda)$, де $n=|w|$.

Зазначимо, що ця ж машина розв'язує мову $\{a,b,c\}^*\setminus\{a^nb^nc^n\colon n\geqslant 0\}$ зі станом допуску q_r.

Вправа 2.2. Показати роботу машини Тьюрінга з прикладу 2.4 у термінах відношення «$\vdash$» для вхідних слів $a^2b^2c^2\in L$ та $a^2b^2c^3\notin L$.

Означення 2.5. Нехай $M=\langle Q,X,T,\Lambda,\Delta,I,F\rangle$ – довільна, не обов'язково детермінована, машина Тьюрінга. Кажуть, що машина M допускає (сприймає) слово $w\in T^*$, якщо $(\varepsilon,q_0,w)\overset{*}{\vdash}_M(\Lambda^k,q,\Lambda^m)$ для деяких $q_0\in I$, $q\in F$, $k\geqslant 0$, $m\geqslant 0$. Множину слів, які допускає машина M, називають мовою, яку допускає (сприймає) M. Формальну мову, яку допускає принаймні одна машина Тьюрінга, називають напіврозв'язною, або рекурсивно перераховною.

Зауваження 2.3. У термінах неформального опису, якщо машина не допускає заданого вхідного слова, можливий один із трьох випадків:

1) машина зупинилась не в заключному стані (аварійна зупинка);
2) деякі комірки після зупинки залишились непорожніми;
3) машина не зупинилась.

Зауваження 2.4. Якщо машина Тьюрінга $\langle Q,X,T,\Lambda,\Delta,\{q_0\},\{q_a,q_r\}\rangle$ розв'язує хоча б одну мову $L\subset T^*$, то ця машина допускає будь-яке слово $w\in T^*$, тобто допускає формальну мову T^*.

Зауваження 2.5. Нехай машина Тьюрінга $\langle Q,X,T,\Lambda,\Delta,\{q_0\},\{q_a,q_r\}\rangle$ розв'язує мову $L\subset T^*$ зі станом допуску q_a. Тоді машина Тьюрінга $\langle Q,X,T,\Lambda,\Delta,\{q_0\},\{q_a\}\rangle$ допускає мову L, тобто будь-яка розв'язна мова L є напіврозв'язною.

Приклад 2.5. Формальна мова $L = \{a^n b^n c^n : n \geqslant 0\}$ із прикладу 2.4 розв'язна, а отже і напіврозв'язна – її допускає машина $\langle Q, X, T, \Lambda, \Delta, \{q_0\}, \{q_a\}\rangle$, де множини Q, X та Δ такі ж, як і в прикладі 2.4. У термінах неформального визначення машини Тьюрінга ця машина зупиняється в заключному стані q_a з очищенням стрічки для вхідних слів $w \in L$; для вхідних слів $w \notin L$ машина зупиняється у стані q_r, який не є заключним.

Зазначимо, що можна змінити множину переходів так, щоб машина на вхідних словах $w \notin L$ взагалі не зупинялась:

$$\begin{gathered}\widetilde{\Delta} = \{((q_0,a),(q_1,*,r)),((q_1,a),(q_1,a,r)),((q_1,b),(q_2,*,r)),((q_1,*),(q_1,*,r)),\\ ((q_2,b),(q_2,b,r)),((q_2,c),(q_3,*,l)),((q_2,*),(q_2,*,r)),((q_3,b),(q_3,b,l)),\\ ((q_3,*),(q_3,*,l)),((q_3,a),(q_1,*,r)),((q_3,\Lambda),(q_4,\Lambda,r)),((q_4,*),(q_4,\Lambda,r)),\\ ((q_4,\Lambda),(q_a,\Lambda,s)),((q_0,\Lambda),(q_a,\Lambda,s)),((q_0,b),(q_r,*,l)),((q_0,c),(q_r,*,l)),\\ ((q_1,c),(q_r,*,l)),((q_1,\Lambda),(q_r,\Lambda,l)),((q_2,a),(q_r,*,l)),((q_2,\Lambda),(q_r,\Lambda,l)),\\ ((q_4,a),(q_r,*,l)),((q_4,b),(q_r,*,l)),((q_4,c),(q_r,*,l)),((q_r,a),(q_r,*,l)),\\ ((q_r,b),(q_r,*,l)),((q_r,c),(q_r,*,l)),((q_r,*),(q_r,*,l)),((q_r,\Lambda),(q_r,\Lambda,l))\};\end{gathered}$$

стани q_5 та q_6 тепер можна вилучити: $\widetilde{Q} = \{q_0, q_1, q_2, q_3, q_4, q_a, q_r\}$. Очевидно, що машина $\langle \widetilde{Q}, X, T, \Lambda, \widetilde{\Delta}, \{q_0\}, \{q_a\}\rangle$ допускає ту саму мову $\{a^n b^n c^n : n \geqslant 0\}$.

Вправа 2.3. Показати роботу машини Тьюрінга з прикладу 2.5 у термінах відношення «$\vdash$» для вхідного слова $a^2b^2c^3 \notin L$ та порівняти з результатом вправи 2.2.

Приклад 2.6. Наведемо машину Тьюрінга $\langle Q, X, \{a, b\}, \Lambda, \Delta, I, F\rangle$, яка допускає мову $\{a^2, ab\}^*$. Нехай $X = \{a, b, \Lambda\}$, $Q = \{q_0, q_1, q_2, q_3\}$, $I = \{q_0\}$, $\Delta = \{((q_0,a),(q_1,\Lambda,r)),((q_1,a),(q_0,\Lambda,r)),((q_0,a),(q_2,\Lambda,r)),((q_2,b),(q_0,\Lambda,r)),((q_0,\Lambda),(q_3,\Lambda,s))\}$, $F = \{q_3\}$. Очевидно, що ця машина допускає ті й тільки ті слова над алфавітом $\{a, b\}$, які можна зобразити у формі $\alpha_1\alpha_2\ldots\alpha_n$, де $\alpha_1, \alpha_2, \ldots, \alpha_n \in \{a^2, ab\}$, $n \geqslant 0$. Зазначимо, що це недетермінована машина, оскільки містить два переходи з однаковими лівими частинами.

Вправа 2.4. Побудувати детерміновану машину Тьюрінга, яка допускає мову $\{a^2, ab\}^*$.

Теорема 2.1. *Довільну напіврозв'язну формальну мову L допускає деяка детермінована машина Тьюрінга.*

Схема доведення. Нехай мову $L \subset T^*$ допускає деяка, можливо недетермінована, машина $M = \langle Q, X, T, \Lambda, \Delta, I, F\rangle$, $I = \{q_0^1, q_0^2, \ldots, q_0^n\}$. Не втрачаючи загальності можна вважати, що $Q \cap X = \varnothing$. Наведемо схему роботи детермінованої машини $\widetilde{M} = \langle \widetilde{Q}, \widetilde{X}, T, \Lambda, \widetilde{\Delta}, \{q_0\}, \{q_f\}\rangle$, яка допускає слово $w \in T^*$ тоді й тільки тоді, коли це слово допускає машина M.

1. Машина $\widetilde{M}$ записує на своїй стрічці слово

$$\#_l\#q_0^1w\#q_0^2w\#\ldots\#q_0^nw\#\#_r,$$

де «$\#$», «$\#_l$», «$\#_r$» – деякі символи, які не містяться у множинах Q та X.

2. Машина знаходить на стрічці слово вигляду $\#_l\#uqv\#$, де $u, v \in X^*$, $q \in Q$; якщо такого слова на стрічці немає (між символами $\#_l$ та $\#_r$ містяться лише два символи $\#$), машина закінчує роботу аварійно.

3. Якщо $q \in F$, $u = \Lambda^k$, $v = \Lambda^m$, де $k \geqslant 0$, $m \geqslant 0$, машина очищує стрічку та переходить у заключний стан q_f, завершуючи роботу сприйняттям слова w.

4. Для кожної конфігурації $(\widetilde{u}, \widetilde{q}, \widetilde{v}) \in (X^* \times Q \times X^*)$, такої, що $(u, q, v) \vdash_M (\widetilde{u}, \widetilde{q}, \widetilde{v})$, машина $\widetilde{M}$ записує слово $\widetilde{u}\widetilde{q}\widetilde{v}\#\#_r$, починаючи з комірки, де міститься символ $\#_r$.

5. Машина $\widetilde{M}$ записує на стрічку замість слова $\#_l\#uqv\#$ слово $\Lambda^{|u|+|v|+2}\#_l\#$ та переходить до виконання п. 2.

Під час роботи машина $\widetilde{M}$ відтворює на стрічці всі конфігурації, у які машина M може потрапити з початкових конфігурацій (ε, q_0^1, w), (ε, q_0^2, w), $\ldots$,(ε, q_0^n, w) за скінченну кількість кроків. Отже, якщо вихідна машина M допускає слово $w \in T^*$, машина $\widetilde{M}$ врешті-решт знайде конфігурацію $(\Lambda^k, q, \Lambda^n)$ з деякими $q \in F$, $u = \Lambda^k$, $v = \Lambda^m$, $k \geqslant 0$, $m \geqslant 0$, очистить стрічку та завершить роботу. Якщо машина M не допускає слова w, машина $\widetilde{M}$ завершить роботу аварійно. Таким чином, $\widetilde{M}$ допускає слово $w \in T^*$ тоді й тільки тоді, коли слово w допускає машина M, що й треба було довести. □

Вправа 2.5. Для машини Тьюрінга з прикладу 2.6 та вхідного слова a^3b змоделювати роботу машини $\widetilde{M}$, послідовно відтворюючи на стрічці всі конфігурації, у які машина M може потрапити.

Вправа 2.6. 1. Навести приклад формальної мови, яка не є напіврозв'язною.

2. Навести приклад нерозв'язної формальної мови, яка є напіврозв'язною.

Вказівка. Скористатися теоремою про алгоритмічну нерозв'язність проблеми самозастосованості машини Тьюрінга (див., наприклад, [15]).

2.3. Напіврозв'язні мови і формальні граматики

Доведемо, що клас напіврозв'язних мов збігається із класом мов, які породжуються деякою формальною граматикою.

Наступна проста лема – допоміжна, однак має і самостійне значення.

Лема 2.1. *Для будь-якої граматики $G = \langle V, T, P, S\rangle$ існує еквівалентна граматика $G' = \langle V', T, P', S'\rangle$, така, що*

$$((\alpha \to \beta) \in P') \Rightarrow (|\beta|_{S'} = 0)\,,$$

тобто S' не входить до правої частини жодної продукції із P'.

Доведення. Достатньо вибрати довільний символ $S' \notin (V \cup T)$ і покласти $V' = V \cup \{S'\}$ та $P' = P \cup \{S' \to S\}$. □

Ще один простий результат сформулюємо у вигляді вправи.

Вправа 2.7. Для фіксованих $u, v \in A^*$ побудувати детерміновану машину Тьюрінга $M_{u,v} = \langle Q, X, A, \Lambda, \Delta, \{q_0\}, \varnothing\rangle$, таку, що:

1) $\forall \gamma_1, \gamma_2 \in A^*\ \exists\, k, m \geqslant 0\colon (\gamma_1, q_0, u\gamma_2) \overset{*}{\models} (\Lambda^k\gamma_1, q_0, v\gamma_2\Lambda^m)$;

2) $\forall \gamma_1 \in A^*\ \exists\, k, m \geqslant 0\colon (\gamma_1, q_0, w) \overset{*}{\models} (\Lambda^k\gamma_1, q_0, w\Lambda^m)$, якщо $w \neq u\gamma_2$ для жодного $\gamma_2 \in A^*$.

Вказівка. У випадку $|u| \neq |v|$ для коректної заміни u на v зсунути γ_1 або γ_2, використовуючи техніку копіювання з прикладу 2.3.

Теорема 2.2. *Нехай формальна мова L породжується деякою граматикою. Тоді мова L є напіврозв'язною.*

Доведення. Нехай формальна мова L породжується деякою граматикою $G = \langle V, T, P, S\rangle$. З огляду на лему 2.1 вважатимемо, що S не входить до правої частини жодної продукції із P. Побудуємо машину Тьюрінга $M = \langle Q, X, T, \Lambda, \Delta, \{q_0\}, \{q_f\}\rangle$, яка допускає мову L.

Нехай $P = \{\alpha_i \to \beta_i : 1 \leqslant i \leqslant n\}$. Для кожної продукції $(\alpha_i \to \beta_i) \in P$ існує, згідно з результатом вправи 2.7, машина Тьюрінга $M_{\beta_i,\alpha_i} = \langle Q_i, X_i, V \cup T, \Lambda, \Delta_i, \{q_{0,i}\}, \varnothing\rangle$, така, що:

1) $\forall \gamma_1, \gamma_2 \in (V \cup T)^* \ \exists k, m \geqslant 0 \colon (\gamma_1, q_0, \beta_i\gamma_2) \overset{*}{\vdash} (\Lambda^k\gamma_1, q_0, \alpha_i\gamma_2\Lambda^m)$;

2) $\forall \gamma_1 \in (V \cup T)^* \ \exists k, m \geqslant 0 \colon (\gamma_1, q_0, w) \overset{*}{\vdash} (\Lambda^k\gamma_1, q_0, w\Lambda^m)$, якщо $w \neq \beta_i\gamma_2$ для жодного $\gamma_2 \in (V \cup T)^*$.

Без втрати загальності можна вважати, що $Q_i \cap Q_j = \varnothing$ для будь-яких $i \neq j$. Також вважатимемо, що $q_0, q_f \notin Q_i$ для всіх $1 \leqslant i \leqslant n$. Шукану машину M визначимо таким чином:

$$X = X_1 \cup X_2 \cup \cdots \cup X_n, \quad Q = Q_1 \cup Q_2 \cup \cdots \cup Q_n \cup \{q_0, q_f\},$$
$$\Delta = \Delta_1 \cup \Delta_2 \cup \cdots \cup \Delta_n \cup \{r_a, l_a \colon a \in (V \cup T)\} \cup$$
$$\cup \{c_{a,i} \colon a \in (V \cup T),\ 1 \leqslant i \leqslant n\} \cup \{((q_0, S), (q_f, \Lambda, s))\},$$

де $r_a = ((q_0, a), (q_0, a, r))$, $l_a = ((q_0, a), (q_0, a, l))$, $c_{a,i} = ((q_0, a), (q_{0,i}, a, s))$. Зазначимо, що машина M недетермінована.

За побудовою машини M команди $c_{a,i}$ $(a \in (V \cup T))$ забезпечують виконання команд машини M_{β_i,α_i} і, як наслідок, заміну β_i на α_i, якщо курсор вказує на початок слова β_i. Команди r_a та l_a $(a \in (V \cup T))$ пересувають курсор відповідно вправо та вліво вздовж поточного слова. Нарешті, команда $((q_0, S), (q_f, \Lambda, s))$ забезпечує коректне завершення роботи, щойно на стрічці з'явиться символ S. Отже, за побудовою для довільного $w \in T^*$ маємо, що $S \overset{*}{\underset{G}{\Rightarrow}} w$ тоді й тільки тоді, коли $(\varepsilon, q_0, w) \overset{*}{\vdash} (\Lambda^k, q_0, S, \Lambda^m) \vdash (\Lambda^k, q_f, \Lambda, \Lambda^m)$ для деяких $k \geqslant 0, m \geqslant 0$. Таким чином, машина Тьюрінга M допускає ті й тільки ті слова $w \in T^*$, які породжує граматика G. Теорему доведено. □

Теорема 2.3. *Будь-яка напіврозв'язна формальна мова L породжується деякою граматикою.*

Доведення. Нехай мова L напіврозв'язна, тобто допускається деякою машиною Тьюрінга $M = \langle Q, X, T, \Lambda, \Delta, I, F \rangle$.

Побудуємо граматику $G = \langle V, T, P, S \rangle$, таку, що $L = L[G]$. Візьмемо $V = (X \setminus T) \cup \{S\} \cup \{A_{q,a}, \#_l, \#_r \colon a \in (X \setminus T), q \in Q\}$, вважаючи, що X не містить символів $A_{q,a}, \#_l, \#_r$ та S. Далі візьмемо

$$P = \{\beta A_{q_2,\xi} \to A_{q_1,\alpha}\xi \colon ((q_1, \alpha), (q_2, \beta, r)) \in \Delta, \xi \in V\} \cup$$
$$\cup \{A_{q_2,\xi}\beta \to \xi A_{q_1,\alpha} \colon ((q_1, \alpha), (q_2, \beta, l)) \in \Delta, \xi \in V\} \cup$$
$$\cup \{A_{q_2,\beta} \to A_{q_1,\alpha} \colon ((q_1, \alpha), (q_2, \beta, s)) \in \Delta\} \cup$$
$$\cup \{S \to \#_l A_{q_f,\Lambda} \#_r \colon q_f \in F\} \cup \{\#_l A_{q_0,a} \to a \colon q_0 \in I, a \in T\} \cup$$
$$\cup \{\#_r \to \Lambda\#_r, \#_l \to \#_l\Lambda, \Lambda\#_r \to \#_r, \#_l\Lambda \to \#_l, \#_l\#_r \to \varepsilon\}.$$

Легко зрозуміти, що побудована граматика G породжує ті й тільки ті слова, які допускає машина Тьюрінга M. □

Зауваження 2.6. Існують різноманітні модифікації машин Тьюрінга, які, проте, не збільшують «обчислювальної потужності» пристрою і не розширюють класу мов, що сприймаються (див. [12] та, детальніше, [13]).

Додатково про теорію машин Тьюрінга див., наприклад, [16–18].

Запитання та завдання для самоконтролю

1. Навести неформальний опис та принцип роботи машини Тьюрінга. Які дії виконує машина Тьюрінга для команди $((q_1, \alpha_1), (q_2, \alpha_2, d))$?

2. Навести формальне означення машини Тьюрінга.

3. Навести означення детермінованої машини Тьюрінга. Яка, на вашу думку, перевага детермінованих машин Тьюрінга як розпізнавачів слів порівняно з недетермінованими, і навпаки?

4. Навести означення конфігурації та такту роботи машини Тьюрінга. Як вводять відношення «$\vdash_M^*$»?

5. Дати визначення поняттю «машина Тьюрінга розв'язує (розпізнає) формальну мову зі станом допуску». Яку мову називають розв'язною (рекурсивною)?

6. Дати визначення поняттю «машина Тьюрінга допускає формальну мову». Яку мову називають напіврозв'язною (рекурсивно перераховною)?

7. Навести схему побудови детермінованої машини Тьюрінга, яка допускає ту ж саму мову, що й недетермінована.

8. Навести схему доведення напіврозв'язності мови, яку породжує деяка граматика.

9. Навести схему побудови граматики, що породжує напіврозв'язну мову.

10. Навести приклади модифікацій машини Тьюрінга; чи розширюють ці модифікації клас мов, що допускаються, порівняно з класичною машиною Тьюрінга?

Розділ 3

Скінченні автомати та регулярні граматики

3.1. Скінченні автомати: основні поняття

3.1.1. Неформальний опис

Скінченний автомат можна уявити як абстрактний пристрій, що містить необмежену в обидва боки стрічку, розбиту на комірки, та курсор, який в кожний момент часу вказує на певну комірку і може пересуватися вздовж стрічки в одному напрямку – зліва направо. Кожна комірка може містити один символ або не містити жодного; якщо комірка не містить жодного символа, то вважають, що вона містить *порожній символ* Λ. Комірку, на яку вказує курсор, називають *поточною*; символ, що міститься у поточній комірці, також називають *поточним*. Множина T символів, відмінних від Λ, які можуть містити комірки стрічки, складає *алфавіт* скінченного автомата; ця множина є скінченною та непорожньою.

Скінченний автомат працює покроково, за дискретними моментами часу $t = 0, 1, 2, \ldots$. На початку роботи, тобто у момент $t = 0$, на стрічку подають вхідне слово скінченної довжини, яке складається із символів алфавіту T (нагадаємо, що $\Lambda \notin T$). Якщо вхідне слово непорожнє, курсор скінченного автомата в момент $t = 0$ вказує на перший зліва символ вхідного слова (рис. 3.1).

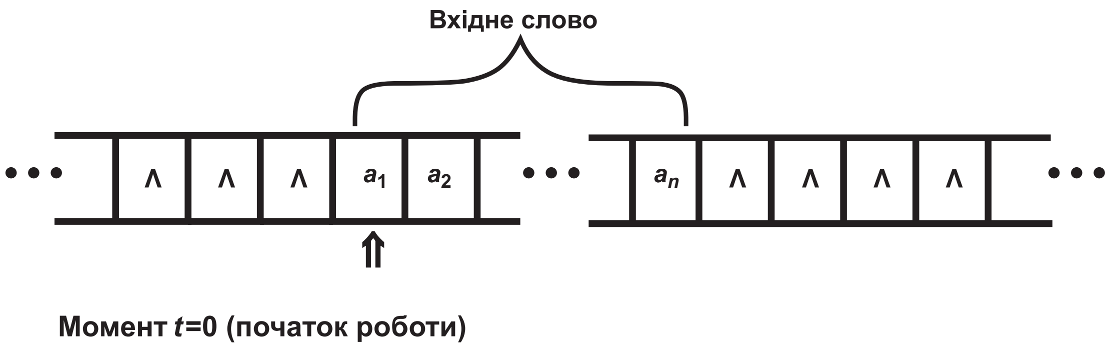

Рис. 3.1

У кожний момент часу скінченний автомат перебуває в одному зі своїх *станів*; стан скінченного автомата в конкретний момент t називають *поточним*. Множина Q станів скінченного автомата є скінченною і непорожньою. У множині Q виділяють дві підмножини: множину I *початкових станів* і множину F *допускаючих станів*. На початку роботи скінченний автомат перебуває в одному із початкових станів $q_0 \in I$.

Дії скінченного автомата на кожному кроці визначає непорожня скінченна множина *переходів* $\Delta \subset (Q \times T \times Q)$. На кожному кроці скінченний автомат за поточним станом $q_1 \in Q$ та поточним символом $\alpha \in T$ шукає перехід (q_1, α, q_2) і виконує такі дії:

1) видаляє поточний символ (записує у поточну комірку порожній символ Λ);

2) переміщує курсор на одну комірку праворуч;

3) змінює поточний стан на стан $q_2 \in Q$ (можливий випадок $q_2 = q_1$).

Якщо для будь-яких $q_1 \in Q$ та $\alpha \in T$ існує єдиний стан $q_2 \in Q$, такий, що $(q_1, \alpha, q_2) \in \Delta$, дії скінченного автомата на кожному кроці визначені однозначно. Такий скінченний автомат, який, до того ж, має лише один початковий стан, називають *детермінованим*. Скінченний автомат без обмежень щодо детермінованості називають *недетермінованим*.

Скінченний автомат закінчує роботу, якщо вхідне слово прочитане повністю (поточним символом є Λ) або для поточного стану та символа немає відповідного переходу.

Кажуть, що скінченний автомат допускає слово $w = a_1 a_2 \dots a_n$ $(a_i \in T,\ 1 \leqslant i \leqslant n)$, якщо існує послідовність станів $q_0 \in I$, $q_1 \in Q$, $q_2 \in Q$, …, $q_{n-1} \in Q$, $q_n \in F$, така, що $(q_{i-1}, a_i, q_i) \in \Delta$ для всіх $1 \leqslant i \leqslant n$. Зокрема, детермінований скінченний автомат допускає вхідне слово $w \in T^*$, якщо, прочитавши слово w, цей автомат закінчує роботу в одному з допускаючих станів. Множину слів, які допускає заданий скінченний автомат, називають формальною мовою, яку цей автомат допускає. Формальну мову, яку допускає принаймні один скінченний автомат, називають *автоматною*.

Приклад 3.1. Над алфавітом $T = \{a, b\}$ розглянемо скінченний автомат із множиною станів $Q = \{q_0, q_1, q_2\}$, множинами початкових та допускаючих станів $I = F = \{q_0\}$ та множиною Δ, що містить шість переходів: $(q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_0), (q_0, b, q_2), (q_2, a, q_2), (q_2, b, q_2)$. Зазначимо, що цей автомат детермінований.

Якщо вхідне слово починається з символа b, буде виконаний перехід (q_0, b, q_2), скінченний автомат перейде у стан q_2 і далі, зчитуючи як a, так і b (переходи (q_2, a, q_2) та (q_2, b, q_2) відповідно), залишиться у стані $q_2 \notin F$. Якщо вхідне слово починається з символа a, буде виконаний перехід (q_0, a, q_1), автомат перейде у стан q_1 і далі, зчитуючи як a, так і b (переходи (q_1, a, q_0) та (q_1, b, q_0) відповідно), перейде у стан $q_0 \in F$. Таким чином, заданий автомат допускає ті й тільки ті слова, що є конкатенацією скінченної кількості слів aa та ab, а отже допускає мову $\{aa, ab\}^*$. На рис. 3.2 показано роботу автомата на вхідному слові $aaab$.

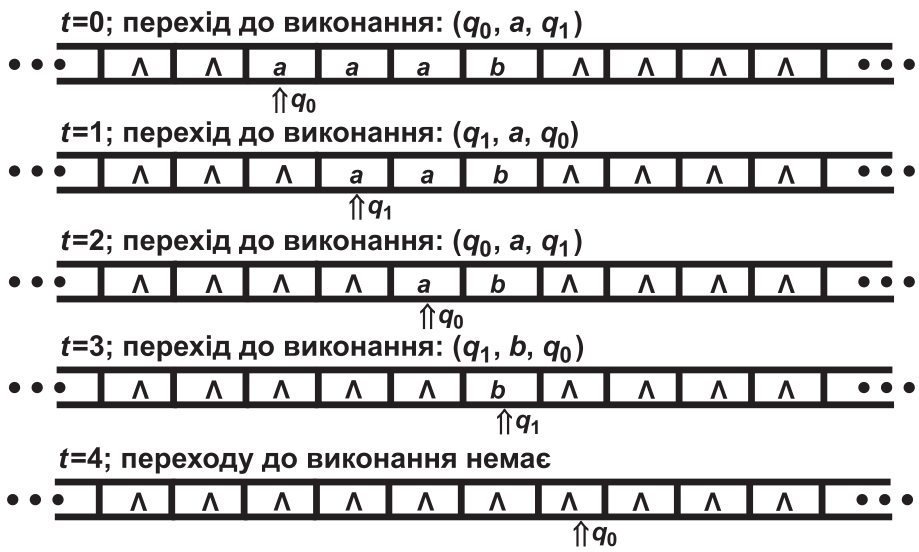

Рис. 3.2

Скінченний автомат завершив роботу у стані $q_0 \in F$, а отже допустив слово $aaab$.

Зауваження 3.1. Скінченний автомат використовує лише ту частину стрічки, яка містить вхідне слово. Стрічка з нескінченною кількістю комірок введена лише для зручності неформального опису, оскільки вхідне слово, хоча й має скінченну довжину, але може бути як завгодно великим.

Зауваження 3.2. Розглядають автомати, які мають вихідний потік (дискретні перетворювачі), серед яких виділяють автомати Мілі та автомати Мура (детальніше див. [4, 12, 19]).

3.1.2. Формальний опис

Означення 3.1. Скінченним автоматом називають впорядкований набір $\langle Q, T, \Delta, I, F\rangle$, де Q та T – скінченні непорожні множини, $\Delta \subset (Q \times T \times Q)$, $I \subset Q$, $F \subset Q$. Множину Q називають множиною станів, T – алфавітом, Δ – відношенням (множиною) переходів, I – множиною початкових станів, F – множиною допускаючих станів.

Означення 3.2. Скінченний автомат $\langle Q, T, \Delta, I, F\rangle$ називають частковим детермінованим, якщо $|I| = 1$ (автомат має лише один початковий стан), і для будь-якої пари $(q_1, a) \in (Q \times T)$ існує не більше одного стану $q_2 \in Q$, такого, що $(q_1, a, q_2) \in \Delta$, та детермінованим, якщо стан q_2 єдиний.

Зауваження 3.3. Перейти від часткового детермінованого автомата до детермінованого можна таким чином: доповнити множину Q додатковим станом $\widetilde{q} \notin Q$ і доповнити множину Δ переходами $(\widetilde{q}, a, \widetilde{q})$, $a \in T$ та переходами $(q_1, a, \widetilde{q})$ для тих пар (q_1, a), для яких не існує відповідного стану q_2 (фактично доповнити станом, у який потрапляють всі невизначені пари q_1 та a).

Означення 3.3. Конфігурацією скінченного автомата $\langle Q, T, \Delta, I, F\rangle$ називають довільний набір $(q, w) \in (Q \times T^*)$.

Зауваження 3.4. Порожній символ Λ у неформальному описі скінченного автомата використовують лише для описання ситуації завершення роботи, коли вхідне слово прочитане повністю, і курсор вказує саме на Λ. Для формального опису скінченного автомата, що не передбачає введення стрічки та курсора, порожній символ не потрібний.

На множині конфігурацій скінченного автомата $M = \langle Q, T, \Delta, I, F\rangle$ визначимо бінарне відношення «$\vdash_M$», яке називають *тактом роботи*:

$$((q_1, w) \vdash_M (q_2, u)) \Leftrightarrow \begin{cases} w = au,\ a \in T; \\ (q_1, a, q_2) \in \Delta, \end{cases}$$

де $q_1, q_2 \in Q$, $w \in T^*$, $u \in T^*$. Інакше кажучи, відношення «$\vdash_M$» містить пари конфігурацій вигляду $((q_1, au), (q_2, u))$, де $(q_1, a, q_2) \in \Delta$.

Якщо $(q_1, a) \vdash_M (q_2, \varepsilon)$, тобто $(q_1, a, q_2) \in \Delta$ для деяких $q_1, q_2 \in Q$, $a \in T$, то кажуть, що стан q_2 *досяжний* зі стану q_1 символом $a \in T$. Зазначимо, що із визначення відношення «$\vdash_M$» негайно випливає еквівалентність

$$((q_1, a) \vdash_M (q_2, \varepsilon)) \Leftrightarrow \big(\forall\, u \in T^* \colon (q_1, au) \vdash_M (q_2, u)\big). \tag{3.1}$$

Конфігурацію $c \in (Q \times T^*)$ скінченного автомата M назвемо *тупиковою*, якщо не існує жодної конфігурації $\tilde{c}$, такої, що $c \vdash_M \tilde{c}$. Очевидно, що конфігурація (q, au) ($q \in Q$, $a \in T$, $u \in T^*$) є тупиковою тоді й тільки тоді, коли $\forall\, \tilde{q} \in Q \colon (q, a, \tilde{q}) \notin \Delta$. Конфігурація (q, ε) є, очевидно, тупиковою для будь-якого $q \in Q$.

Відношення «$\vdash^*_M$» – транзитивно-рефлексивне замикання відношення «$\vdash_M$» (див. означення 1.4), – складається із таких пар конфігурацій (c_0, c_n), що

$$c_0 \vdash_M c_1 \vdash_M c_2 \cdots \vdash_M c_n$$

для деякої послідовності конфігурацій $c_1, c_2, \ldots, c_{n-1}$ ($n \geqslant 0$).

Якщо $(q_1, w) \vdash^*_M (q_2, \varepsilon)$, то кажуть, що стан q_2 *досяжний* із стану q_1 словом $w \in T^*$; зокрема, будь-який стан q досяжний із $q \in Q$ (із самого себе) порожнім словом $\varepsilon \in T^*$, оскільки $(q, \varepsilon) \vdash^*_M (q, \varepsilon)$. З еквівалентності (3.1) індукцією за довжиною слова $w \in T^*$ легко вивести еквівалентність

$$((q_1, w) \vdash^*_M (q_2, \varepsilon)) \Leftrightarrow \big(\forall\, u \in T^* \colon (q_1, wu) \vdash^*_M (q_2, u)\big).$$

Надалі, якщо з контексту зрозуміло, що йдеться саме про скінченний автомат M, замість «$\vdash_M$» та «$\vdash^*_M$» писатимемо відповідно «$\vdash$» та «$\vdash^*$».

Означення 3.4. Кажуть, що скінченний автомат $M = \langle Q, T, \Delta, I, F\rangle$ допускає (сприймає) слово $w \in T^*$, якщо $(q_0, w) \vdash^*_M (q, \varepsilon)$ для деяких $q_0 \in I$ та $q \in F$. Множину слів $L[M]$, які допускає скінченний автомат M, називають формальною мовою, яку допускає (сприймає, розпізнає) скінченний автомат M. Формальну мову, яку допускає хоча б один скінченний автомат, називають автоматною.

Приклад 3.2. Продемонструємо роботу скінченного автомата із прикладу 3.1 на слові $aaab$ у термінах відношення «$\vdash$»:

$$(q_0, aaab) \vdash (q_1, aab) \vdash (q_0, ab) \vdash (q_1, b) \vdash (q_0, \varepsilon),$$

тобто $(q_0, aaab) \overset{*}{\vdash}_M (q_0, \varepsilon)$.

Оскільки $q_0 \in I$ та $q_0 \in F$, автомат допускає слово $aaab$. Як вже зазначалось (див. приклад 3.1), заданий скінченний автомат допускає ті й тільки ті слова, що є конкатенацією скінченної кількості слів aa та ab, тобто допускає мову $\{aa, ab\}^*$.

Означення 3.5. Скінченні автомати M_1 та M_2 називають еквівалентними, якщо M_1 та M_2 задані над спільним алфавітом та допускають ту саму мову:

$$(M_1 \sim M_2) \Leftrightarrow (L[M_1] = L[M_2]).$$

Приклад 3.3. Скінченний автомат із прикладу 3.1 еквівалентний скінченному автомату над алфавітом $\{a, b\}$ із множиною станів $\{q_0, q_1, q_2\}$, єдиним початковим і єдиним допускаючим станом q_0 та множиною переходів $\{(q_0, a, q_1), (q_1, a, q_0), (q_0, a, q_2), (q_2, b, q_0)\}$. Зазначимо, що цей автомат недетермінований. Так, вхідним словом $aaab$ із q_0 можна потрапити у три різні тупикові конфігурації:

$$\begin{aligned}
&(q_0, aaab) \vdash (q_2, aab);\\
&(q_0, aaab) \vdash (q_1, aab) \vdash (q_0, ab) \vdash (q_1, b);\\
&(q_0, aaab) \vdash (q_1, aab) \vdash (q_0, ab) \vdash (q_2, b) \vdash (q_0, \varepsilon).
\end{aligned}$$

Таким чином, $(q_0, aaab) \overset{*}{\vdash} (q_2, aab)$, $(q_0, aaab) \overset{*}{\vdash} (q_1, b)$ та $(q_0, aaab) \overset{*}{\vdash} (q_0, \varepsilon)$. Отже, автомат допускає слово $aaab$, оскільки $(q_0, aaab) \overset{*}{\vdash} (q_0, \varepsilon)$, де $q_0 \in I$, $q_0 \in F$. Легко зрозуміти, що цей автомат допускає ті й тільки ті слова, які є конкатенацією скінченної кількості слів aa та ab, тобто допускає мову $\{aa, ab\}^*$, а отже справді еквівалентний автомату із прикладу 3.1.

3.1.3. Задання скінченного автомата за допомогою графа

Скінченний автомат $\langle Q, T, \Delta, I, F \rangle$ можна однозначно (у межах фіксованого алфавіту T) задати за допомогою міченого орієнтованого мультиграфа, побудованого за такими правилами:

1) кожна вершина графа помічена деяким станом $q \in Q$, і кожний стан помічає в точності одну вершину;

2) кожне ребро графа помічене деяким символом $a \in T$;

3) орієнтоване ребро з міткою $a \in T$ веде від вершини з міткою $q_1 \in Q$ до вершини з міткою q_2 тоді й тільки тоді, коли $(q_1, a, q_2) \in \Delta$;

4) вершини графа позначені колами однакового (у межах певного графа) радіуса, відповідну мітку стану проставляють всередині кола (рис. 3.3, *а*); вершини, що відповідають початковим станам, помічають стрілкою (рис. 3.3, *б*); вершини, що відповідають допускаючим станам, позначають додатковим зовнішнім колом (рис. 3.3, *в*); вершини, що відповідають станам, які є одночасно початковими і допускаючими, мають стрілку і додаткове зовнішнє коло (рис. 3.3, *г*).

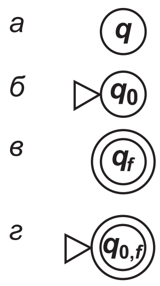

Рис. 3.3

Приклад 3.4. На рис. 3.4 наведено граф скінченного автомата з прикладу 3.1, на рис. 3.5 – граф еквівалентного автомата з прикладу 3.3.

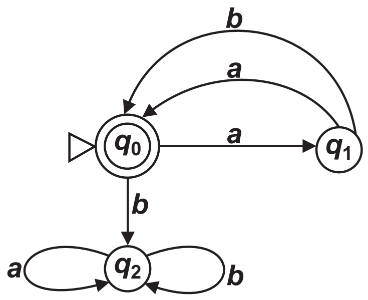

Рис. 3.4

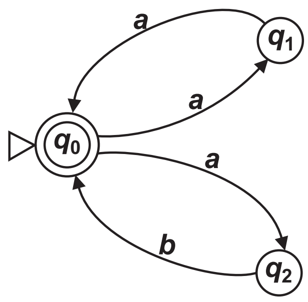

Рис. 3.5

Приклад 3.5. На рис. 3.6 зображено граф скінченного автомата, що допускає таку формальну мову:

$$\{a^n b^{m+1} c^{2k+1} : n \geqslant 0, m \geqslant 0, k \geqslant 0\} \cup$$
$$\cup \{a^n : n \geqslant 0\} \cup \{b^m c^{2k+1} : m \geqslant 0, k \geqslant 0\}.$$

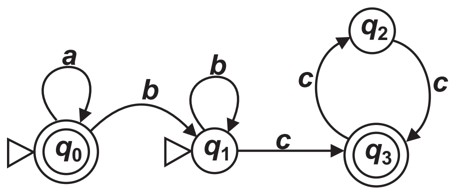

Рис. 3.6

Цей граф зображує автомат із множиною станів $Q = \{q_0, q_1, q_2, q_3\}$, множинами початкових та допускаючих станів $I = \{q_0, q_1\}$ та $F = \{q_0, q_3\}$ відповідно, і множиною Δ із шести переходів:

$$\Delta = \{(q_0, a, q_0), (q_0, b, q_1), (q_1, b, q_1), (q_1, c, q_3), (q_3, c, q_2), (q_2, c, q_3)\}.$$

Якщо вважати, що алфавіт не містить символів, відмінних від a, b та c, то граф зображує скінченний автомат $\langle Q, \{a, b, c\}, \Delta, I, F\rangle$ із щойно вказаними множинами Q, Δ, I, F. Очевидно, що автомат не є детермінованим, оскільки має два початкових стани.

3.2. Детерміновані скінченні автомати

3.2.1. Функція переходів

Нехай $M = \langle Q, T, \Delta, I, F\rangle$ – детермінований скінченний автомат (див. означення 3.2). Поведінку автомата M, враховуючи його детермінованість, зручно вивчати за допомогою *функції переходів* $\delta\colon (Q \times T) \to Q$:

$$(\delta(q_1, a) = q_2) \Leftrightarrow ((q_1, a, q_2) \in \Delta) \tag{3.2}$$

для всіх $q_1, q_2 \in Q$, $a \in T$. Зазначимо, що співвідношення (3.2), завдяки детермінованості автомата M, дозволяє однозначно визначити функцію переходів δ через відношення переходів Δ, і навпаки – відношення Δ через функцію δ. Також зазначимо, що для недетермінованих автоматів розглядають [4,10,11] (багатозначну) функцію переходів $\delta\colon (Q\times T) \to 2^Q$, де 2^Q – множина всіх підмножин множини Q.

Приклад 3.6. Зведемо в таблицю значення функції переходів для детермінованого скінченного автомата із прикладу 3.1 (див. табл. 3.1). Рядки таблиці відповідають символам алфавіту $a, b \in T$, стовпці – станам $q_0, q_1, q_2 \in Q$.

Таблиця 3.1

	q_0	q_1	q_2
a	q_1	q_0	q_2
b	q_2	q_0	q_2

Таблицю типу 3.1 із значеннями функції переходів детермінованого автомата M називають *таблицею переходів* автомата M.

Введемо до розгляду функцію $\delta^* : (Q \times T^*) \to Q$, яка є природним розширенням функції переходів δ:

$$\delta^*(q, \varepsilon) = q, \quad \delta^*(q, wa) = \delta(\delta^*(q, w), a), \tag{3.3}$$

де $q \in Q$, $w \in T^*$, $a \in T$. Визначену рекурентними співвідношеннями (3.3) функцію δ^* назвемо *розширеною функцією переходів* автомата M.

Вправа 3.1. Користуючись співвідношенням (3.3), довести такі властивості розширеної функції переходів δ^*:

1) $\forall\, a \in T\, \forall\, q \in Q \colon \delta^*(q, a) = \delta(q, a)$;
2) $\forall\, a, b \in T\, \forall\, q \in Q \colon \delta^*(q, ab) = \delta(\delta(q, a), b)$;
3) $\forall\, u, v \in T^*\, \forall\, q \in Q \colon \delta^*(q, uv) = \delta^*(\delta^*(q, u), v)$.

Зауваження 3.5. Для доведення властивості 3 скористатись індукцією за $|v| \geqslant 0$.

Вправа 3.2. Узагальнити властивості 1 і 2 на випадок слова довільної довжини n:

$\forall\, a_1, a_2, \ldots, a_n \in T\, \forall\, q \in Q \colon \delta^*(q, a_1 a_2 \ldots a_n) = \delta(\cdots \delta(\delta(q, a_1), a_2), \ldots, a_n)$.

Індукцією за довжиною слова $w \in T^*$ легко вивести еквівалентність, яка пов'язує розширену функцію переходів з відношенням «$\overset{*}{\vdash}$»:

$$(\delta^*(q_1, w) = q_2) \Leftrightarrow ((q_1, w) \overset{*}{\vdash} (q_2, \varepsilon)),$$

де $q_1, q_2 \in Q$, $w \in T^*$.

Формальну мову $L[M]$, яку допускає детермінований скінченний автомат $M = \langle Q, T, \Delta, \{q_0\}, F \rangle$, також зручно описати в термінах розширеної функції переходів:

$$L[M] = \{w \in T^* \colon \delta^*(q_0, w) \in F\}. \tag{3.4}$$

Зауваження 3.6. У термінах неформального опису розширена функція переходів δ^* обчислює стан $q_2 = \delta^*(q_1, w)$ автомата M після зчитування слова $w \in T^*$, якщо на початку зчитування слова w автомат перебував у стані q_1.

Приклад 3.7. Опишемо розширену функцію переходів для автомата із прикладу 3.1 ($w \in \{aa, ab\}^*$, $u \in \{a, b\}^*$):

1) $\delta^*(q_0, w) = q_0$, $\delta^*(q_0, wa) = q_1$, $\delta^*(q_0, wbu) = q_2$;
2) $\delta^*(q_1, aw) = \delta^*(q_1, bw) = q_0$, $\delta^*(q_1, awa) = q_1$, $\delta^*(q_1, awbu) = q_2$;
3) $\delta^*(q_2, u) = q_2$.

Бачимо, що $\delta^*(q_0, w) \in F = \{q_0\}$ тоді й тільки тоді, коли $w \in \{aa, ab\}^*$, тобто $L[M] = \{aa, ab\}^*$.

Приклад 3.8. Побудуємо скінченний детермінований автомат, який допускає мову $L = \{w \in \{a, b\}^* : |w|_a \,\vdots\, 3\}$.

Можливі варіанти кількості входжень a: $3k$, $3k{+}1$, $3k{+}2$, $k \in \mathbb{N} \cup \{0\}$, автомат має допустити слова з $3k$ входженнями a та не допустити слова з $3k + 1$ і $3k + 2$ входженнями a. Отже, для побудови автомата достатньо трьох станів q_0, q_1 та q_2; $(q_0, w) \overset{*}{\vdash} (q_i, \varepsilon)$ тоді й тільки тоді, коли w містить $3k + i$ входжень a ($i = 0, 1, 2$). Граф відповідного автомата зображено на рис. 3.7.

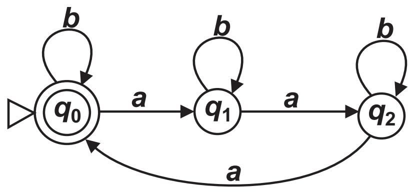

Рис. 3.7

Функція переходів має вигляд $\delta(q_i, a) = q_{(i+1) \bmod 3}$, $\delta(q_i, b) = q_i$, де « mod 3» тут і далі позначає остачу від ділення на 3; розширена функція переходів – $\delta^*(q_i, w) = q_{(i+|w|_a) \bmod 3}$.

Вправа 3.3. Побудувати скінченний детермінований автомат, який допускає мову $L = \{w \in \{a, b\}^* : |w|_a \geqslant 3\}$.

Вказівка. Використовуючи прийом, викладений у прикладі 3.8, відслідкувати можливі варіанти кількості входжень a: 0, 1, 2, 3 та більше. Розширена функція переходів $\delta^*(q_i, w) = q_{\min\{i+|w|_a, 3\}}$.

Приклад 3.9. Побудуємо скінченний детермінований автомат, який допускає мову $L = \{w \in \{a, b\}^* : |w|_a \,\vdots\, 3, |w|_b \,\vdots\, 2\}$.

Можливі варіанти кількості входжень a: $3k$, $3k + 1$, $3k + 2$, $k \in \mathbb{N} \cup \{0\}$; можливі варіанти кількості входжень b: $2m$, $2m + 1$, $m \in \mathbb{N} \cup \{0\}$. Отже, для побудови автомата достатньо шести станів q_{ij} ($i = 0, 1, 2$, $j = 0, 1$): $(q_{00}, w) \overset{*}{\vdash} (q_{ij}, \varepsilon)$ тоді й тільки тоді, коли w містить $3k + i$ входжень a та $2m + j$ входжень b. Граф відповідного автомата показано на рис 3.8. Розширена функція переходів $\delta^*(q_{ij}, w) = q_{((i+|w|_a) \bmod 3)\ ((j+|w|_b) \bmod 2)}$.

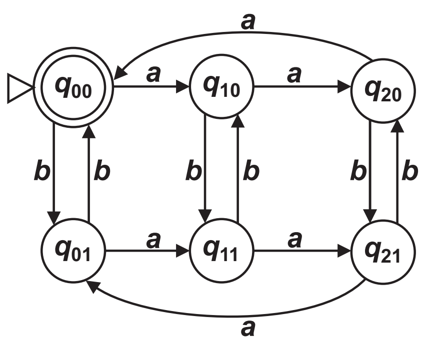

Рис. 3.8

Зауважимо, що в загальному випадку для мови $|w|_a \,\vdots\, n_1$, $|w|_b \,\vdots\, n_2$ автомат має n_1 стовпців та n_2 рядків.

Вправа 3.4. Використовуючи прийом, викладений у прикладі 3.9, побудувати скінченні детерміновані автомати, які допускають мови:

1. $L = \{w \in \{a, b\}^* : (|w|_a \,\vdots\, 3) \vee (|w|_b \,\vdots\, 2)\}$;
2. $L = \{w \in \{a, b\}^* : |w|_a \,\vdots\, 3, 1 \leqslant |w|_b \leqslant 2\}$;
3. $L = \{w \in \{a, b\}^* : |w|_a \geqslant 3, 1 \leqslant |w|_b \leqslant 2\}$;

4. $L = \{w \in \{a,b\}^* : (|w|_a \geqslant 3) \vee (1 \leqslant |w|_b \leqslant 2)\}$;
5. $L = \{w \in \{a,b\}^* : |w|_a \bmod 3 = |w|_b \bmod 3\}$;
6. $L = \{w \in \{a,b,c\}^* : |w|_a \,\vdots\, 3, |w|_b \,\vdots\, 3, |w|_c \,\vdots\, 2\}$.

Виявляється, що класи довільних скінченних автоматів та класи детермінованих скінченних автоматів збігаються.

Теорема 3.1. *Нехай $M = \langle Q, T, \Delta, I, F\rangle$ – довільний скінченний автомат. Тоді існує детермінований скінченний автомат M_0, еквівалентний автомату M.*

Доведення. Визначимо автомат $M_0 = \langle Q_0, T, \Delta_0, I_0, F_0\rangle$ таким чином: $Q_0 = 2^Q$, $I_0 = \{I\}$, $F_0 = \{A \subset Q \colon A \cap F \neq \varnothing\}$. Враховуючи співвідношення (3.2), замість множини переходів Δ_0 задамо функцію переходів δ_0:

$$\begin{aligned}\delta_0(A, \alpha) = \{q_2 \in Q \colon \exists\, q_1 \in A \colon (q_1, \alpha, q_2) \in \Delta\} = \\ = \{q_2 \in Q \colon \exists\, q_1 \in A \colon (q_1, \alpha) \vdash_M (q_2, \varepsilon)\},\end{aligned}$$

де $A \in 2^Q$ $(A \subset Q)$, $\alpha \in T$. Враховуючи рівність (3.3) та визначення транзитивно-рефлексивного замикання, запишемо значення розширеної функції переходів автомата M_0 через відношення «$\vdash_M^*$»:

$$\delta_0^*(A, w) = \{q_2 \in Q \colon \exists\, q_1 \in A \colon (q_1, w) \vdash_M^* (q_2, \varepsilon)\},$$

де $A \in 2^Q$, $w \in T^*$. Враховуючи співвідношення (3.4), доводимо, що детермінований автомат M_0 допускає $w \in T^*$ тоді й тільки тоді, коли w допускає вихідний автомат M:

$$\begin{gathered}(w \in L[M_0]) \Leftrightarrow (\delta_0^*(I, w) \in F_0) \Leftrightarrow (\delta_0^*(I, w) \cap F \neq \varnothing) \Leftrightarrow (\exists\, q_2 \in F \colon \\ q_2 \in \delta_0^*(I, w)) \Leftrightarrow (\exists\, q_2 \in F\, \exists\, q_1 \in I \colon (q_1, w) \vdash_M^* (q_2, \varepsilon)) \Leftrightarrow (w \in L[M]).\end{gathered}$$

Таким чином, $L[M] = L[M_0]$, тобто вихідний автомат M еквівалентний детермінованому автомату M_0, що завершує доведення теореми. □

Доведення теореми 3.1 надає прямий метод побудови детермінованого скінченного автомата $M_0 = \langle 2^Q, T, \Delta_0, I_0, F_0\rangle$, еквівалентного заданому довільному (можливо недетермінованому) скінченному автомату $M = \langle Q, T, \Delta, I, F\rangle$.

Приклад 3.10. Для недетермінованого скінченного автомата

$$\langle\{q_0,q_1,q_2\},\{a,b\},\{(q_0,a,q_1),(q_1,a,q_0),(q_0,a,q_2),(q_2,b,q_0)\},\{q_0\},\{q_0\}\rangle$$

(див. приклад 3.3) побудуємо відповідний детермінований автомат за методом із доведення теореми 3.1 (див. таблицю переходів 3.2). Справді, рівності $\delta_0(\varnothing,a) = \delta_0(\varnothing,b) = \varnothing$ очевидні. Із стану q_0 символом a досяжні стани q_1 та q_2, символом b жодний зі станів недосяжний: $\delta_0(\{q_0\},a) = \{q_1,q_2\}$, $\delta_0(\{q_0\},b) = \varnothing$. Із стану q_1 символом a досяжний стан q_0, символом b жодний зі станів недосяжний: $\delta_0(\{q_1\},a) = \{q_0\}$, $\delta_0(\{q_1\},b) = \varnothing$. Із стану q_2 символом a жодний зі станів недосяжний; символом b досяжний стан q_0: $\delta_0(\{q_2\},a) = \varnothing$, $\delta_0(\{q_2\},b) = \{q_0\}$. Далі,

$$\delta_0(\{q_0,q_1\},a) = \delta_0(\{q_0\},a)\cup\delta_0(\{q_1\},a) = \{q_1,q_2\}\cup\{q_0\} = \{q_0,q_1,q_2\},$$
$$\delta_0(\{q_0,q_1\},b) = \delta_0(\{q_0\},b)\cup\delta_0(\{q_1\},b) = \varnothing\cup\varnothing = \varnothing;$$

останні три стовпці табл. 3.2 заповнюються аналогічно.

Таблиця 3.2

	$\varnothing$	$\{q_0\}$	$\{q_1\}$	$\{q_2\}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$	$\{q_1,q_2\}$	$\{q_0,q_1,q_2\}$
a	$\varnothing$	$\{q_1,q_2\}$	$\{q_0\}$	$\varnothing$	$\{q_0,q_1,q_2\}$	$\{q_1,q_2\}$	$\{q_0\}$	$\{q_0,q_1,q_2\}$
b	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_0\}$	$\varnothing$	$\{q_0\}$	$\{q_0\}$	$\{q_0\}$

Множини початкових та допускаючих станів відповідно $I_0 = \{q_0\}$ та $F_0 = \{\{q_0\},\{q_0,q_1\},\{q_0,q_2\},\{q_0,q_1,q_2\}\}$.

Отже, автомат M_0 має $2^3 = 8$ станів, із яких лише 3 досяжні з початкового стану I_0: стани $\{q_1,q_2\}$, $\varnothing$ і, власне, I_0 (див. табл. 3.2). Вилучивши із автомата M_0 ті 5 станів, які недосяжні з початкового, отримуємо детермінований скінченний автомат, граф якого зображено на рис. 3.9. Зазначимо, що побудований автомат збігається з автоматом із прикладу 3.1 з точністю до перейменування станів.

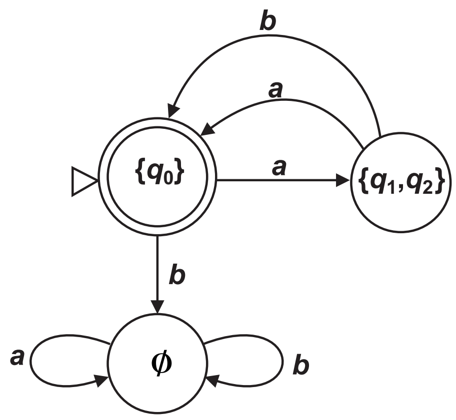

Рис. 3.9

Зазначимо, що з переходом до детермінованого скінченного автомата M_0 може суттєво зрости кількість станів: якщо автомат M має n станів, то M_0 має 2^n станів – підмножин множини Q. Проте деякі зі станів автомата M_0 можуть бути недосяжні з початкового стану $I_0 = \{I\}$, і їх можна вилучити (як це і було зроблено у прикладі 3.10). Таке вилучення не змінює мови $L[M_0] = L[M]$ (детальніше див. підрозд. 3.6.2).

Отже, для побудови таблиці переходів автомата M_0 доцільно обмежуватись станами, досяжними з початкового, використовуючи, наприклад, механізм черги: на початку роботи в чергу записуємо стан I_0; на кожному кроці дістаємо з черги стан $A \in 2^Q$ та записуємо у чергу всі стани, які досяжні з A деяким символом $\alpha \in T$ та які ще не містяться у таблиці переходів.

Приклад 3.11. Для недетермінованого скінченного автомата, граф якого зображено на рис. 3.10, відповідний детермінований автомат, побудований за методом із доведення теореми 3.1 з використанням механізму черги, має $2^3 = 8$ станів, з яких 5 досяжні з початкового стану $\{q_0, q_1\}$ (див. табл. 3.3). Вилучивши інші 3 стани, недосяжні з $\{q_0, q_1\}$, отримуємо детермінований скінченний автомат, граф якого зображено на рис. 3.11.

Таблиця 3.3

	$\{q_0, q_1\}$	$\{q_2\}$	$\varnothing$	$\{q_0\}$	$\{q_1\}$
a	$\{q_2\}$	$\{q_0\}$	$\varnothing$	$\{q_2\}$	$\{q_2\}$
b	$\varnothing$	$\{q_1\}$	$\varnothing$	$\varnothing$	$\varnothing$

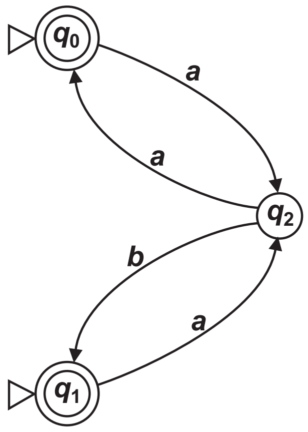

Рис. 3.10

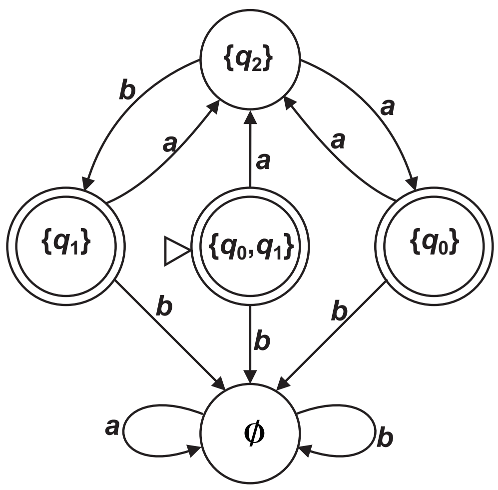

Рис. 3.11

Зазначимо, що наведені скінченні автомати також допускають мову $\{aa, ab\}^*$, тобто еквівалентні автоматам з прикладів 3.1 та 3.3.

Вправа 3.5. Для недетермінованого скінченного автомата, граф якого зображено на рис. 3.6, побудувати відповідний детермінований автомат за методом із доведення теореми 3.1.

3.3. Характеризація класу регулярних мов через скінченні автомати

Доведемо, що класи регулярних і автоматних мов збігаються.

Теорема 3.2. *Нехай G – регулярна граматика. Тоді існує скінченний автомат M, який допускає мову $L[G]$.*

Доведення. Нехай $G = \langle V, T, P, S\rangle$ – регулярна граматика, тобто множина P містить продукції вигляду $A \to a$, $A \to aB$, $A \to \varepsilon$, де $A, B \in V$, $a \in T$. Шуканий скінченний автомат $M = \langle Q, T, \Delta, I, F\rangle$ визначимо таким чином:

1) множина станів $Q = V \cup \{q_f\}$, де q_f – довільний фіксований символ, що не належить множині V;

2) множина Δ містить переходи (A, a, B) $(A, B \in V,\ a \in T)$, такі, що $(A \to aB) \in P$, та переходи (A, a, q_f) $(A \in V,\ a \in T)$, такі, що $(A \to a) \in P$;

3) множина $I = \{S\}$, тобто автомат M має один початковий стан – джерело S;

4) множина F містить стан q_f та всі такі стани $A \in V$, для яких $(A \to \varepsilon) \in P$.

Очевидно, що побудований автомат M допускає слово $w \in T^*$ тоді й тільки тоді, коли $S \overset{*}{\underset{G}{\Rightarrow}} w$, тобто $L[M] = L[G]$. Теорему доведено. □

Приклад 3.12. Розглянемо регулярну граматику $G = \langle V, T, P, S\rangle$, де $T = \{a, b\}$, $V = \{S, A\}$, $P = \{S \to aA|\varepsilon,\ A \to bS|a\}$. Застосувавши метод із доведення теореми 3.2, побудуємо скінченний автомат $M = \langle Q, T, \Delta, I, F\rangle$, такий, що $L[G] = L[M]$:

1) $Q = \{S, A, q_f\}$;
2) $\Delta = \{(S, a, A), (A, b, S), (A, a, q_f)\}$;
3) $I = \{S\}$;
4) $F = \{q_f, S\}$.

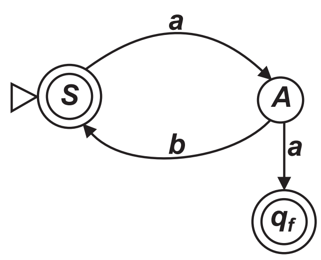

Рис. 3.12

Граф побудованого скінченного автомата зображено на рис. 3.12. Легко перевірити, що $L[M] = L[G] = \{(ab)^n, a(ba)^n a\colon n \geqslant 0\}$.

Приклад 3.13. Застосуємо метод із доведення теореми 3.2 до регулярної граматики $G = \langle V, T, P, S\rangle$, де $T = \{a, b\}$, $V = \{S, A\}$, $P = \{S \to aA|\varepsilon,\ A \to bS\}$.

Одразу зазначимо, що множина P не містить жодної продукції вигляду $X \to \xi$ ($X \in V$, $\xi \in T$), тобто множина переходів шуканого автомата не містить жодного переходу вигляду (X, ξ, q_f) ($X \in V$, $\xi \in T$), і стан q_f є недосяжним з початкового стану S. Видаливши непотрібний стан q_f, отримуємо такий автомат $\langle Q, T, \Delta, I, F\rangle$:

1) $Q = \{S, A\}$;
2) $\Delta = \{(S, a, A), (A, b, S)\}$;
3) $I = \{S\}$;
4) $F = \{S\}$.

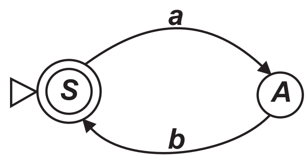

Рис. 3.13

Граф побудованого скінченного автомата зображено на рис. 3.13. Легко перевірити, що $L[M] = L[G] = \{(ab)^n : n \geqslant 0\}$.

Вправа 3.6. Побудувати скінченний автомат M, який допускає мову $L[G]$ для граматики G із прикладу 1.17, застосувавши метод із доведення теореми 3.2.

Теорема 3.3. *Нехай M – скінченний автомат. Тоді існує регулярна граматика G, яка породжує мову $L[M]$.*

Доведення. Нехай скінченний автомат $M = \langle Q, T, \Delta, I, F\rangle$ містить лише один початковий стан: $I = \{q_0\}$ (в інших випадках за теоремою 3.1 вводимо до розгляду детермінований скінченний автомат M_0, такий, що $L[M] = L[M_0]$). Визначимо шукану регулярну граматику $G = \langle V, T, P, S\rangle$:

1) $V = Q$;
2) $P = \{A \to aB : A \in Q, B \in Q, a \in T, (A, a, B) \in \Delta\} \cup \{A \to \varepsilon : A \in F\}$;
3) $S = q_0$.

Із побудови граматики G випливає, що вихідний автомат M допускає слово $w \in T^*$ тоді й тільки тоді, коли $S \overset{*}{\underset{G}{\Rightarrow}} w$, тобто $L[G] = L[M]$.

Теорему доведено. □

Приклад 3.14. Для скінченного автомата $M = \langle Q, T, \Delta, I, F\rangle$, побудованого у прикладі 3.12 (відповідний граф зображено на рис. 3.12), знайдемо регулярну граматику $G = \langle V, T, P, S\rangle$, таку, що $L[G] = L[M]$. Автомат має лише один початковий стан, а отже можемо одразу застосувати метод із доведення теореми 3.3:

1) $V = \{S, A, q_f\}$;
2) $P = \{S \to aA, A \to bS, A \to aq_f, S \to \varepsilon, q_f \to \varepsilon\}$.

Легко перевірити, що $L[G] = L[M] = \{(ab)^n, a(ba)^n a : n \geqslant 0\}$.

Вправа 3.7. Побудувати регулярну граматику G, що породжує мову $L[M]$ для скінченного автомата M, граф якого зображено на рис. 3.6, застосувавши метод із доведення теореми 3.3.

Вказівка. Використати результат вправи 3.5.

3.4. Скінченні автомати з ε-переходами

Автомат з ε-переходами є природним узагальненням скінченного автомата, визначеного в означенні 3.1.

Означення 3.6. Скінченним автоматом з ε-переходами називають впорядкований набір $\langle Q, T, \Delta, I, F\rangle$, де Q та T – скінченні непорожні множини, $\Delta \subset (Q \times (T \cup \{\varepsilon\}) \times Q)$, $I \subset Q$, $F \subset Q$. Множину Q називають множиною станів, T – алфавітом, Δ – відношенням (множиною) переходів, I – множиною початкових станів, F – множиною допускаючих станів. Перехід вигляду (q_1, ε, q_2) $(q_1, q_2 \in Q)$ називають ε-переходом.

Скінченний автомат, визначений в означенні 3.1, є частковим випадком скінченного автомата з ε-переходами.

Конфігурацією скінченного автомата $\langle Q, T, \Delta, I, F\rangle$ з ε-переходами називають (аналогічно означенню 3.3) довільний набір $(q, w) \in (Q \times T^*)$. На множині всіх конфігурацій скінченного автомата $M = \langle Q, T, \Delta, I, F\rangle$ з ε-переходами визначимо бінарне відношення «$\vdash_M$», яке назвемо *тактом роботи* автомата з ε-переходами:

$$((q_1, w) \vdash_M (q_2, u)) \Leftrightarrow \begin{cases} w = \xi u, \ \xi \in (T \cup \{\varepsilon\}); \\ (q_1, \xi, q_2) \in \Delta, \end{cases}$$

де $q_1, q_2 \in Q$, $w \in T^*$, $u \in T^*$. Інакше кажучи, відношення «$\vdash_M$» містить пари конфігурацій вигляду $((q_1, au), (q_2, u))$, якщо $(q_1, a, q_2) \in \Delta$, та $((q_1, u), (q_2, u))$, якщо $(q_1, \varepsilon, q_2) \in \Delta$.

Відношення «$\vdash^*_M$» – транзитивно-рефлексивне замикання відношення «$\vdash_M$», – складається, згідно з означенням 1.4, із таких пар конфігурацій (c_0, c_n), що

$$c_0 \vdash_M c_1 \vdash_M c_2 \cdots \vdash_M c_n$$

для деякої послідовності конфігурацій $c_1, c_2, \ldots, c_{n-1}$ $(n \geqslant 0)$. Індукцією за довжиною слова $w \in T^*$ легко вивести еквівалентність

$$((q_1, w) \vdash^*_M (q_2, \varepsilon)) \Leftrightarrow \big(\forall\, u \in T^* \colon (q_1, wu) \vdash^*_M (q_2, u)\big).$$

Надалі, якщо із контексту зрозуміло, що йдеться саме про скінченний автомат з ε-переходами M, замість «$\vdash_M$» та «$\vdash^*_M$» писатимемо відповідно «$\vdash$» та «$\vdash^*$».

Означення 3.7. Нехай $M = \langle Q, T, \Delta, I, F\rangle$ – скінченний автомат з ε-переходами. Кажуть, що автомат M допускає (сприймає) слово $w \in T^*$, якщо $(q_0, w) \vdash^*_M (q, \varepsilon)$ для деяких $q_0 \in I$ та $q \in F$. Множину слів $L[M]$, які допускає скінченний автомат з ε-переходами M, називають формальною мовою, яку допускає (сприймає) скінченний автомат з ε-переходами M. Скінченні автомати з ε-переходами M_1 та M_2 називають еквівалентними, якщо M_1 та M_2 задані над спільним алфавітом та допускають ту саму мову: $(M_1 \sim M_2) \Leftrightarrow (L[M_1] = L[M_2])$.

На скінченні автомати з ε-переходами поширюють техніку задання за допомогою графів, описану в підрозд. 3.1.3: для кожного ε-переходу (q_1, ε, q_2) до відповідного графа включають ребро з міткою ε, що веде від вершини з міткою q_1 до вершини з міткою q_2.

Приклад 3.15. На рис. 3.14 зображено граф скінченного автомата з ε-переходами, що допускає формальну мову $\{a^n b^m c^k \colon n \geqslant 0, m \geqslant 0, k \geqslant 0\}$.

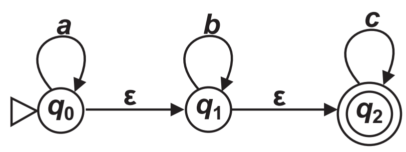

Рис. 3.14

Наведений граф відповідає скінченному автомату над алфавітом $T = \{a, b, c\}$ із множиною станів $Q = \{q_0, q_1, q_2\}$, множинами початкових та допускаючих станів $I = \{q_0\}$ та $F = \{q_2\}$ відповідно, і множиною переходів Δ, що містить три «класичні» переходи (q_0, a, q_0), (q_1, b, q_1) і (q_2, c, q_2), та два ε-переходи (q_0, ε, q_1) і (q_1, ε, q_2). Легко перевірити, що цей автомат справді допускає ті й тільки ті слова з $\{a, b, c\}^*$, які мають вигляд $a^n b^m c^k$, де n, m, k – цілі невід'ємні числа. Так, для вхідного слова $ac^2 = a^1 b^0 c^2$ можлива послідовність конфігурацій

$$(q_0, ac^2) \vdash (q_0, c^2) \vdash (q_1, c^2) \vdash (q_2, c^2) \vdash (q_2, c) \vdash (q_2, \varepsilon),$$

де другий та третій такти відповідають ε-переходам (q_0, ε, q_1) та (q_1, ε, q_2).

Приклад 3.16. На рис. 3.15 зображено граф скінченного автомата над алфавітом $T = \{0, 1, +, -\}$, що допускає формальну мову L, яка складається із двійкових записів цілих чисел (можливо, з провідними нулями):

$$L = \{sw \colon w \in \{0,1\}^+, s \in \{+, -, \varepsilon\}\} = \{+, -, \varepsilon\}\{0,1\}^+.$$

Наведений граф відповідає скінченному автомату із множиною станів $Q = \{q_0, q_1, q_2\}$, множинами початкових та допускаючих станів $I = \{q_0\}$ та $F = \{q_2\}$ відповідно, і множиною переходів Δ, яка містить сім переходів, серед яких один ε-перехід (q_0, ε, q_1).

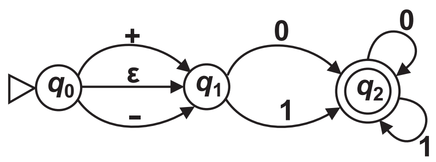

Рис. 3.15

Теорема 3.4. *Довільний скінченний автомат M з ε-переходами еквівалентний деякому скінченному автомату M' без ε-переходів.*

Доведення. Множину станів та множину початкових станів шуканого автомата $M' = \langle Q', T, \Delta', I', F' \rangle$ залишимо такими ж, як у заданому автоматі $M = \langle Q, T, \Delta, I, F \rangle$, тобто покладемо $Q' = Q$, $I' = I$. Далі за допомогою рекурсії визначимо послідовність множин F_i $(i \geqslant 0)$:

1) $F_0 = F$;

2) $F_{i+1} = F_i \cup \{q \in Q \colon \exists\, p \in F_i \colon (q, \varepsilon, p) \in \Delta\}$.

Легко зрозуміти, що вихідний автомат M еквівалентний кожному з автоматів $\langle Q', T, \Delta, I', F_i \rangle$ $(i \geqslant 0)$, оскільки, якщо $(q, \varepsilon, q_1) \in \Delta$, $q_0 \in I'$ і $(q_0, w) \vdash_M^* (q, \varepsilon)$, то $(q_0, w) \vdash_M^* (q_1, \varepsilon)$. Оскільки $F_i \subset F_{i+1} \subset Q$, а множина Q скінченна, має існувати $m \geqslant 0$, таке, що $F_{m+1} = F_m$; очевидно, що $F_{m+k} = F_m$ для кожного $k \geqslant 0$. Множину допускаючих станів шуканого автомата визначимо як «найширшу» із множин F_i $(i \geqslant 0)$, тобто $F' = F_m$, де $m = \min\{i \geqslant 0 \colon F_i = F_{i+1}\}$.

Далі рекурсивно визначимо послідовність множин Δ_k $(k \geqslant 0)$:

1) $\Delta_0 = \Delta$;

2) $\Delta_{k+1} = \Delta_k \cup \left\{ (q_1, a, q_2) \in (Q \times T \times Q) \colon \exists\, q \in Q \colon \begin{cases} (q_1, \varepsilon, q) \in \Delta_k, \\ (q, a, q_2) \in \Delta_k \end{cases} \right\}$.

Зазначимо, що еквівалентність автомата M та кожного з автоматів $\langle Q', T, \Delta_k, I', F' \rangle$ $(k \geqslant 0)$ легко довести індукцією за номером $k \geqslant 0$.

Оскільки $\Delta_k \subset \Delta_{k+1} \subset (Q \times T \times Q)$ $(k \geqslant 0)$, а множина $Q \times T \times Q$ скінченна, має існувати $n \geqslant 0$, таке, що $\Delta_{n+1} = \Delta_n$. Множину переходів шуканого автомата отримуємо, видаливши ε-переходи із множини Δ_n:

$$\Delta' = \Delta_n \setminus \{(p, \varepsilon, q) \colon p, q \in Q\}.$$

Легко зрозуміти, що видалення ε-переходів із множини Δ_n не звужує множину слів, які допускає автомат $\langle Q', T, \Delta_n, I', F' \rangle$. $\square$

Приклад 3.17. Позбудемося ε-переходів в автоматі з прикладу 3.15 (рис. 3.14), застосовуючи метод із доведення теореми 3.4. Спочатку зазначимо, що множина всіх станів і множина початкових станів не змінюються: $Q' = Q = \{q_0, q_1, q_2\}$, $I' = I = \{q_0\}$. Множину допускаючих станів сформуємо рекурсивно, додаючи до множини $F = \{q_2\}$ нові стани (згідно з наявними ε-переходами):

1) $F_0 = F = \{q_2\}$;

2) $F_1 = F_0 \cup \{q \in Q : \exists p \in F_0 : (q, \varepsilon, p) \in \Delta\} = \{q_2, q_1\}$, оскільки $(q_1, \varepsilon, q_2) \in \Delta$ та $q_2 \in F_0$;

3) $F_2 = F_1 \cup \{q \in Q : \exists p \in F_1 : (q, \varepsilon, p) \in \Delta\} = \{q_2, q_1, q_0\}$, оскільки $(q_0, \varepsilon, q_1) \in \Delta$ та $q_1 \in F_1$;

4) $F_3 = F_2 \cup \{q \in Q \colon \exists p \in F_2 \colon (q, \varepsilon, p) \in \Delta\} = F_2$.

Отже, $F' = F_2$.

Сформуємо послідовність множин Δ_k ($k \geqslant 0$), додаючи до вихідної множини переходів Δ нові переходи:

1) $\Delta_0 = \Delta$;

2) $\Delta_1 = \Delta_0 \cup \{(q_0, b, q_1), (q_1, c, q_2)\}$; додано перехід (q_0, b, q_1), оскільки $(q_0, \varepsilon, q_1) \in \Delta_0$, $(q_1, b, q_1) \in \Delta_0$; додано перехід (q_1, c, q_2), оскільки $(q_1, \varepsilon, q_2) \in \Delta_0$, $(q_2, c, q_2) \in \Delta_0$;

3) $\Delta_2 = \Delta_1 \cup \{(q_0, c, q_2)\}$, оскільки $(q_0, \varepsilon, q_1) \in \Delta_1$, $(q_1, c, q_2) \in \Delta_1$;

4) $\Delta_3 = \Delta_2$.

Отже, $\Delta_k = \Delta_2$ для всіх $k \geqslant 2$.

Нарешті, видаливши з Δ_2 обидва ε-переходи, отримуємо множину переходів Δ' шуканого автомата без ε-переходів. Побудований автомат (див. рис. 3.16) еквівалентний вихідному, тобто обидва допускають ту ж саму мову $\{a^n b^m c^k \colon n \geqslant 0, m \geqslant 0, k \geqslant 0\}$.

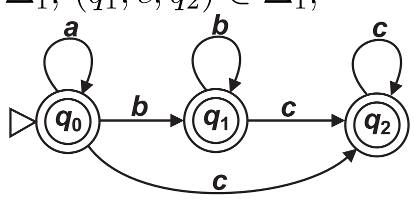

Рис. 3.16

Приклад 3.18. Позбудемося ε-переходів в автоматі із прикладу 3.16 (рис. 3.15), застосовуючи метод із доведення теореми 3.4. Видаленню єдиного ε-переходу (q_0, ε, q_1) передує додавання переходів $(q_0, 0, q_2)$ та $(q_0, 1, q_2)$; множина допускаючих станів не змінюється, оскільки $q_1 \notin F$. Отриманий скінченний автомат (див. рис. 3.17) еквівалентний вихідному, тобто допускає ту ж саму формальну мову

$$\{sw \colon w \in \{0,1\}^+, s \in \{+, -, \varepsilon\}\}.$$

Зауваження 3.7. Для видалення ε-переходів можна застосувати метод, у певному сенсі симетричний до описаного у доведенні теореми 3.4. Для заданого скінченного автомата $M = \langle Q, T, \Delta, I, F\rangle$ з ε-переходами покладемо $Q' = Q$, $F' = F$. За допомогою рекурсії визначимо послідовність множин I_i $(i \geqslant 0)$:

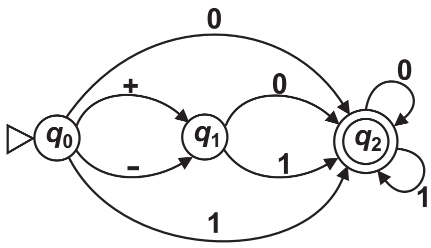

Рис. 3.17

1) $I_0 = I$;

2) $I_{i+1} = I_i \cup \{p \in Q \colon \exists\, q \in I_i \colon (q, \varepsilon, p) \in \Delta\}$,

та покладемо $I' = I_m$, де $m = \min\{i \geqslant 0 : I_i = I_{i+1}\}$. Далі рекурсивно визначимо послідовність множин Δ_k $(k \geqslant 0)$:

1) $\Delta_0 = \Delta$;

2) $\Delta_{k+1} = \Delta_k \cup \left\{ (q_1, a, q_2) \in (Q \times T \times Q) \colon \exists\, q \in Q \colon \begin{cases} (q_1, a, q) \in \Delta_k, \\ (q, \varepsilon, q_2) \in \Delta_k \end{cases} \right\}$.

Покладемо $\Delta' = \Delta_n \setminus \{(p, \varepsilon, q) \colon p, q \in Q\}$, де $n = \min\{k \geqslant 0 \colon \Delta_k = \Delta_{k+1}\}$. Отриманий скінченний автомат $M' = \langle Q', T, \Delta', I', F'\rangle$ еквівалентний заданому автомату M та не містить ε-переходів,

Приклад 3.19. Позбудемося ε-переходів в автоматі з прикладу 3.16 (рис. 3.15), застосовуючи метод, запропонований у зауваженні 3.7. Видаленню єдиного ε-переходу (q_0, ε, q_1) передує лише додавання початкового стану q_1, оскільки $q_0 \in I$ та $(q_0, \varepsilon, q_1) \in \Delta$, і множина переходів Δ не містить жодного переходу вигляду (p, a, q_0) $(p \in Q$, $a \in T)$. Отриманий скінченний автомат (рис. 3.18) еквівалентний вихідному автомату, тобто допускає ту ж саму формальну мову $\{sw \colon w \in \{0,1\}^+, s \in \{+, -, \varepsilon\}\}$.

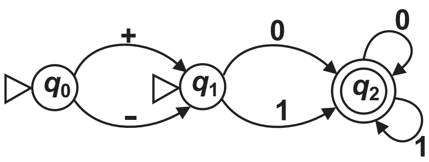

Рис. 3.18

Вправа 3.8. Позбутися ε-переходів в автоматі з прикладу 3.15, застосовуючи метод, запропонований у зауваженні 3.7.

Вправа 3.9. Позбутися ε-переходів в автоматі, зображеному на рис. 3.19, застосовуючи як метод із доведення теореми 3.4, так і метод, запропонований у зауваженні 3.7. Порівняти результати.

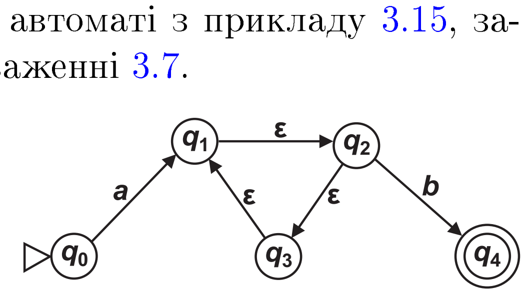

Рис. 3.19

3.5. Мінімізація детермінованих скінченних автоматів: теорема Майхілла–Нерода

У цьому підрозділі розглянемо проблему існування мінімального за кількістю станів детермінованого скінченного автомата, який допускає задану формальну мову.

3.5.1. Еквівалентність слів відносно мови

Нехай L – формальна мова над алфавітом T. Слово $\alpha \in T^*$ називають *правим контекстом* слова $w \in T^*$ відносно мови L, якщо $w\alpha \in L$. Множину всіх правих контекстів слова $w \in T^*$ відносно L позначають через $C_L^{(r)}(w)$.

Приклад 3.20. Нехай $T = \{a, b\}$, $L = \{a^n b^m : n, m \geqslant 0\}$. Тоді маємо такі множини правих контекстів:

1) $C_L^{(r)}(a^n) = \{a^i b^j : i \geqslant 0,\ j \geqslant 0\} = L$ для всіх $n \geqslant 0$;
2) $C_L^{(r)}(a^n b^m) = \{b^j : j \geqslant 0\}$ для всіх $n \geqslant 0$, $m \geqslant 1$;
3) $C_L^{(r)}(w) = \varnothing$, якщо w містить підслово ba.

Приклад 3.21. Нехай $T = \{a, b\}$, $L = \{a^n b^m : n \geqslant 1,\ m \geqslant 0\}$. Тоді маємо такі множини правих контекстів:

1) $C_L^{(r)}(\varepsilon) = L$;
2) $C_L^{(r)}(a^n) = \{a^i b^j : i \geqslant 0,\ j \geqslant 0\}$ для всіх $n \geqslant 1$;
3) $C_L^{(r)}(a^n b^m) = \{b^j : j \geqslant 0\}$ для всіх $n \geqslant 1$, $m \geqslant 1$;
4) $C_L^{(r)}(b^m) = \varnothing$ для всіх $m \geqslant 1$;
5) $C_L^{(r)}(w) = \varnothing$, якщо w містить підслово ba.

Приклад 3.22. Нехай $T = \{a, b, c\}$, $L = \{a^n c b^m : n, m \geqslant 0\}$. Тоді маємо такі множини правих контекстів:

1) $C_L^{(r)}(a^n) = L$ для всіх $n \geqslant 0$;
2) $C_L^{(r)}(a^n c b^m) = \{b^j : j \geqslant 0\}$ для всіх $n \geqslant 0$, $m \geqslant 0$;
3) $C_L^{(r)}(w) = \varnothing$, якщо $w \notin \{a^i, a^i c b^j : i, j \geqslant 0\}$.

Приклад 3.23. Нехай $T = \{a, b\}$, $L = \{a^n b^n : n \geqslant 0\}$. Тоді маємо такі множини правих контекстів:

1) $C_L^{(r)}(\varepsilon) = L$;
2) $C_L^{(r)}(a^n) = \{a^i b^{i+n} : i \geqslant 0\}$ для всіх $n \geqslant 1$;
3) $C_L^{(r)}(a^n b^m) = \{b^{n-m}\}$ для всіх $n \geqslant m \geqslant 1$;
4) $C_L^{(r)}(a^n b^m) = \varnothing$ для всіх $m > n$;
5) $C_L^{(r)}(w) = \varnothing$, якщо w містить підслово ba.

Зазначимо, що в цьому випадку маємо нескінченну сукупність різних множин правих контекстів (на відміну від прикладів 3.20–3.22).

Слова $w_1, w_2 \in T^*$ називають *еквівалентними відносно формальної мови* L, якщо вони мають однакові множини правих контекстів:

$$(w_1 \underset{L}{\sim} w_2) \Leftrightarrow (C_L^{(r)}(w_1) = C_L^{(r)}(w_2)).$$

Інакше кажучи, слова $w_1, w_2 \in T^*$ є еквівалентними відносно формальної мови L, якщо для будь-якого $\alpha \in T^*$ обидва слова $w_1\alpha$ та $w_2\alpha$ одночасно належать L або обидва одночасно не належать L:

$$(w_1 \underset{L}{\sim} w_2) \Leftrightarrow (\forall\, \alpha \in T^* : (w_1\alpha \in L) \leftrightarrow (w_2\alpha \in L)).$$

Очевидно, що « $\underset{L}{\sim}$ » є відношенням еквівалентності на множині T^*, і множи́ни слів з однаковими правими контекстами є відповідними класами еквівалентності, тобто утворюють фактор-множину:

$$T^*/_{\underset{L}{\sim}} = \{[w]_L : w \in T^*\}, \text{ де } [w]_L = \{u \in T^* : u \underset{L}{\sim} w\}.$$

Приклад 3.24. Нехай $T = \{a, b\}$, $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$. Тоді згідно з результатом прикладу 3.20 отримуємо:

$$\{a,b\}^*/_{\underset{L}{\sim}} = \{\{u \in \{a,b\}^* : C_L^{(r)}(u) = L\},$$
$$\{u \in \{a,b\}^* : C_L^{(r)}(u) = \{b^j : j \geqslant 0\}\}, \{u \in \{a,b\}^* : C_L^{(r)}(u) = \varnothing\}\} =$$
$$= \{\{a^n : n \geqslant 0\}, \{a^n b^m : n \geqslant 0, m \geqslant 1\}, \{u \in \{a,b\}^* : |u|_{ba} \geqslant 1\}\} =$$
$$= \{[\varepsilon]_L, [b]_L, [ba]_L\}.$$

Вправа 3.10. Довести наслідок: якщо $w_1 \underset{L}{\sim} w_2$ та $u \in T^*$, то $w_1 u \underset{L}{\sim} w_2 u$. Чи справджується зворотний наслідок?

Вправа 3.11. Довести наслідок: якщо $w_1 \underset{L}{\sim} w_2$ і $w_1 \in L$, то $w_2 \in L$.

Зауваження 3.8. Зазначимо, що між сукупністю можливих множин правих контекстів $\{C_L^{(r)}(u) \colon u \in T^*\}$ і фактор-множиною $T^*/\underset{L}{\sim}$ (множиною класів еквівалентності відносно мови L) існує взаємно однозначна відповідність:

$$C_L^{(r)}(u) \leftrightsquigarrow [u]_L = \{w \in T^* \colon w \underset{L}{\sim} u\} = \{w \in T^* \colon C_L^{(r)}(w) = C_L^{(r)}(u)\},$$

де $u \in T^*$. Так, для формальної мови $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$ (див. приклади 3.20, 3.24) отримуємо відповідність

$$C_L^{(r)}(\varepsilon) = L \leftrightsquigarrow [\varepsilon]_L = \{a^n \colon n \geqslant 0\};$$
$$C_L^{(r)}(b) = \{b^j \colon j \geqslant 0\} \leftrightsquigarrow [b]_L = \{a^n b^m \colon n \geqslant 0, m \geqslant 1\};$$
$$C_L^{(r)}(ba) = \varnothing \leftrightsquigarrow [ba]_L = \{u \in \{a, b\}^* \colon |u|_{ba} \geqslant 1\}.$$

Вправа 3.12. Виписати фактор-множину $T^*/\underset{L}{\sim}$ та вказану відповідність для прикладів 3.21–3.23.

3.5.2. Еквівалентність слів відносно автомата

Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат, який допускає регулярну мову L. Слова $w_1, w_2 \in T^*$ називають *еквівалентними відносно автомата* M, якщо $\delta^*(q_0, w_1) = \delta^*(q_0, w_2)$, де $\delta^* : (Q \times T^*) \to Q$ – розширена функція переходів автомата M (див. підрозд. 3.2.1). Еквівалентність слів $w_1, w_2 \in T^*$ відносно скінченного автомата M позначатимемо як $w_1 \underset{M}{\sim} w_2$:

$$(w_1 \underset{M}{\sim} w_2) \Leftrightarrow (\delta^*(q_0, w_1) = \delta^*(q_0, w_2)).$$

Легко перевірити, що «$\underset{M}{\sim}$» є відношенням еквівалентності на T^*.

Приклад 3.25. Детермінований скінченний автомат M, зображений на рис. 3.20, допускає мову $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$ над алфавітом $T = \{a, b\}$. Для розширеної функції переходів отримуємо:

$$\delta^*(q_0, w) = \begin{cases} q_0, \text{ якщо } w = \varepsilon, \\ q_1, \text{ якщо } w = a^n, n \geqslant 1, \\ q_2, \text{ якщо } |w|_{ba} = 0, \ |w|_b \geqslant 1, \\ q_3, \text{ якщо } |w|_{ba} \geqslant 1. \end{cases}$$

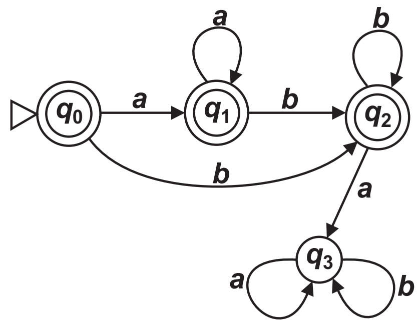

Рис. 3.20

Отже, маємо такі класи еквівалентності за відношенням « $\underset{M}{\sim}$ »:

$$[\varepsilon]_M = \{\varepsilon\},$$
$$[a]_M = \{a^n : n \geqslant 1\},$$
$$[b]_M = \{a^n b^m : n \geqslant 0, m \geqslant 1\},$$
$$[ba]_M = \{u \in T^* : |u|_{ba} \geqslant 1\}.$$

Вправа 3.13. Довести наслідок: якщо $w_1 \underset{M}{\sim} w_2$ та $u \in T^*$, то $w_1 u \underset{M}{\sim} w_2 u$.

Зауваження 3.9. Зазначимо, що між множиною станів, досяжних з q_0, і фактор-множиною $T^*/_{\underset{M}{\sim}}$ (множиною класів еквівалентності відносно автомата M) існує взаємно однозначна відповідність:

$$q = \delta^*(q_0, w) \leftrightsquigarrow [w]_M,$$

де $w \in T^*$. Так, для формальної мови $L = \{a^n b^m : n \geqslant 0, \ m \geqslant 0\}$, яку допускає автомат M, зображений на рис. 3.20 (див. приклади 3.20, 3.25), отримуємо відповідність

$$q_0 = \delta^*(q_0, \varepsilon) \leftrightsquigarrow [\varepsilon]_M = \{\varepsilon\};$$
$$q_1 = \delta^*(q_0, a) \leftrightsquigarrow [a]_M = \{a^n : n \geqslant 1\};$$
$$q_2 = \delta^*(q_0, b) \leftrightsquigarrow [b]_M = \{a^n b^m : n \geqslant 0, m \geqslant 1\};$$
$$q_3 = \delta^*(q_0, ba) \leftrightsquigarrow [ba]_M = \{u \in T^* : |u|_{ba} \geqslant 1\}.$$

3.5.3. Зв'язок між відношеннями еквівалентності на T^* за мовою і за скінченним автоматом

Нехай L – регулярна мова над алфавітом T, що допускається детермінованим скінченним автоматом $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$. Сформулюємо у вигляді леми простий зв'язок між відношеннями еквівалентності за формальною мовою L і за автоматом M.

Лема 3.1. *Нехай $w_1, w_2 \in T^*$. Тоді справджується логічний наслідок*

$$(w_1 \underset{M}{\sim} w_2) \Rightarrow (w_1 \underset{L}{\sim} w_2).$$

Доведення. Нехай $w_1 \underset{M}{\sim} w_2$, тобто $\delta^*(q_0, w_1) = \delta^*(q_0, w_2)$. Тоді для довільного фіксованого $\alpha \in T^*$, використовуючи властивість функції δ^* із вправи 3.1 (п. 3), отримуємо:

$$\delta^*(q_0, w_1\alpha) = \delta^*(\delta^*(q_0, w_1), \alpha) = \delta^*(\delta^*(q_0, w_2), \alpha) = \delta^*(q_0, w_2\alpha).$$

Отже, обидва слова $w_1\alpha$ та $w_2\alpha$ або одночасно належать L, або обидва одночасно не належать L. Враховуючи довільність вибору $\alpha \in T^*$, отримуємо еквівалентність $w_1 \underset{L}{\sim} w_2$, що й треба було довести. ☐

Нехай «$\sim$» – відношення еквівалентності на множині X. *Індексом $i_\sim$ відношення «$\sim$»* називають кількість класів еквівалентності за відношенням «$\sim$»:

$$i_\sim = \left| X/_\sim \right|.$$

Якщо X нескінченна, індекс $i_\sim$ може бути як скінченним ($i_\sim < \infty$), так і нескінченним ($i_\sim = \infty$). Якщо множина X скінченна, індекс $i_\sim$ є скінченним.

Індекси відношень « $\underset{L}{\sim}$ » та « $\underset{M}{\sim}$ » позначатимемо як i_L та i_M відповідно. Зі скінченності множини станів Q випливає, що $i_M < \infty$.

Лема 3.2. *Нехай L – регулярна мова над алфавітом T, що допускається детермінованим скінченним автоматом M. Тоді $i_L \leqslant i_M$.*

Доведення. Нехай $w \in T^*$. Якщо $u \underset{L}{\sim} w$, то $[u]_M \subset [u]_L = [w]_L$ за лемою 3.1, звідки маємо вкладення $\bigcup\limits_{u \underset{L}{\sim} w} [u]_M \subset [w]_L$. Між тим, якщо $u \in [w]_L$ ($u \underset{L}{\sim} w$), то, оскільки $u \in [u]_M$ (для будь-якого $u \in T^*$), отримуємо належність $u \in \bigcup\limits_{v \underset{L}{\sim} w} [v]_M$, тобто маємо зворотне вкладення $[w]_L \subset \bigcup\limits_{u \underset{L}{\sim} w} [u]_M$. Отже, отримано рівність $[w]_L = \bigcup\limits_{u \underset{L}{\sim} w} [u]_M$, де об'єднання скінченне, оскільки $i_M < \infty$, а $[v_1]_L \cap [v_2]_L = \varnothing$ за $v_1 \underset{L}{\not\sim} v_2$ (нагадаємо, що різні класи за фіксованим відношенням еквівалентності попарно не перетинаються). Таким чином, кожний клас еквівалентності за відношенням « $\underset{L}{\sim}$ » містить (як підмножину) принаймні один клас еквівалентності за відношенням « $\underset{M}{\sim}$ ». Тому класів еквівалентності за відношенням « $\underset{M}{\sim}$ » не менше ніж класів еквівалентності за відношенням « $\underset{L}{\sim}$ », тобто $i_L \leqslant i_M$. □

Наслідок. *Нехай L – регулярна мова над алфавітом T. Тоді $i_L < \infty$.*

Приклад 3.26. Для формальної мови $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$ над алфавітом $T = \{a, b\}$, яку допускає детермінований скінченний автомат, зображений на рис. 3.20 (приклади 3.24 та 3.25), отримуємо такий зв'язок між класами еквівалентності за відношеннями « $\underset{L}{\sim}$ » та « $\underset{M}{\sim}$ »:

$$[\varepsilon]_L = \{a^n : n \geqslant 0\} = [\varepsilon]_M \cup [a]_M = \{\varepsilon\} \cup \{a^n : n \geqslant 1\};$$
$$[b]_L = \{a^n b^m : n \geqslant 0, m \geqslant 1\} = [b]_M;$$
$$[ba]_L = \{u \in \{a, b\}^* : |u|_{ba} \geqslant 1\} = [ba]_M.$$

Бачимо, що $i_L = 3 < i_M = 4$, що відповідає твердженню леми 3.2.

Приклад 3.27. Детермінований скінченний автомат M, зображений на рис. 3.21, як і автомат на рис. 3.20, також допускає формальну мову $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$ над алфавітом $T = \{a, b\}$. Легко перевірити, що класи еквівалентності за відношеннями « $\underset{L}{\sim}$ » та « $\underset{M}{\sim}$ » збігаються:

$$[\varepsilon]_L = [\varepsilon]_M = \{a^n : n \geqslant 0\};$$
$$[b]_L = [b]_M = \{a^n b^m : n \geqslant 0, m \geqslant 1\};$$
$$[ba]_L = [ba]_M = \{u \in T^* : |u|_{ba} \geqslant 1\}.$$

Бачимо, що $i_L = i_M = 3$, що відповідає твердженню леми 3.2.

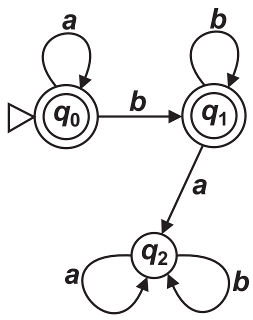

Рис. 3.21

Приклад 3.28. Для формальної мови $L = \{a^n b^n : n \geqslant 0\}$ над алфавітом $T = \{a, b\}$ відношення « $\underset{L}{\sim}$ » допускає нескінченну кількість класів еквівалентності (див. приклад 3.23), тобто $i_L = \infty$. Отже, за наслідком із леми 3.2, мова L нерегулярна.

Якщо формальну мову L допускає детермінований скінченний автомат $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$, то кількість станів $|Q|$ не менша за кількість класів еквівалентності $i_M = \left|T^*/\underset{M}{\sim}\right|$ (див. зауваження 3.9), тобто

$|Q| \geqslant i_{M}$. Враховуючи результат леми 3.2, отримуємо ланцюжок нерівностей

$$|Q| \geqslant i_{M} \geqslant i_{L}. \tag{3.5}$$

Зазначимо, що у випадку, коли всі стани автомата M досяжні з q_0, співвідношення (3.5) набувають вигляду $|Q| = i_{M} \geqslant i_{L}$.

Нерівності (3.5) показують, що будь-який детермінований автомат, який допускає мову L, має не менш ніж i_{L} станів. Приклади 3.26 та 3.27 демонструють, що рівність $|Q| = i_{L}$ може, але не зобов'язана досягатись.

3.5.4. Теорема Майхілла–Нерода

Теорема 3.5 (Майхілл[1], Нерод[2], 1958 р.)**.** *Для будь-якої регулярної мови L над алфавітом T існує скінченний детермінований автомат $M_L = \langle Q_L, T, \Delta_L, \{q_{0,L}\}, F_L\rangle$ з i_L станами, який допускає мову L.*

Доведення. Побудуємо автомат M_L безпосередньо. Покладемо

$$\begin{aligned}
&Q_L = T^*/\underset{L}{\sim}\,;\\
&\Delta_L = \{([w]_L, a, [wa]_L) \colon w \in T^*,\ a \in T\};\\
&q_{0,L} = [\varepsilon]_L;\\
&F_L = \{[w]_L \colon w \in L\}.
\end{aligned} \tag{3.6}$$

Коректність визначення множини F_L випливає з результату вправи 3.11: якщо $[w_1]_L = [w_2]_L$ (тобто $w_1 \underset{L}{\sim} w_2$) і $w_1 \in L$, то $w_2 \in L$; таким чином, належність чи неналежність стану $[w]_L$ до F_L не залежить від вибору представника $u \in [w]_L$ ($[u]_L = [w]_L$).

Доведемо, що побудований автомат детермінований. Нехай $w \in T^*$, $a \in T$. Тоді існує принаймні один стан $[u]_L$ ($u \in T^*$), такий, що $([w]_L, a, [u]_L) \in \Delta_L$: достатньо взяти $[u]_L = [wa]_L$. Припустивши, що існує ще один стан $[v]_L$ ($v \in T^*$), такий, що $([w]_L, a, [v]_L) \in \Delta_L$, за визначенням відношення Δ_L отримуємо: $\exists\, w_1 \in [w]_L : v = w_1 a$. Оскільки

[1]Майхілл Джон (1923-1987) – американський вчений; отримав важливі результати в різних розділах математики, зокрема в теорії формальних мов, теорії обчислень та математичній логіці.

[2]Нерод (Нероуд) Еніл (народ. у 1932 р.) – американський вчений; автор численних робіт з математичної логіки, теорії автоматів тощо.

$w_1 \underset{L}{\sim} w$, за результатом вправи 3.10 отримуємо: $v = w_1 a \underset{L}{\sim} wa$, тобто $[v]_L = [wa]_L$. Таким чином, існує лише один стан $[u]_L = [wa]_L$, такий, що $([w]_L, a, [u]_L) \in \Delta_L$, тобто автомат M_L детермінований.

Відношення Δ_L визначає функцію переходів $\delta_L\colon (Q_L \times T) \to Q_L$ (див. еквівалентність (3.2)), яка для побудованого детермінованого автомата має вигляд $\delta_L([w]_L, a) = [wa]_L$, де $w \in T^*$, $a \in T$.

Розширена функція переходів $\delta_L^*\colon (Q_L \times T^*) \to Q_L$ для побудованого автомата має вигляд $\delta_L^*([w]_L, u) = [wu]_L$, де $w \in T^*$, $u \in T^*$. Справді, рекурентні співвідношення (3.3) для функції δ_L^* перевіряються безпосередньо:

$$\delta_L^*([u]_L, \varepsilon) = [u\varepsilon]_L = [u]_L, \quad \delta_L^*([u]_L, wa) = [uwa]_L = \delta_L(\delta_L^*([u]_L, w), a).$$

Доведемо, що побудований автомат M_L справді допускає мову L. За еквівалентністю (3.4), автомат M_L допускає слово $u \in T^*$ тоді й тільки тоді, коли

$$\delta_L^*([\varepsilon]_L, u) = [u]_L \in F_L = \{[w]_L\colon w \in L\},$$

тобто, враховуючи результат вправи 3.11, автомат M_L допускає u тоді й тільки тоді, коли $u \in L$.

Таким чином, побудований автомат M_L є детермінованим, містить $i_L = \left|T^*/\underset{L}{\sim}\right|$ станів і допускає задану мову L. Теорему доведено. □

Скінченний детермінований автомат, що допускає регулярну мову L і містить i_L (мінімально можливу кількість) станів, називають *мінімальним скінченним детермінованим автоматом, що допускає* L.

Теорема 3.5 не тільки стверджує, що будь-яку регулярну мову L допускає деякий мінімальний детермінований скінченний автомат M, але й надає конкретний спосіб побудови автомата M (якщо відомі всі класи еквівалентності за відношенням «$\underset{L}{\sim}$»).

Приклад 3.29. Для формальної мови $L = \{a^n b^m\colon n \geqslant 0,\ m \geqslant 0\}$ над алфавітом $T = \{a, b\}$, використовуючи побудовані класи еквівалентності за відношенням «$\underset{L}{\sim}$» (див. приклад 3.24), отримуємо такий детермінований автомат, мінімальний за кількістю станів (див. рис. 3.22):

$$Q_L = \{a, b\}^* / \underset{L}{\sim} = \{[\varepsilon]_L, [b]_L, [ba]_L\};$$
$$\Delta_L = \{([w]_L, a, [wa]_L) \colon w \in T^*,\ a \in T\} =$$
$$= \{([\varepsilon]_L, a, [a]_L), ([\varepsilon]_L, b, [b]_L), ([b]_L, a, [ba]_L),$$
$$([b]_L, b, [b^2]_L), ([ba]_L, a, [ba^2]_L), ([ba]_L, b, [bab]_L)\} =$$
$$= \{([\varepsilon]_L, a, [\varepsilon]_L), ([\varepsilon]_L, b, [b]_L), ([b]_L, a, [ba]_L),$$
$$([b]_L, b, [b]_L), ([ba]_L, a, [ba]_L), ([ba]_L, b, [ba]_L)\};$$
$$q_{0,L} = [\varepsilon]_L;$$
$$F_L = \{[w]_L \colon w \in L\} = \{[\varepsilon]_L, [b]_L\}.$$

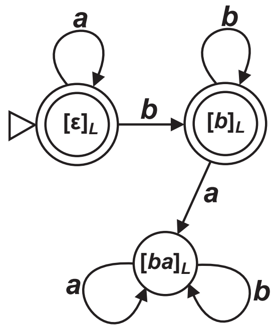

Рис. 3.22

Зазначимо, що отриманий автомат з точністю до перейменування станів збігається з автоматом, зображеним на рис. 3.21.

Під час доведення теореми 3.5 використовувалася скінченність i_L, однак припущення щодо регулярності мови L не використовувалося; для побудови мінімального автомата (3.6) (мінімального детермінованого автомата, що допускає L) необхідною є лише умова $i_L < \infty$.

Таким чином, будь-яку формальну мову L зі скінченним індексом i_L допускає скінченний (навіть мінімальний) детермінований автомат, а отже мова L зі скінченним індексом i_L є регулярною. Враховуючи наслідок із леми 3.2, отримуємо твердження, яке є критерієм регулярності формальної мови.

Теорема 3.6 (наслідок із теореми Майхілла–Нерода)**.** *Формальна мова L є регулярною тоді й тільки тоді, коли $i_L < \infty$.*

Зауваження 3.10. У літературі саме теорему 3.6 іноді називають теоремою Майхілла–Нерода (див., наприклад, [11]).

Вправа 3.14. Для формальних мов із прикладів 3.21–3.22 побудувати мінімальний автомат, застосовуючи метод із доведення теореми 3.5.

3.6. Мінімізація детермінованих скінченних автоматів: єдиність мінімального детермінованого автомата

У цьому підрозділі доведемо єдиність (з точністю до ізоморфізму) мінімального детермінованого автомата, існування якого постулює теорема Майхілла–Нерода.

3.6.1. Еквівалентність станів детермінованого автомата

Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат, $\delta^* : (Q \times T^*) \to Q$ – розширена функція переходів автомата M (див. підрозд. 3.2.1).

Кажуть, що стани $q_1, q_2 \in Q$ розрізняються словом $w \in T^*$, якщо справджується висловлення

$$(\delta^*(q_1, w) \in F) \oplus (\delta^*(q_2, w) \in F),$$

тобто із двох станів $\delta^*(q_1, w)$ та $\delta^*(q_2, w)$ точно один є допускаючим. Стани $q_1, q_2 \in Q$ називають *еквівалентними*, якщо ці стани не розрізняє жодне слово $w \in T^*$:

$$(q_1 \sim q_2) \Leftrightarrow (\forall\, w \in T^* \colon ((\delta^*(q_1, w) \in F) \leftrightarrow (\delta^*(q_2, w) \in F))). \tag{3.7}$$

Зокрема, якщо $q_1 \in F$, $q_2 \notin F$, то стани q_1 і q_2 розрізняє ε.

Легко перевірити, що визначене співвідношенням (3.7) бінарне відношення «$\sim$» є відношенням еквівалентності на Q.

Приклад 3.30. Розглянемо автомат $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$ із прикладу 3.25, зображений на рис. 3.20. Стани q_0 і q_2 розрізняє, наприклад, слово a, оскільки $\delta^*(q_0, a) = q_1 \in F$, $\delta^*(q_2, a) = q_3 \notin F$. Аналогічно, слово a розрізняє стани q_1 та q_2. Стани q_0 та q_1 не розрізняє жодне слово:

$$\begin{gathered}
\delta^*(q_0, \varepsilon) = q_0 \in F, \quad \delta^*(q_1, \varepsilon) = q_1 \in F,\\
\delta^*(q_0, a^n) = \delta^*(q_1, a^n) = q_1 \in F, \quad n \geqslant 1,\\
\delta^*(q_0, a^n b^m) = \delta^*(q_1, a^n b^m) = q_2 \in F, \quad n \geqslant 0, \quad m \geqslant 1,\\
\delta^*(q_0, w) = \delta^*(q_1, w) = q_3 \notin F, \quad |w|_{ba} \geqslant 1.
\end{gathered}$$

Нарешті, порожнє слово розрізняє стан q_3 від станів q_0, q_1, q_2, оскільки $q_3 \notin F$, $q_0, q_1, q_2 \in F$. Отже, відношення еквівалентності станів «$\sim$» розбиває множину $Q = \{q_0, q_1, q_2, q_3\}$ на такі класи еквівалентності:

$$Q/_{\sim} = \{\{q_0, q_1\}, \{q_2\}, \{q_3\}\}.$$

Приклад 3.31. Відношення еквівалентності станів для скінченного детермінованого автомата $M = \langle Q, \{a, b, c\}, \Delta, \{q_0\}, F\rangle$, зображеного на рис. 3.23, розбиває $Q = \{q_0, q_1, q_2\}$ на два класи:

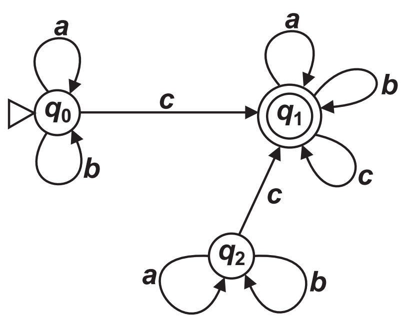

Рис. 3.23

$$\{q_0, q_1, q_2\}/_{\sim} = \{\{q_0, q_2\}, \{q_1\}\}.$$

Зазначимо, що стан q_2 недосяжний з початкового стану q_0, однак це не впливає на коректність визначення відношення «$\sim$».

Лема 3.3. *Нехай $q_1 \sim q_2$ і $u \in T^*$. Тоді $\delta^*(q_1, u) \sim \delta^*(q_2, u)$.*

Вправа 3.15. Довести лему 3.3 самостійно.

Лема 3.4. *Нехай $q_1 = \delta^*(q_0, w_1)$, $q_2 = \delta^*(q_0, w_2)$ $(w_1, w_2 \in T^*)$, $L = L[M]$. Тоді справджується еквівалентність*

$$(q_1 \sim q_2) \Leftrightarrow (w_1 \underset{L}{\sim} w_2).$$

Доведення. Еквівалентність слів $w_1, w_2 \in T^*$ відносно мови L означає правдивість еквіваленції

$$\forall\, u \in T^* \colon (w_1 u \in L) \leftrightarrow (w_2 u \in L).$$

Належність $w_i u \in L$ $(i = 1, 2)$ запишемо через розширену функцію переходів δ^* з урахуванням формули (3.4) та результату вправи 3.1 (п. 3):

$$(w_i u \in L) \Leftrightarrow (\delta^*(q_0, w_i u) \in F) \Leftrightarrow (\delta^*(\delta^*(q_0, w_i), u) \in F) \Leftrightarrow (\delta^*(q_i, u) \in F),$$

де $i = 1, 2$. Звідси випливає співвідношення (3.7), тобто $q_1 \sim q_2$. □

Зауваження 3.11. Якщо всі стани скінченного детермінованого автомата $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ досяжні з q_0, то між фактор-множиною $Q/_{\sim}$ і фактор-множиною $T^*/_{\underset{L}{\sim}}$ існує взаємно однозначна відповідність:

$$[q] = [\delta^*(q_0, w)] \leftrightsquigarrow [w]_L, \tag{3.8}$$

де $w \in T^*$, таке, що $q = \delta^*(q_0, w)$. Якщо $q_1 = \delta^*(q_0, w_1)$, $q_2 = \delta^*(q_0, w_2)$, то згідно з лемою 3.4 $q_1 \sim q_2$ ($[q_1] = [q_2]$) тоді й тільки тоді, коли $w_1 \underset{L}{\sim} w_2$ ($[w_1]_L = [w_2]_L$), тобто відповідність (3.8) визначена коректно.

Так, для формальної мови $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$, яку допускає автомат, зображений на рис. 3.20 (див. приклад 3.25), отримуємо відповідність

$$\begin{aligned}
[q_0] = [q_1] = \{q_0, q_1\} = [\delta^*(q_0, \varepsilon)] &\leftrightsquigarrow [\varepsilon]_L = \{a^n : n \geqslant 0\};\\
[q_2] = \{q_2\} = [\delta^*(q_0, b)] &\leftrightsquigarrow [b]_L = \{a^n b^m : n \geqslant 0, m \geqslant 1\};\\
[q_3] = \{q_3\} = [\delta^*(q_0, ba)] &\leftrightsquigarrow [ba]_L = \{u \in T^* : |u|_{ba} \geqslant 1\}.
\end{aligned}$$

3.6.2. Видалення станів, недосяжних з початкового

Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат з функцією переходів $\delta : (Q \times T) \to Q$. Рекурсією за індексом k введемо множини X_k ($k \geqslant 0$):

$$\begin{gathered}
X_0 = \{q_0\};\\
X_{k+1} = \{\delta(q, a) : q \in X_k, a \in T\},\ k \geqslant 0.
\end{gathered}$$

За побудовою кожна X_k ($k \geqslant 0$) є множиною станів, досяжних з q_0 деяким словом довжиною k. Звідси $\widetilde{Q} = \bigcup\limits_{k=0}^{\infty} X_k$ містить всі стани, досяжні з q_0. Сформулюємо дві прості властивості послідовності X_k ($k \geqslant 0$).

Лема 3.5. *1. Якщо* $X_{k+1} \subset \bigcup\limits_{j=0}^{k} X_j$, *то* $X_m \subset \bigcup\limits_{j=0}^{k} X_j$ *для всіх* $m > k$.

2. Кожний стан, досяжний з q_0*, досягається з* q_0 *деяким словом довжиною не більш ніж* $|Q| - 1$*, тобто* $\widetilde{Q} = \bigcup\limits_{k=0}^{|Q|-1} X_k$.

Вправа 3.16. Довести лему 3.5 самостійно.

Отже, для побудови множини $\widetilde{Q}$ можна послідовно будувати множини $\widetilde{Q}_k = \bigcup\limits_{j=0}^{k} X_j$ $(k \geqslant 0)$, закінчуючи процес на кроці $k = |Q| - 1$ або за умови $\widetilde{Q}_k = \widetilde{Q}_{k-1}$ (на кроці $k \geqslant 1$). Остання побудована множина $\widetilde{Q}_k$, очевидно, збігається з $\widetilde{Q}$. Зазначимо, що множини $\widetilde{Q}_k$ $(k \geqslant 0)$ доцільно будувати за рекурсією

$$\widetilde{Q}_0 = X_0;$$
$$\widetilde{Q}_{k+1} = \widetilde{Q}_k \cup X_{k+1},\ \ k \geqslant 0.$$

Видаляючи стани, недосяжні з початкового, слід також видалити всі такі переходи $(q_1, a, q_2) \in \Delta$, де принаймні один зі станів q_1 або q_2 недосяжний з q_0, тобто перейти до множини переходів

$$\widetilde{\Delta} = \{(q_1, a, q_2) \colon a \in T, q_1 \in \widetilde{Q}, q_2 \in \widetilde{Q}, (q_1, a, q_2) \in \Delta\}.$$

Далі, оскільки будь-який стан досяжний із самого себе порожнім словом, отримуємо, що $q_0 \in \widetilde{Q}$.

Нарешті, деякі зі станів $q \in Q \setminus \widetilde{Q}$ можуть бути в автоматі M допускаючими, тому, видаляючи такі стани, слід перейти до множини допускаючих станів $\widetilde{F} = F \cap \widetilde{Q}$.

Отже, отримано скінченний автомат $\widetilde{M} = \langle \widetilde{Q}, T, \widetilde{\Delta}, \{q_0\}, \widetilde{F} \rangle$, який за побудовою еквівалентний вихідному автомату M та містить лише стани, досяжні з початкового стану q_0.

Теорема 3.7. *Скінченний автомат* $\widetilde{M} = \langle \widetilde{Q}, T, \widetilde{\Delta}, \{q_0\}, \widetilde{F} \rangle$ *є детермінованим.*

Доведення. Нехай $q_1 \in \widetilde{Q}$, $a \in T$. Тоді, оскільки вихідний автомат M детермінований, існує єдиний стан $q_2 \in Q$, такий, що $(q_1, a, q_2) \in \Delta$. Але, оскільки $q_2 = \delta(q_1, a)$ і q_1 досяжний з q_0, стан q_2 також досяжний з q_0, тобто $q_2 \in \widetilde{Q}$ та $(q_1, a, q_2) \in \widetilde{\Delta}$. Оскільки скінченний автомат $\widetilde{M}$ має лише один початковий стан $q_0 \in \widetilde{Q}$, детермінованість автомата $\widetilde{M}$ повністю доведено. □

Приклад 3.32. Розглянемо автомат $M = \langle Q, \{a, b, c\}, \Delta, \{q_0\}, F \rangle$ із прикладу 3.31, зображений на рис. 3.23. Щоб визначити, які стани досяжні з початкового стану q_0, рекурсією за $0 \leqslant k \leqslant |Q| - 1 = 2$ побудуємо множини X_k та $\widetilde{Q}_k$:

$$X_0 = \{q_0\},\ \widetilde{Q}_0 = X_0 = \{q_0\};$$
$$X_1 = \{q_0, q_1\},\ \widetilde{Q}_1 = \widetilde{Q}_0 \cup X_1 = \{q_0, q_1\};$$
$$X_2 = \{q_0, q_1\},\ \widetilde{Q}_2 = \widetilde{Q}_1 \cup X_2 = \{q_0, q_1\}.$$

Зазначимо, що умова $\widetilde{Q}_k = \widetilde{Q}_{k-1}$ була виконана лише на останньому кроці за $k = |Q| - 1 = 2$.

Отже, $\widetilde{Q} = \widetilde{Q}_2 = \{q_0, q_1\}$, тобто недосяжним з q_0 є лише стан q_2. Видаляючи з Δ переходи, що містять q_2, отримуємо множину переходів

$$\widetilde{\Delta} = \{(q_0, a, q_0), (q_0, b, q_0), (q_0, c, q_1), (q_1, a, q_1), (q_1, b, q_1), (q_1, c, q_1)\}.$$

Нарешті, стан q_2 в автоматі M не є допускаючим, тобто $\widetilde{F} = F$. Отже, отримано скінченний автомат $\widetilde{M} = \langle \widetilde{Q}, \{a, b, c\}, \widetilde{\Delta}, \{q_0\}, \widetilde{F} \rangle$ (рис. 3.24), який за побудовою еквівалентний вихідному автомату M та містить лише стани, досяжні з початкового стану q_0.

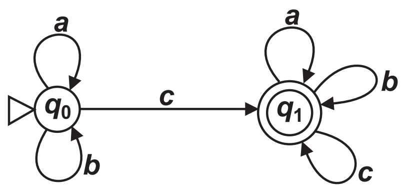

Рис. 3.24

Приклад 3.33. Розглянемо автомат $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F \rangle$, зображений на рис. 3.25. Щоб визначити, які стани досяжні з q_0, побудуємо множини X_k та $\widetilde{Q}_k$, $0 \leqslant k \leqslant |Q| - 1 = 6$:

$$X_0 = \{q_0\},\ \widetilde{Q}_0 = X_0 = \{q_0\};$$
$$X_1 = \{q_1, q_2\},\ \widetilde{Q}_1 = \widetilde{Q}_0 \cup X_1 = \{q_0, q_1, q_2\};$$
$$X_2 = \{q_0, q_3, q_4\},\ \widetilde{Q}_2 = \widetilde{Q}_1 \cup X_2 = \{q_0, q_1, q_2, q_3, q_4\};$$
$$X_3 = \{q_1, q_2\},\ \widetilde{Q}_3 = \widetilde{Q}_1 \cup X_3 = \{q_0, q_1, q_2, q_3, q_4\}.$$

Отже, умова $\widetilde{Q}_k = \widetilde{Q}_{k-1}$ виконується на кроці $k = 3$, тобто немає потреби будувати X_k та $\widetilde{Q}_k$ для $k \geqslant 4$. Таким чином, $\widetilde{Q} = \widetilde{Q}_3$, тобто недосяжними з q_0 є стани q_5 та q_6. Множину переходів $\widetilde{\Delta}$ отримуємо із множини переходів вихідного автомата M, видаляючи чотири переходи, що містять стани q_5 та q_6:

$$\widetilde{\Delta} = \Delta \setminus \{(q_5, a, q_4), (q_5, b, q_3), (q_6, a, q_3), (q_6, b, q_4)\}.$$

Нарешті, множину допускаючих станів $\widetilde{F}$ отримуємо із множини F, видаляючи стани q_5 та q_6, які є допускаючими у вихідному автоматі:

$$\widetilde{F} = F \setminus \{q_5, q_6\} = \{q_3, q_4\}.$$

Таким чином, отримано скінченний автомат $\widetilde{M} = \langle \widetilde{Q}, \{a, b\}, \widetilde{\Delta}, \{q_0\}, \widetilde{F}\rangle$ (рис. 3.26), який за побудовою еквівалентний вихідному автомату M та містить лише стани, досяжні з початкового стану q_0.

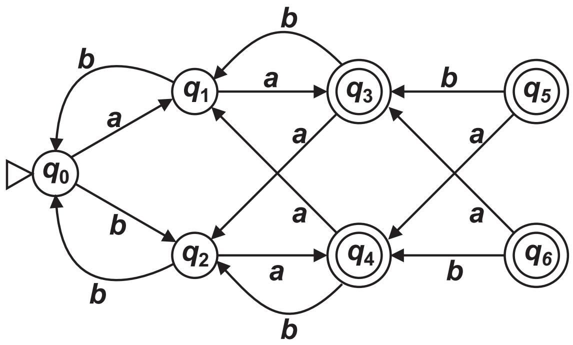

Рис. 3.25

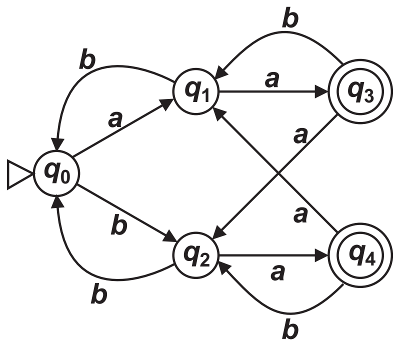

Рис. 3.26

Зауважимо, що процедуру видалення станів, недосяжних з жодного початкового, легко узагальнити на довільні скінченні автомати (див., наприклад, [4]).

3.6.3. Мінімальний детермінований автомат, отриманий злиттям еквівалентних станів

Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат, всі стани якого досяжні з q_0; $L = L[M]$ – мова, яку допускає автомат M; $\delta\colon (Q \times T) \to Q$ – функція переходів автомата M; «$\sim$» – відношення еквівалентності на Q (див. підрозд. 3.6.1). Введемо до розгляду автомат $M_{\text{merge}} = \langle Q_{\text{merge}}, T, \Delta_{\text{merge}}, I_{\text{merge}}, F_{\text{merge}}\rangle$, де

$$\begin{aligned}
&Q_{\text{merge}} = Q/_{\sim};\\
&\Delta_{\text{merge}} = \{([q_1], a, [q_2])\colon (q_1, a, q_2) \in \Delta\};\\
&I_{\text{merge}} = \{[q_0]\};\\
&F_{\text{merge}} = \{[q]\colon q \in F\}.
\end{aligned} \tag{3.9}$$

Теорема 3.8. *Автомат M_{merge} – мінімальний детермінований автомат, що допускає мову L.*

Доведення. **Детермінованість.** Автомат M_{merge} має один початковий стан $[q_0]$. Отже, для доведення детермінованості автомата M_{merge}

необхідно показати, що для будь-якого стану $[q_1] \in Q_{\text{merge}}$ ($q_1 \in Q$) і символа $a \in T$ існує єдиний стан $\alpha \in Q_{\text{merge}}$, такий, що $([q_1], a, \alpha) \in \Delta_{\text{merge}}$. Враховуючи детермінованість автомата M, для фіксованої пари $q_1 \in Q$, $a \in T$ існує єдиний стан $q_2 \in Q$, такий, що $(q_1, a, q_2) \in \Delta$, тобто $([q_1], a, [q_2]) \in \Delta_{\text{merge}}$, і можемо покласти $\alpha = [q_2]$. Припустимо, що існує інший стан $\beta \in Q_{\text{merge}}$, такий, що $([q_1], a, \beta) \in \Delta_{\text{merge}}$. Тоді, за визначенням відношення Δ_{merge}, існують $p_1, p_2 \in Q$, такі, що

$$[p_1] = [q_1],\ [p_2] = \beta,\ (p_1, a, p_2) \in \Delta,$$

тобто $p_1 \sim q_1$, $p_2 = \delta(p_1, a)$. Враховуючи, що $q_2 = \delta(q_1, a)$, за лемою 3.3 отримуємо еквівалентність $p_2 \sim q_2$, тобто $\beta = [p_2] = [q_2] = \alpha$. Отже, автомат M_{merge} детермінований з функцією переходів $\delta_{\text{merge}}\colon (Q_{\text{merge}} \times T) \to Q_{\text{merge}}$, яка визначається рівністю

$$\delta_{\text{merge}}([q_1], a) = [q_2] = [\delta(q_1, a)].$$

Підкреслимо, що значення виразу $\delta_{\text{merge}}([q_1], a)$ не залежить від вибору представника $p_1 \in [q_1]$ ($p_1 \sim q_1$), оскільки за лемою 3.3 маємо еквівалентність $\delta(q_1, a) \sim \delta(p_1, a)$, тобто $[\delta(q_1, a)] = [\delta(p_1, a)]$. Із рекурентних співвідношень (3.3) індукцією за довжиною слова $w \in T^*$ легко встановити вигляд розширеної функції переходів $\delta^*_{\text{merge}}\colon (Q_{\text{merge}} \times T^*) \to Q_{\text{merge}}$:

$$\delta^*_{\text{merge}}([q_1], w) = [q_2] = [\delta^*(q_1, w)], \quad q_2 = \delta^*(q_1, w),$$

де $\delta^*\colon (Q \times T^*) \to Q$ – розширена функція переходів автомата M.

Еквівалентність вихідному автомату. Спочатку зазначимо, що належність $\alpha = [q]$ до F_{merge} не залежить від вибору представника $q \in \alpha$, тобто, якщо $q_1 \sim q_2$ ($q_1, q_2 \in Q$), то (див. формулу (3.7))

$$([q_1] \in F_{\text{merge}}) \Leftrightarrow ([q_2] \in F_{\text{merge}}).$$

Таким чином, з урахуванням визначення множини F_{merge} отримуємо:

$$([q] \in F_{\text{merge}}) \Leftrightarrow (q \in F).$$

Тепер доведемо, що слово $w \in T^*$ допускається автоматом M_{merge} тоді й тільки тоді, коли воно допускається автоматом M (див. (3.4)):

$$(w \in L[M_{\text{merge}}]) \Leftrightarrow (\delta^*_{\text{merge}}([q_0], w) \in F_{\text{merge}}) \Leftrightarrow ([\delta^*(q_0, w)] \in F_{\text{merge}}) \Leftrightarrow$$
$$\Leftrightarrow (\delta^*(q_0, w) \in F) \Leftrightarrow (w \in L[M]).$$

Мінімальність. Оскільки між множинами $Q_{\text{merge}} = Q^*/_{\sim}$ та $T^*/_{\underset{L}{\sim}}$ існує взаємно однозначна відповідність (див. зауваження 3.11), отримуємо рівність $|Q_{\text{merge}}| = \left|T^*/_{\underset{L}{\sim}}\right| = i_L$. Таким чином, автомат M_{merge} має мінімально можливу кількість станів серед детермінованих автоматів, які допускають мову L. □

Будемо казати, що автомат M_{merge} отримано з автомата M *об'єднанням еквівалентних станів.*

Приклад 3.34. Об'єднуючи еквівалентні стани скінченного автомата $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$ із прикладів 3.25, 3.30 (див. рис. 3.20), отримуємо автомат, зображений на рис. 3.27. Класи еквівалентності за відношенням «$\sim$», як було показано у прикладі 3.30, мають вигляд

$$[q_0] = [q_1] = \{q_0, q_1\},\ [q_2] = \{q_2\},\ [q_3] = \{q_3\}.$$

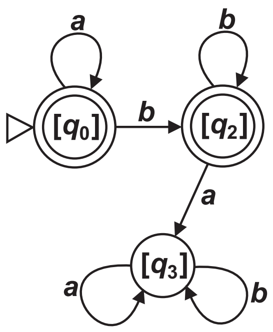

Рис. 3.27

Отриманий автомат справді еквівалентний вихідному автомату M, оскільки допускає ту саму мову $\{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$.

Доведемо, що автомат M_{merge} збігається з автоматом M_L (див. процедуру (3.6)) з точністю до перейменування станів. Для цього введемо поняття ізоморфних автоматів.

Означення 3.8. Скінченні автомати $M_1 = \langle Q_1, T, \Delta_1, I_1, F_1\rangle$ та $M_2 = \langle Q_2, T, \Delta_2, I_2, F_2\rangle$ називають ізоморфними, якщо існує бієктивне відображення (ізоморфізм) $f : Q_1 \to Q_2$, таке, що для будь-яких $q \in Q_1$, $p \in Q_1$, $a \in T$

$$\begin{gathered}((q, a, p) \in \Delta_1) \Leftrightarrow ((f(q), a, f(p)) \in \Delta_2);\\ (q \in I_1) \Leftrightarrow (f(q) \in I_2);\\ (q \in F_1) \Leftrightarrow (f(q) \in F_2).\end{gathered} \tag{3.10}$$

Ізоморфність скінченних автоматів фактично означає, що ці автомати збігаються з точністю до перейменування станів.

Наведемо декілька властивостей ізоморфності скінченних автоматів, які є очевидними або легко перевіряються.

1. Будь-який автомат ізоморфний самому собі.

2. Якщо автомат M_1 ізоморфний автомату M_2, то автомат M_2 ізоморфний автомату M_1.

3. Якщо автомат M_1 ізоморфний автомату M_2 та автомат M_2 ізоморфний автомату M_3, то автомат M_1 ізоморфний автомату M_3.

4. Ізоморфні автомати еквівалентні.

5. Якщо автомат M_1 ізоморфний автомату M_2 і автомат M_1 детермінований, то автомат M_2 також детермінований.

Зауваження 3.12. Для детермінованих автоматів $\langle Q_1, T, \Delta_1, I_1, F_1\rangle$ та $\langle Q_2, T, \Delta_2, I_2, F_2\rangle$ еквівалентність (3.10) у визначенні ізоморфізму можна замінити співвідношенням для функцій переходів $\delta_1 \colon (Q_1 \times T) \to Q_1$ та $\delta_2 \colon (Q_2 \times T) \to Q_2$:

$$\delta_2(f(q), a) = f(\delta_1(q, a)), \quad q \in Q_1,\ a \in T. \tag{3.11}$$

Приклад 3.35. Скінченний автомат, зображений на рис. 3.22, ізоморфний автоматам, зображеним на рис. 3.21 та 3.27, однак жоден із цих автоматів не ізоморфний автомату на рис. 3.20. Зазначимо, що всі чотири автомати еквівалентні, оскільки допускають ту саму формальну мову $\{a^n b^m \colon n \geqslant 0,\ m \geqslant 0\}$.

Нехай $M_L = \langle Q_L, T, \Delta_L, \{q_{0,L}\}, F_L\rangle$, мінімальний детермінований автомат, який допускає мову $L = L[M]$, отриманий за процедурою (3.6):

$$\begin{aligned}
&Q_L = T^*/_{\underset{L}{\sim}}\,;\\
&\delta_L([w]_L, a) = [wa]_L, \quad w \in T^*,\ a \in T;\\
&q_{0,L} = [\varepsilon]_L;\\
&F_L = \{[w]_L \colon w \in L\}.
\end{aligned}$$

Оскільки всі стани вихідного автомата M досяжні з початкового стану q_0, існує бієкція $f : Q/_{\sim} \to T^*/_{\underset{L}{\sim}}$, яка визначається рівністю $f([q]) = [w]_L$ (див. зауваження 3.11), де $w \in T^*$, $q = \delta^*(q_0, w)$, $\delta^* \colon (Q \times T^*) \to Q$ – розширена функція переходів автомата M.

Лема 3.6. *Відображення f є ізоморфізмом автомата M_{merge} в автомат M_L.*

Доведення. Перевіримо умову (3.11). Зафіксуємо $q \in Q$, $w \in T^*$, $q = \delta^*(q_0, w)$, $a \in T$. Тоді $\delta^*(q_0, wa) = \delta(\delta^*(q_0, w), a) = \delta(q, a)$, і співвідношення (3.11) справджується:

$$\delta_L(f([q]), a) = \delta_L([w]_L, a) = [wa]_L = f([\delta(q, a)]) = f(\delta_{\text{merge}}([q], a)).$$

Далі для початкових станів $[q_0] = [\delta^*(q_0, \varepsilon)] \in Q_{\text{merge}}$ та $[\varepsilon]_L \in Q_L$ отримуємо:

$$f([q_0]) = [\varepsilon]_L = q_{0,L}.$$

Нарешті, множини допускаючих станів F_{merge} та F_L також пов'язані відображенням f (див. формулу (3.4)):

$$([q] \in F_{\text{merge}}) \Leftrightarrow (q \in F) \Leftrightarrow (w \in L) \Leftrightarrow ([w]_L \in F_L)$$

для довільного $q \in Q$ із фіксованим $w \in T^*$, таким, що $q = \delta^*(q_0, w)$.

Отже, визначена бієкція $f : Q/_{\sim} \to T^*/_{\underset{L}{\sim}}$ справді є ізоморфізмом M_{merge} в автомат M_L. □

Приклад 3.36. Автомат M_{merge}, зображений на рис. 3.27, отримано об'єднанням еквівалентних станів з автомата M, зображеного на рис. 3.20 (див. приклад 3.34). Автомат M_L, зображений на рис. 3.22 (див. приклад 3.29) – мінімальний автомат, який допускає формальну мову $L = L[M] = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$. Згідно з лемою 3.6 автомати M_{merge} та M_L ізоморфні, і за ізоморфізм M_{merge} в M_L можна вибрати бієкцію $f : Q/_{\sim} \to T^*/_{\underset{L}{\sim}}$, яка визначається рівністю $f([q]) = [w]_L$, де $q = \delta^*(q_0, w)$. Обчислимо значення $f(\alpha)$ для кожного класу еквівалентності $\alpha \in Q/_{\sim}$:

$$f([q_0]) = f([q_1]) = [\varepsilon]_L,\ \text{оскільки}\ \delta^*(q_0, \varepsilon) = q_0;$$
$$f([q_2]) = [b]_L,\ \text{оскільки}\ \delta^*(q_0, b) = q_2;$$
$$f([q_3]) = [ba]_L,\ \text{оскільки}\ \delta^*(q_0, ba) = q_3.$$

Підкреслимо, що значення $f(\alpha)$ не залежить ані від вибору представника $q \in \alpha$, ані від вибору $w \in T^*$, такого, що $q = \delta^*(q_0, w)$. Так, $[q_0] = [q_1]$, стан q_0 досягається з q_0 лише словом ε, стан q_1 досягається з q_0 будь-яким словом a^{n+1} $(n \geqslant 0)$, але всі ці слова (тобто слова a^n, $n \geqslant 0$) породжують той самий клас еквівалентності, який збігається з класом $[\varepsilon]_L$.

Теорема 3.9. *Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат, який допускає мову $L = L[M]$ і містить i_L станів. Тоді автомат M ізоморфний автомату M_L, отриманому із мови L за процедурою (3.6).*

Доведення. Зазначимо, що детермінований автомат M має найменшу (а саме i_L) кількість станів серед усіх детермінованих автоматів, які допускають мову L. Із мінімальності автомата M випливає:

1) кожний стан $q \in Q$ автомата M досяжний із початкового стану q_0 (стан, який недосяжний з q_0, можна видалити, отримавши еквівалентний автомат з меншою кількістю станів);

2) кожний стан $q \in Q$ автомата M еквівалентний лише собі, тобто $[q] = \{q\}$, і $Q/_{\sim} = \{\{q\} : q \in Q\}$ (різні еквівалентні стани можна об'єднати, отримавши еквівалентний автомат з меншою кількістю станів).

Таким чином, можна встановити очевидну бієкцію $\rho : Q \to Q/_{\sim}$, $\rho(q) = \{q\}$ для кожного $q \in Q$. Легко перевірити, що ρ є ізоморфізмом M в автомат M_{merge}, отриманий з M злиттям еквівалентних станів (див. (3.9)). Далі, оскільки кожний стан $q \in Q$ досяжний з q_0, за лемою 3.6 отримуємо, що автомати M_{merge} та M_L ізоморфні. Отже, M ізоморфний M_{merge}, M_{merge} ізоморфний M_L, звідки випливає ізоморфність автоматів M та M_L. □

Зауваження 3.13. Теорема 3.9 фактично встановлює єдиність (з точністю до ізоморфізму) мінімального детермінованого скінченного автомата, який допускає формальну мову L.

Приклад 3.37. На рис. 3.21, 3.22 та 3.27 зображено автомати, які допускають мову $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$ та мають $i_L = 3$ стани (див., зокрема, приклад 3.27), тобто є мінімальними детермінованими автоматами, які допускають мову L. Очевидно, що всі три автомати справді між собою ізоморфні.

3.7. Алгоритми мінімізації детермінованих скінченних автоматів

У цьому підрозділі розглянемо конкретні алгоритми побудови мінімального за кількістю станів детермінованого скінченного автомата, який еквівалентний заданому скінченному автомату.

3.7.1. Відношення k-еквівалентності станів

Введемо до розгляду узагальнення еквівалентності «$\sim$» (див. підрозд. 3.6.1) для станів детермінованого автомата $M = \langle Q, T, \Delta, \{q_0\}, F \rangle$.

Стани $q_1, q_2 \in Q$ називають k-еквівалентними ($k \geqslant 0$), якщо q_1 і q_2 не розрізняє жодне слово $w \in T^*$ довжиною $|w| \leqslant k$:

$$(q_1 \overset{k}{\sim} q_2) \Leftrightarrow (\forall\, w \in T^* \, (|w| \leqslant k) \colon ((\delta^*(q_1, w) \in F) \leftrightarrow (\delta^*(q_2, w) \in F))) \,.$$

Зокрема, для $k = 0$ отримуємо не більше двох класів:

$$Q/_{\overset{0}{\sim}} = \{F, Q \setminus F\}, \text{ якщо } F \neq \varnothing \text{ та } F \neq Q;$$

у тривіальних випадках $F = \varnothing$ та $F = Q$ отримуємо один клас еквівалентності ($Q \setminus F$ або F відповідно).

Із визначення відношень «$\overset{k}{\sim}$» ($k \geqslant 0$) негайно отримуємо ланцюжок вкладень

$$«\overset{0}{\sim}» \supset «\overset{1}{\sim}» \supset \cdots \supset «\overset{k}{\sim}» \supset \cdots \supset «\sim». \tag{3.12}$$

Лема 3.7. *Існує таке $n \geqslant 0$, що «$\overset{n}{\sim}$» = «$\sim$».*

Доведення. Позначимо через X множину пар нееквівалентних станів:

$$X = \{(p, q) \in Q \times Q \colon p \not\sim q\}.$$

Для кожної пари $(p, q) \in X$ зафіксуємо слово $w_{p,q}$, яке розрізняє стани p та q. Тепер можемо вибрати шукане число n:

$$n = \max\{|w_{p,q}| \colon (p, q) \in X\}. \qquad \square$$

Таким чином, ланцюжок вкладень (3.12) насправді є скінченним:

$$«\overset{0}{\sim}» \supset «\overset{1}{\sim}» \supset \cdots \supset «\overset{n}{\sim}» = «\sim» \tag{3.13}$$

для деякого $n \geqslant 0$.

Так, для формальної мови $L = \{a^n b^m \colon n \geqslant 0,\ m \geqslant 0\}$, яку допускає автомат, зображений на рис. 3.20 (див. приклади 3.25 та 3.30), отримуємо ланцюжок «$\overset{0}{\sim}$» $\supsetneq$ «$\overset{1}{\sim}$» = «$\sim$», де

$$Q/_{\overset{0}{\sim}} = \{F, Q \setminus F\} = \{\{q_0, q_1, q_2\}, \{q_3\}\},\ Q/_{\overset{1}{\sim}} = Q/_{\sim} = \{\{q_0, q_1\}, \{q_2\}, \{q_3\}\}.$$

Лема 3.8. *Нехай $q_1 \overset{k+1}{\sim} q_2$ для деякого $k \geqslant 0$, $a \in T$, $p_1 = \delta(q_1, a)$, $p_2 = \delta(q_2, a)$. Тоді $p_1 \overset{k}{\sim} p_2$.*

Доведення. Нехай $k \geqslant 0$ та $q_1 \overset{k+1}{\sim} q_2$. Припустимо, що стани p_1 і p_2 розрізняє слово $w \in T^*$, $|w| \leqslant k$, тобто із двох станів $\delta^*(p_1, w)$ і $\delta^*(p_2, w)$ в точності один допускаючий. Тоді стани q_1 і q_2 розрізняє слово $u = aw$, оскільки (див. п. 3 вправи 3.1)

$$\delta^*(q_1, u) = \delta^*(q_1, aw) = \delta^*(\delta(q_1, a), w) = \delta^*(p_1, w),$$
$$\delta^*(q_2, u) = \delta^*(q_2, aw) = \delta^*(\delta(q_2, a), w) = \delta^*(p_2, w),$$

і з двох станів $\delta^*(q_1, u) = \delta^*(p_1, w)$ та $\delta^*(q_2, u) = \delta^*(p_2, w)$ в точності один допускаючий. Але $|u| = |aw| \leqslant 1 + k$, тобто $q_1 \overset{k+1}{\nsim} q_2$, що суперечить умові леми. Таким чином, жодне слово $w \in T^*$ довжиною $|w| \leqslant k$ не розрізняє стани p_1 і p_2, тобто $p_1 \overset{k}{\sim} p_2$. □

Зауваження 3.14. Лема 3.3 є наслідком щойно доведеної леми.

Лема 3.9. *Нехай* «$\overset{k}{\sim}$»=«$\overset{k+1}{\sim}$» *для деякого* $k \geqslant 0$. *Тоді* «$\overset{k+1}{\sim}$»=«$\overset{k+2}{\sim}$».

Доведення. Нехай $k \geqslant 0$ та «$\overset{k}{\sim}$» = «$\overset{k+1}{\sim}$». Необхідно довести рівність «$\overset{k+1}{\sim}$» = «$\overset{k+2}{\sim}$», а з урахуванням ланцюжка (3.13) – лише вкладення «$\overset{k+1}{\sim}$» $\subset$ «$\overset{k+2}{\sim}$». Припустимо, що існують такі стани q_1 і q_2, що $q_1 \overset{k+1}{\sim} q_2$, але $q_1 \overset{k+2}{\nsim} q_2$. Тоді стани q_1 і q_2 має розрізняти деяке слово $w \in T^*$, $|w| \leqslant k+2$ (оскільки $q_1 \overset{k+2}{\nsim} q_2$); між тим $|w| > k+1$ (оскільки $q_1 \overset{k+1}{\sim} q_2$). Таким чином, стани q_1 і q_2 має розрізняти слово $w \in T^*$ довжиною $|w| = k+2$.

Оскільки слово $w \in T^*$ розрізняє стани q_1 і q_2, із двох станів $\delta^*(q_1, w)$ і $\delta^*(q_2, w)$ в точності один є допускаючим.

Зобразимо слово w у вигляді $w = au$, $a \in T$, $u \in T^*$, $|u| = k+1$ і розглянемо стани $p_1 = \delta(q_1, a)$, $p_2 = \delta(q_2, a)$. Зазначимо, що слово u розрізняє стани p_1 і p_2:

$$\delta^*(p_1, u) = \delta^*(\delta(q_1, a), u) = \delta^*(q_1, au) = \delta^*(q_1, w),$$
$$\delta^*(p_2, u) = \delta^*(\delta(q_2, a), u) = \delta^*(q_2, au) = \delta^*(q_2, w),$$

тобто з двох станів $\delta^*(p_1, u) = \delta^*(q_1, w)$ та $\delta^*(p_2, u) = \delta^*(q_2, w)$ в точності один є допускаючим. Таким чином, $p_1 \overset{k+1}{\nsim} p_2$. Враховуючи рівність «$\overset{k}{\sim}$» = «$\overset{k+1}{\sim}$», отримуємо, що $p_1 \overset{k}{\nsim} p_2$. Але, за припущенням, $q_1 \overset{k+1}{\sim} q_2$, звідки, за лемою 3.8, $p_1 \overset{k}{\sim} p_2$. Отримана суперечність доводить лему. □

З урахуванням щойно доведеної леми всі вкладення в ланцюжку (3.13) насправді строгі:

$$«\overset{0}{\sim}» \supsetneq «\overset{1}{\sim}» \supsetneq \cdots \supsetneq «\overset{n}{\sim}» = «\sim» \tag{3.14}$$

для деякого $n \geqslant 0$.

Нехай i_k $(k \geqslant 0)$ – індекс відношення «$\overset{k}{\sim}$», i_Q – індекс відношення еквівалентності станів «$\sim$», тобто $i_k = \left|Q/_{\overset{k}{\sim}}\right|$, $i_Q = \left|Q/_{\sim}\right|$.

Вправа 3.17. Довести, що $i_k \leqslant i_{k+1}$ для будь-якого $k \geqslant 0$. Якщо «$\overset{k}{\sim}$» $\supsetneq$ «$\overset{k+1}{\sim}$», то $i_k < i_{k+1}$.

Вказівка. Скористатися вкладенням «$\overset{k}{\sim}$» $\supset$ «$\overset{k+1}{\sim}$» та технікою доведення леми 3.2.

Отже, з урахуванням (3.14) отримуємо ланцюжок нерівностей для індексів i_k $(k \geqslant 0)$:

$$i_0 < i_1 < \cdots < \cdots < i_n = i_Q \tag{3.15}$$

для деякого $n \geqslant 0$.

Приклад 3.38. Розглянемо скінченний детермінований автомат $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$, зображений на рис. 3.28. Стани q_0 та q_1, так само, як і q_0 та q_2, розрізняє слово b (довжиною 1); можна довести, що стани q_1 та q_2, так само, як і стани q_3 та q_4, не розрізняє жодне слово. Отже, ланцюжок вкладень (3.14) набуває вигляду «$\overset{0}{\sim}$» $\supsetneq$ «$\overset{1}{\sim}$» $=$ «$\sim$», де

$$Q/_{\overset{0}{\sim}} = \{\{q_0, q_1, q_2\}, \{q_3, q_4\}\},$$
$$Q/_{\overset{1}{\sim}} = Q/_{\sim} = \{\{q_0\}, \{q_1, q_2\}, \{q_3, q_4\}\}.$$

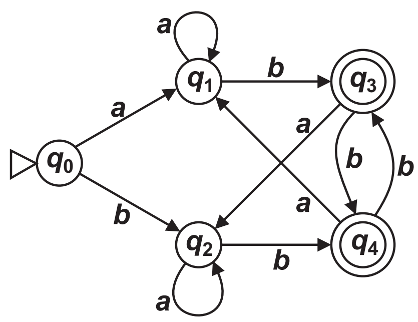

Рис. 3.28

Ланцюжок нерівностей (3.15), очевидно, має вигляд

$$i_0 = 2 < i_1 = i_Q = 3.$$

Вправа 3.18. Перевірити, що стани q_1 та q_2 справді не розрізняє жодне слово; що стани q_3 та q_4 справді не розрізняє жодне слово.

Приклад 3.39. Розглянемо скінченний детермінований автомат $M = \langle Q, \{a\}, \Delta, \{q_0\}, \{q_0\}\rangle$, де

$$Q = \{q_0, q_1, \ldots, q_{m-1}\},$$
$$\Delta = \{(q_k, a, q_{k+1})\colon 0 \leqslant k \leqslant m-2\} \cup \{(q_{m-1}, a, q_0)\}.$$

Запишемо розширену функцію переходів цього автомата:

$$\delta^*(q_k, a^j) = q_{(k+j) \bmod m} \quad 0 \leqslant k \leqslant m-1, j \geqslant 0.$$

Очевидно, що стан q_0 і будь-який стан q_k $(1 \leqslant k \leqslant m-1)$ розрізняє слово ε (а також будь-яке слово a^{mj}, $j \geqslant 0$). Далі, найкоротшим словом, яке розрізняє стан q_{m-1} і будь-який стан q_k $(0 \leqslant k \leqslant m-2)$, є слово a. Взагалі, найкоротшим словом, яке розрізняє стан q_{m-j} $(1 \leqslant j \leqslant m-1)$ і будь-який стан q_k $(0 \leqslant k \leqslant m-j-1)$, є слово a^j. Отже, отримуємо таку послідовність відношень «$\overset{k}{\sim}$» $(k \geqslant 0)$:

$$
\begin{gathered}
«\overset{0}{\sim}» = \{\{q_0\}, \{q_1, q_2, \ldots, q_{m-1}\}\}, \\
«\overset{1}{\sim}» = \{\{q_0\}, \{q_{m-1}\}, \{q_1, q_2, \ldots, q_{m-2}\}\}, \\
\vdots \\
«\overset{m-2}{\sim}» = \{\{q_0\}, \{q_{m-1}\}, \{q_{m-2}\}, \ldots, \{q_1\}\}.
\end{gathered}
$$

Отже, ланцюжок (3.15) набуває вигляду

$$
i_0 = 2 < i_1 = 3 < i_2 = 4 < \cdots < i_{m-3} = m-1 < i_{m-2} = i_Q = m.
$$

Лема 3.10. *Для детермінованого автомата $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$, який містить принаймні два стани, існує не більше ніж $|Q|-1$ різних відношень «$\overset{k}{\sim}$» $(k \geqslant 0)$.*

Доведення. Лема стверджує, що довжина ланцюжків (3.15) та (3.14) (кількість різних відношень «$\overset{k}{\sim}$», $k \geqslant 0$) не перевищує $|Q|-1$. Одразу зазначимо, що $i_Q = \left|Q/_{\sim}\right| \leqslant |Q|$; до того ж, $i_0 \geqslant 2$ у нетривіальних випадках $(\varnothing \subsetneq F \subsetneq Q)$ та $i_0 = 1$ у тривіальних випадках (якщо $F = \varnothing$ або $F = Q$). Розглянемо тривіальні ($F = \varnothing$ або $F = Q$) та нетривіальні ($F \neq \varnothing$ та $F \neq Q$) випадки окремо.

Нехай $\varnothing \subsetneq F \subsetneq Q$. Тоді з урахуванням (3.15) маємо ланцюжок нерівностей

$$
2 \leqslant i_0 < i_1 < i_2 < \cdots < i_n = i_Q \leqslant |Q|,
$$

де n – деяке ціле невід'ємне число. Оскільки на відрізку $[2; |Q|]$ міститься лише $|Q|-1$ ціле число, твердження леми у цьому випадку справджується.

Нехай $F = \varnothing$. Тоді автомат не допускає жодного слова, всі стани є попарно еквівалентними, і ланцюжок нерівностей (3.15) набуває тривіального вигляду

$$
1 = i_0 = i_Q,
$$

тобто ланцюжок вкладень (3.14) містить лише одне відношення «$\overset{0}{\sim}$». Оскільки, за умовою, $|Q| \geqslant 2$, твердження леми справджується.

Нехай $F = Q$. Тоді автомат допускає всі слова, всі стани є попарно еквівалентними, і, аналогічно випадку $F = \varnothing$, ланцюжок вкладень (3.14) містить лише одне відношення «$\overset{0}{\sim}$». Оскільки, за умовою, $|Q| \geqslant 2$, твердження леми справджується. □

Наслідок. *Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – скінченний детермінований автомат, $q_1, q_2 \in Q$ і $q_1 \not\sim q_2$. Тоді q_1 і q_2 розрізняються деяким словом довжиною не більше $|Q| - 2$.*

Доведення. Нехай q_1 і q_2 розрізняються деяким словом довжиною $m \geqslant |Q| - 1$, але не розрізняються жодним словом довжиною не більше $|Q| - 2$. Тоді $q_1 \overset{|Q|-2}{\sim} q_2$, але $q_1 \overset{m}{\not\sim} q_2$. Це означає, що «$\overset{|Q|-2}{\sim}$» $\supsetneq$ «$\overset{m}{\sim}$» для деякого $m \geqslant |Q| - 1$ і, за лемою 3.9, відношення «$\overset{k}{\sim}$» $(0 \leqslant k \leqslant |Q| - 1)$ є попарно різними. Таким чином, ланцюжок (3.14) містить не менш ніж $|Q|$ різних відношень, що суперечить твердженню леми 3.10. □

Приклад 3.40. Розглянемо детермінований скінченний автомат $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$, зображений на рис. 3.29.

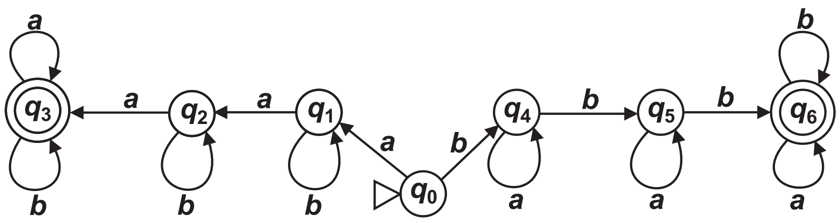

Рис. 3.29

Неважко зрозуміти, що ланцюжок вкладень (3.14) набуває вигляду «$\overset{0}{\sim}$» $\supsetneq$ «$\overset{1}{\sim}$» $\supsetneq$ «$\overset{2}{\sim}$» = «$\sim$», де

$$Q/_{\overset{0}{\sim}} = \{\{q_0, q_1, q_2, q_4, q_5\}, \{q_3, q_6\}\},$$
$$Q/_{\overset{1}{\sim}} = \{\{q_0, q_1, q_4\}, \{q_2\}, \{q_5\}, \{q_3, q_6\}\},$$
$$Q/_{\overset{2}{\sim}} = Q/_{\sim} = \{\{q_0\}, \{q_1\}, \{q_4\}, \{q_2\}, \{q_5\}, \{q_3, q_6\}\}.$$

Ланцюжок нерівностей (3.15), очевидно, має вигляд

$$i_0 = 2 < i_1 = 4 < i_2 = i_Q = 6,$$

тобто містить не більше (а насправді менше) ніж $|Q|-1=7-1=6$ відношень, що узгоджується з твердженням леми 3.10. Згідно з наслідком із леми 3.10 будь-яку пару нееквівалентних станів цього автомата можна розрізнити деяким словом довжиною не більш ніж $|Q|-2=7-2=5$. Насправді у цьому випадку для будь-якої пари нееквівалентних станів можна вибрати слово довжиною 2 або менше (так, q_1 і q_4 розрізняє слово aa), оскільки ланцюжок (3.14) містить лише 3 різних відношення.

Приклад 3.41. Для автомата $M=\langle Q,\{a\},\Delta,\{q_0\},\{q_0\}\rangle$ із прикладу 3.39 ланцюжок (3.14) містить в точності $|Q|-1=m-1$ відношення, всі стани попарно нееквівалентні, і кожна пара станів розрізняється деяким словом довжиною не більш ніж $|Q|-2=m-2$. Зазначимо, що найкоротшим словом, яке розрізняє стани q_1 і q_2, є слово a^{m-2} довжиною в точності $|Q|-2=m-2$.

Наведений приклад показує, що оцінку леми 3.10 на кількість відношень в ланцюжку (3.14) в загальному випадку покращити не можна.

3.7.2. Алгоритми мінімізації

Нехай $M=\langle Q,T,\Delta,\{q_0\},F\rangle$ – детермінований скінченний автомат із функцією переходів $\delta\colon (Q\times T)\to Q$. Розглянемо два алгоритми побудови мінімального за кількістю станів детермінованого автомата, еквівалентного автомату M.

Послідовна побудова відношень k-еквівалентності. Алгоритм ґрунтується на простому факті, який сформулюємо у вигляді леми.

Лема 3.11. *Нехай $q_1,q_2\in Q$, $k\geqslant 0$. Тоді справджується еквівалентність*

$$(q_1 \overset{k+1}{\sim} q_2) \Leftrightarrow \begin{cases} q_1 \overset{0}{\sim} q_2, \\ \forall\, a\in T\colon \delta(q_1,a) \overset{k}{\sim} \delta(q_2,a). \end{cases}$$

Наслідок. *Нехай $q_1,q_2\in Q$, $k\geqslant 0$. Тоді справджується еквівалентність*

$$(q_1 \overset{k+1}{\sim} q_2) \Leftrightarrow \begin{cases} q_1 \overset{k}{\sim} q_2, \\ \forall\, a\in T\colon \delta(q_1,a) \overset{k}{\sim} \delta(q_2,a). \end{cases} \tag{3.16}$$

Вправа 3.19. Самостійно довести лему 3.11 з наслідком.

На основі еквівалентності (3.16) запишемо рекурсивний алгоритм побудови відношень «$\overset{k}{\sim}$» ($k\geqslant 0$). Нехай $Q=\{q_0,\ldots,q_{n-1}\}$ і $|Q|\geqslant 2$ (якщо автомат містить лише один стан, мінімізація, очевидно, неможлива). Також вважатимемо, що всі стани автомата M досяжні з початкового – інакше застосовуємо описану у підрозд. 3.6.2 процедуру видалення станів, які недосяжні з початкового, отримуючи детермінований скінченний автомат, еквівалентний заданому.

1. Для кожної пари станів $p, q \in Q$ покласти

$$(p \stackrel{0}{\sim} q) \Leftrightarrow ((p \in F) \leftrightarrow (q \in F)).$$

2. Покласти $k = 0$.
3. Якщо $k = |Q| - 2$, закінчити роботу.
4. Для кожної пари станів $q_i, q_j \in Q$ $(i < j)$ покласти

$$(q_i \stackrel{k+1}{\sim} q_j) \Leftrightarrow \begin{cases} q_i \stackrel{k}{\sim} q_j, \\ \forall\, a \in T \colon \delta(q_i, a) \stackrel{k}{\sim} \delta(q_j, a). \end{cases} \tag{3.17}$$

5. Якщо «$\stackrel{k+1}{\sim}$» = «$\stackrel{k}{\sim}$», закінчити роботу.
6. Збільшити k на 1 і перейти до п. 3.

Зауваження 3.15. Зазначимо, що алгоритм визначає факт $q_i \stackrel{k}{\sim} q_j$ чи $q_i \stackrel{k}{\nsim} q_j$ $(k \geqslant 0)$ лише для таких пар станів $q_i, q_j \in Q$, коли $i < j$. З урахуванням симетричності відношень еквівалентності це визначає $q_j \stackrel{k}{\sim} q_i$ чи $q_j \stackrel{k}{\nsim} q_i$ $(k \geqslant 0,\ i < j)$. Нарешті, за рефлексивністю відношень еквівалентності, $q \stackrel{k}{\sim} q$ для всіх $q \in Q$, $k \geqslant 0$.

Зауваження 3.16. Згідно з лемами 3.9 та 3.10 останнє обчислене наведеним алгоритмом відношення «$\stackrel{k+1}{\sim}$» збігається з відношенням еквівалентності станів «$\sim$» автомата M. Відношення «$\sim$» визначає за співвідношеннями (3.9) автомат M_{merge} із множиною станів $Q/_{\sim}$, який, за теоремою 3.8, є мінімальним за кількістю станів детермінованим автоматом, еквівалентним вихідному автомату M.

Зауваження 3.17. Нехай у п. 4 стани q_i та q_j належать одному класу еквівалентності A за відношенням «$\stackrel{k}{\sim}$». Якщо існує $a \in T$, такий, що $\delta(q_i, a) \stackrel{k+1}{\nsim} \delta(q_j, a)$, клас еквівалентності A розділяється за символом a. Якщо $\delta(q_i, a) \stackrel{k+1}{\sim} \delta(q_j, a)$ для всіх $a \in T$, клас еквівалентності A не розділяється за символом a.

Приклад 3.42. Розглянемо детермінований скінченний автомат $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$, зображений на рис. 3.30. Для мінімізації автомата M насамперед визначимо, які зі станів автомата досяжні з початкового (див. підрозд. 3.6.2):

$$\widetilde{Q}_0 = \{q_0\};$$
$$\widetilde{Q}_1 = \{q_0, q_2, q_3\};$$
$$\widetilde{Q}_2 = \{q_0, q_1, q_2, q_3, q_4\};$$
$$\widetilde{Q}_3 = \{q_0, q_1, q_2, q_3, q_4, q_5\}.$$

Отже, $\widetilde{Q}_3 = Q$, тобто всі стани автомата досяжні з початкового стану q_0, і можемо застосувати описану в цьому підрозділі процедуру побудови відношень «$\overset{k}{\sim}$» ($k \geqslant 0$). Відношення «$\overset{0}{\sim}$» містить два відношення еквівалентності:

$$Q/_{\overset{0}{\sim}} = \{\{q_0, q_3\}, \{q_1, q_2, q_4, q_5\}\}.$$

Поклавши $k = 0$, обчислимо відношення «$\overset{1}{\sim}$» за рекурентним співвідношенням (3.17). Для зручності зведемо значення функції переходів δ в окремі таблиці для кожного класу еквівалентності за відношенням «$\overset{0}{\sim}$» (див. табл. 3.4 і 3.5).

Таблиця 3.4

	q_0	q_3
a	q_3	q_3
b	q_2	q_4

Таблиця 3.5

	q_1	q_2	q_4	q_5
a	q_1	q_4	q_2	q_5
b	q_0	q_1	q_5	q_3

Із табл. 3.4 отримуємо, що $q_0 \overset{1}{\sim} q_3$, оскільки

$$\delta(q_0, a) = q_3 \overset{0}{\sim} \delta(q_3, a) = q_3; \quad \delta(q_0, b) = q_2 \overset{0}{\sim} \delta(q_3, b) = q_4.$$

З табл. 3.5 аналогічно отримуємо, що $q_1 \overset{1}{\sim} q_5$, $q_2 \overset{1}{\sim} q_4$, однак $q_1 \overset{1}{\not\sim} q_2$, оскільки $\delta(q_1, b) = q_0 \overset{0}{\not\sim} \delta(q_2, b) = q_1$. Отже, відношення «$\overset{1}{\sim}$» містить три класи еквівалентності:

$$Q/_{\overset{1}{\sim}} = \{\{q_0, q_3\}, \{q_1, q_5\}, \{q_2, q_4\}\}.$$

Поклавши $k = 1$, обчислимо відношення «$\overset{2}{\sim}$» за рекурентним співвідношенням (3.17). Для зручності зведемо значення функції переходів δ в окремі таблиці для кожного класу еквівалентності за відношенням «$\overset{1}{\sim}$» (табл. 3.6, 3.7 і 3.8).

Таблиця 3.6

	q_0	q_3
a	q_3	q_3
b	q_2	q_4

Таблиця 3.7

	q_1	q_5
a	q_1	q_5
b	q_0	q_3

Таблиця 3.8

	q_2	q_4
a	q_4	q_2
b	q_1	q_5

Бачимо, що $q_0 \overset{2}{\sim} q_3$, $q_1 \overset{2}{\sim} q_5$, $q_2 \overset{2}{\sim} q_4$, тобто відношення «$\overset{1}{\sim}$» та «$\overset{2}{\sim}$» збігаються, а отже за лемою 3.9 збігаються з відношенням еквівалентності станів «$\sim$»:

$$Q/_{\sim} = Q/_{\overset{2}{\sim}} = \{\{q_0, q_3\}, \{q_1, q_5\}, \{q_2, q_4\}\}.$$

Об'єднуючи еквівалентні стани автомата M за процедурою (3.9), отримуємо зображений на рис. 3.31 мінімальний детермінований скінченний автомат M_{merge}, еквівалентний автомату M.

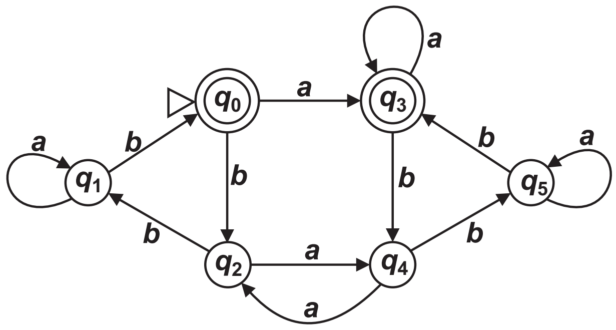

Рис. 3.30

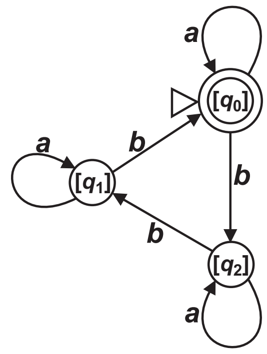

Рис. 3.31

Очевидно, що отриманий мінімальний автомат (а отже, й еквівалентний йому вихідний автомат M) допускає мову $\{w \in \{a,b\}^* : |w|_b \vdots 3\}$, що нелегко побачити безпосередньо з автомата M.

Детальніше про алгоритм побудови відношень k-еквівалентності, а також про його модифікації, див., наприклад, [4].

Послідовне заповнення таблиці нееквівалентних станів. Алгоритм ґрунтується на тривіальному факті, який негайно випливає із визначення відношення k-еквівалентності (див. також лему 3.8): якщо $q_1, q_2 \in Q$, $a \in T$, і стани $p_1 = \delta(q_1, a)$ та $p_2 = \delta(q_2, a)$ розрізняються деяким словом $w \in T^*$ довжиною k, то стани q_1 і q_2 розрізняються словом aw довжиною $k+1$.

Вважатимемо, що $|Q| \geqslant 2$ (якщо автомат містить лише один стан, мінімізація, очевидно, неможлива). Також вважатимемо, що всі стани автомата M досяжні з початкового – інакше застосовуємо описану у підрозд. 3.6.2 процедуру видалення станів, які недосяжні з початкового, отримуючи детермінований скінченний автомат, еквівалентний заданому.

Запишемо рекурсивний алгоритм, який послідовно для $k \geqslant 0$ визначає пари станів, які розрізняються деяким словом довжиною не більше ніж k (не є k-еквівалентними):

1. Покласти $A_0 = \{\{q_1, q_2\} : q_1 \in F, q_2 \in Q \setminus F\}$.
2. Покласти $k = 0$.
3. Якщо $k = |Q| - 2$, закінчити роботу.
4. Покласти

$$A_{k+1} = A_k \cup \{\{q_1, q_2\} : \exists\, q_1 \in Q, q_2 \in Q, a \in T : \{\delta(q_1, a), \delta(q_2, a)\} \in A_k\}.$$

5. Якщо $A_{k+1} = A_k$, закінчити роботу.
6. Збільшити k на 1 і перейти до п. 3.

Вправа 3.20. Довести, що кожна множина A_k ($k \geqslant 0$) містить такі й тільки такі пари станів $\{q_1, q_2\}$, які розрізняються деяким словом довжиною не більш ніж k.

Вказівка. Скористатися індукцією за номером $k \geqslant 0$, крок індукції обґрунтувати за допомогою леми 3.8.

Отже, кожна множина A_k ($k \geqslant 0$) фактично визначає відношення «$\overset{k}{\not\sim}$». Таким чином, згідно з лемою 3.9 та наслідком з леми 3.10 остання обчислена наведеним алгоритмом множина A_k збігається з множиною всіх пар нееквівалентних станів, тобто визначає відношення нееквівалентності станів «$\not\sim$», а отже й відношення еквівалентності станів «$\sim$». Нарешті, отримавши фактор-множину $Q/_\sim$, визначаємо (за співвідношеннями (3.9)) автомат M_{merge}, який за теоремою 3.8 є мінімальним за кількістю станів детермінованим автоматом, еквівалентним вихідному автомату M.

Обчислюючи множини A_k ($k \geqslant 0$), зручно зводити дані про пари нееквівалентних станів у таблицю, рядки і стовпці якої пронумеровано станами автомата M; якщо на деякому кроці алгоритму отримано, що $q_1 \not\sim q_2$, у відповідну комірку ставимо символ «$\times$». Коли алгоритм закінчує роботу, незаповнені комірки таблиці відповідають парам еквівалентних станів, що визначає відношення «$\sim$». Описаний алгоритм називають *алгоритмом заповнення таблиці* (див. [14]).

Приклад 3.43. Застосуємо алгоритм заповнення таблиці до автомата M із прикладу 3.42 (рис. 3.30). Зазначимо, що всі стани автомата M досяжні з початкового стану q_0 (це було перевірено у прикладі 3.42).

У табл. 3.9 зведено дані про пари станів, які розрізняються порожнім словом, тобто такі пари $\{p, q\}$, що $p \in F$, $q \notin Q \setminus F$. Таким чином, множина A_0 містить 8 пар станів. Підкреслимо, що, враховуючи симетричність відношення « $\overset{k}{\not\sim}$ », достатньо заповнити лише «половину» таблиці: якщо $p \overset{k}{\sim} q$ (або $p \overset{k}{\not\sim} q$), то $q \overset{k}{\sim} p$ (відповідно $q \overset{k}{\not\sim} p$).

Таблиця 3.9

	q_0	q_1	q_2	q_3	q_4
q_1	$\times$				
q_2	$\times$				
q_3		$\times$	$\times$		
q_4	$\times$			$\times$	
q_5	$\times$			$\times$	

Поклавши $k = 0$, обчислимо множину A_1, яка, згідно з визначенням, містить усі пари, що входять в A_0, а також всі такі пари $\{p, q\}$, що $\{\delta(p, x), \delta(q, x)\} \in A_0$ для деякого $x \in \{a, b\}$. Для кожної пари $\{\widetilde{p}, \widetilde{q}\} \in A_0$ і кожного символа $x \in \{a, b\}$ визначимо набір пар $\{p, q\}$, таких, що $\{\delta(p, x), \delta(q, x)\} = \{\widetilde{p}, \widetilde{q}\}$ (табл. 3.10).

Таблиця 3.10

	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_4\}$	$\{q_0, q_5\}$	$\{q_1, q_3\}$	$\{q_2, q_3\}$	$\{q_3, q_4\}$	$\{q_3, q_5\}$
a					$\{q_1, q_0\}$ $\{q_1, q_3\}$	$\{q_4, q_0\}$ $\{q_4, q_3\}$	$\{q_0, q_2\}$ $\{q_3, q_2\}$	$\{q_0, q_5\}$ $\{q_3, q_5\}$
b	$\underline{\{q_1, q_2\}}$	$\{q_1, q_0\}$	$\{q_1, q_3\}$	$\underline{\{q_1, q_4\}}$	$\underline{\{q_2, q_5\}}$	$\{q_0, q_5\}$	$\{q_5, q_3\}$	$\underline{\{q_5, q_4\}}$

Так, зафіксувавши пару станів $\{q_0, q_1\} \in A_0$ та $a \in T$, не отримуємо жодної «нової» пари нееквівалентних станів (немає жодного стану $p \in Q$, такого, що $\delta(p, a) = q_0$). Однак для $\{q_0, q_1\} \in A_0$ та $b \in T$ отримуємо «нову» пару $\{q_1, q_2\}$, оскільки $\delta(q_1, b) = q_0$, $\delta(q_2, b) = q_1$ (пара $\{q_1, q_2\} \in A_1$ справді є новою, оскільки не входить в A_0). Для $\{q_0, q_2\} \in A_0$ та $b \in T$ отримуємо також одну «нову» пару $\{q_1, q_0\} \in A_1$ (зазначимо, що ця пара належить і до A_0). Отримані дані про стани, що розрізняються словом довжиною не більш ніж 1, зведено в табл. 3.11. Бачимо, що множина A_1 містить 4 пари станів, які не входять в A_0 (підкреслені в табл. 3.10), тобто розрізняються словом довжиною 1, але не розрізняються словом ε.

Таблиця 3.11

	q_0	q_1	q_2	q_3	q_4
q_1	$\times$				
q_2	$\times$	$\times$			
q_3		$\times$	$\times$		
q_4	$\times$	$\times$		$\times$	
q_5	$\times$		$\times$	$\times$	$\times$

Поклавши $k = 1$, обчислимо множину A_2, яка згідно з визначенням містить всі пари, що входять в A_1, а також всі такі пари $\{p, q\}$, що $\{\delta(p, x), \delta(q, x)\} \in A_1$ для деякого $x \in \{a, b\}$. Для кожної пари

$\{\widetilde{p}, \widetilde{q}\} \in (A_1 \setminus A_0)$ і кожного символа $x \in \{a, b\}$ визначимо набір пар $\{p, q\}$, таких, що $\{\delta(p, x), \delta(q, x)\} = \{\widetilde{p}, \widetilde{q}\}$ (табл. 3.12). Нагадаємо, що для $\{\widetilde{p}, \widetilde{q}\} \in A_0$ та $x \in \{a, b\}$ такі списки пар станів визначено на попередньому кроці (табл. 3.10).

Таблиця 3.12

	$\{q_1, q_2\}$	$\{q_1, q_4\}$	$\{q_2, q_5\}$	$\{q_4, q_5\}$
a	$\{q_1, q_4\}$	$\{q_1, q_2\}$	$\{q_4, q_5\}$	$\{q_2, q_5\}$
b	$\{q_2, q_0\}$	$\{q_2, q_3\}$	$\{q_0, q_4\}$	$\{q_3, q_4\}$

Бачимо, що жодної нової пари нееквівалентних станів не отримано, тобто $A_2 = A_1$. Із табл. 3.11 встановлюємо, що автомат M має три різні пари еквівалентних станів, тобто множина Q розбита відношенням еквівалентності класів «$\sim$» на три класи:

$$Q/_{\sim} = \{\{q_0, q_3\}, \{q_1, q_5\}, \{q_2, q_4\}\},$$

що збігається с результатом, отриманим у прикладі 3.42. Мінімальний детермінований автомат M_{merge}, еквівалентний заданому, містить три стани, його граф показано на рис. 3.31.

Детальніше про алгоритм заповнення таблиці нееквівалентних станів див. [14].

Зауваження 3.18. Розглянуті алгоритми дозволяють знаходити мінімальний за кількістю станів **детермінований** скінченний автомат, еквівалентний заданому. Однак наведені алгоритми не можна застосувати для пошуку мінімального за кількістю станів **довільного** скінченного автомата, еквівалентного заданому (навіть якщо заданий автомат є детермінованим); див. контрприклад у [14]. Детальніше про мінімізацію недетермінованих скінченних автоматів див., наприклад, [20, 21].

Вправа 3.21. Для автоматів із прикладів 3.33, 3.38–3.40 побудувати мінімальний детермінований автомат двома способами: послідовною побудовою відношень k-еквівалентності та алгоритмом заповнення таблиці.

3.8. Основні властивості регулярних мов

3.8.1. Властивості замкнености класу регулярних мов

Означення 3.9. Нехай f – n-арна операція на непорожній множині X, тобто $f : \underbrace{X \times X \times \cdots \times X}_{n} \to X$. Множину $Y \subset X$ називають замкненою відносно операції f, якщо $f(x_1, x_2, \ldots, x_n) \in Y$ для будь-яких $x_1, x_2, \ldots, x_n \in Y$ (детально про алгебричні структури див., наприклад, [7]).

Лема 3.12. *Будь-який скінченний автомат еквівалентний деякому скінченному автомату з* ε*-переходами, який містить в точності один початковий стан і в точності один допускаючий стан, та початковий стан не збігається з допускаючим.*

Доведення. Для скінченного автомата $M = \langle Q, T, \Delta, I, F\rangle$ виберемо довільні $p, q \notin Q$. Тоді автомат M еквівалентний автомату

$$\langle Q \cup \{p, q\}, T, \Delta \cup \{(p, \varepsilon, q_0) \colon q_0 \in I\} \cup \{(q_1, \varepsilon, q) \colon q_1 \in F\}, \{p\}, \{q\}\rangle. \qquad \square$$

Теорема 3.10. *Нехай* L_1 *та* L_2 *– регулярні мови над алфавітом* T*. Тоді* $L_1 \cup L_2$*,* $L_1 \cap L_2$*,* $L_1 \cdot L_2$*,* $\overline{L_1}$*,* L_1^* *та* L_1^R *– регулярні мови над алфавітом* T*. (Клас регулярних мов замкнений відносно операцій об'єднання, перетину, конкатенації, доповнення, замикання Кліні та обертання.)*

Доведення. **Об'єднання.** Нехай мови L_1 та L_2 допускаються скінченними автоматами $\langle Q_1, T, \Delta_1, I_1, F_1\rangle$ та $\langle Q_2, T, \Delta_2, I_2, F_2\rangle$ відповідно, та $Q_1 \cap Q_2 = \varnothing$. Тоді мову $L_1 \cup L_2$ допускає скінченний автомат $\langle Q_1 \cup Q_2, T, \Delta_1 \cup \Delta_2, I_1 \cup I_2, F_1 \cup F_2\rangle$.

Конкатенація. Нехай мови L_1 та L_2 допускаються скінченними автоматами $\langle Q_1, T, \Delta_1, \{p_1\}, \{q_1\}\rangle$ та $\langle Q_2, T, \Delta_2, \{p_2\}, \{q_2\}\rangle$ відповідно (за лемою 3.12 такі автомати існують), та $Q_1 \cap Q_2 = \varnothing$. Тоді мову $L_1 \cdot L_2$ допускає скінченний автомат $\langle Q_1 \cup Q_2, T, \Delta_1 \cup \Delta_2 \cup \{(q_1, \varepsilon, p_2)\}, \{p_1\}, \{q_2\}\rangle$.

Доповнення. Нехай мову L_1 допускає детермінований скінченний автомат $\langle Q_1, T, \Delta_1, \{p_1\}, F\rangle$ (за теоремою 3.1 такий автомат існує). Тоді мову $\overline{L_1}$ допускає автомат $\langle Q_1, T, \Delta_1, \{p_1\}, Q \setminus F\rangle$.

Перетин. За законом де Моргана (див., наприклад, [7, 8]) маємо рівність $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$. Оскільки, за вже доведеним, клас регулярних мов над алфавітом T замкнений відносно об'єднання та доповнення, мова $L_1 \cap L_2$ також є регулярною.

Замикання Кліні. Нехай мова L_1 допускається скінченним автоматом $\langle Q_1, T, \Delta_1, \{p_1\}, \{q_1\}\rangle$ (за лемою 3.12 такий автомат існує). Виберемо $q_* \notin Q_1$. Тоді мову L_1^* допускає автомат

$$\langle Q_1 \cup \{q_*\}, T, \Delta_1 \cup \{(q_*, \varepsilon, p_1), (q_1, \varepsilon, q_*)\}, \{q_*\}, \{q_*\}\rangle.$$

Обертання. Нехай мова L_1 допускається скінченним автоматом $\langle Q_1, T, \Delta_1, I_1, F_1\rangle$. Тоді мову L_1^R допускає скінченний автомат

$$\langle Q_1, T, \{(q_2, a, q_1)\colon (q_1, a, q_2) \in \Delta_1\}, F_1, I_1\rangle. \qquad \square$$

Наслідок. *Множина регулярних мов над алфавітом T є булевою алгеброю з операціями «$\cup$», «$\cap$» та «$\bar{\ }$», нулем $\varnothing$ та одиницею T^*.*

Детально про булеві алгебри та алгебри множин див., наприклад, [22].

Приклад 3.44. Нехай $L_1 = \{a^n : n \geqslant 0\}$, $L_2 = \{b^n : n \geqslant 0\}$, $L_3 = \{c^n : n \geqslant 0\}$ – формальні мови над алфавітом $T = \{a, b, c\}$. Тоді скінченний автомат, що допускає мову $L_1 \cup L_2 \cup L_3$, можна отримати з автоматів, які допускають мови L_1, L_2 та L_3 (див. рис. 3.32).

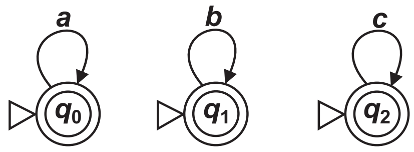

Рис. 3.32. Автомат, що допускає мову $\{a^n : n \geqslant 0\} \cup \{b^n : n \geqslant 0\} \cup \{c^n : n \geqslant 0\}$

Скінченний автомат із прикладу 3.15 (рис. 3.14), що допускає формальну мову $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\} = L_1 \cdot L_2 \cdot L_3$, також отримано з автоматів, які допускають мови L_1, L_2 та L_3. Цей автомат містить в точності один початковий та в точності один допускаючий стан, що дозволяє побудувати скінченний автомат, який допускає формальну мову $(L_1 \cdot L_2 \cdot L_3)^* = \{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^*$ (рис. 3.33). Отриманий автомат можна використати для побудови автомата, що допускає формальну мову $((L_1 \cdot L_2 \cdot L_3)^*)^R = ((L_1 \cdot L_2 \cdot L_3)^R)^* = \{c^n b^m a^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^*$ (рис. 3.34). Зазначимо, що, оскільки слова a, b та c належать обом мовам $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}$ та $\{c^n b^m a^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}$, замикання Кліні цих мов містять всі слова над алфавітом $T = \{a, b, c\}$:

$$\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^* = \{c^n b^m a^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^* = T^*.$$

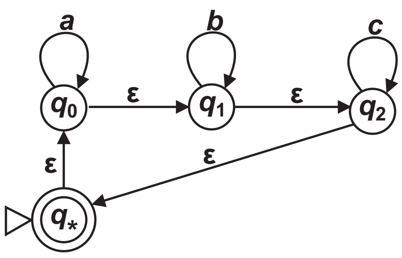

Рис. 3.33. Автомат, що допускає мову $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^*$

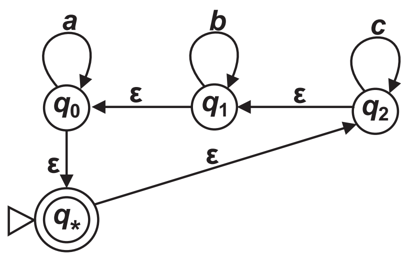

Рис. 3.34. Автомат, що допускає мову $\{c^n b^m a^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^*$

Скінченний автомат, зображений на рис. 3.14, не є детермінованим, а отже не може бути використаний для побудови доповнення до мови $L_1 \cdot L_2 \cdot L_3$. На рис. 3.35 зображено детермінований скінченний автомат, який допускає мову $L_1 \cdot L_2 \cdot L_3 = \{a^n b^m c^k \colon n \geqslant 0, m \geqslant 0, k \geqslant 0\}$, а також автомат, який допускає мову $\overline{\{a^n b^m c^k \colon n \geqslant 0, m \geqslant 0, k \geqslant 0\}}$.

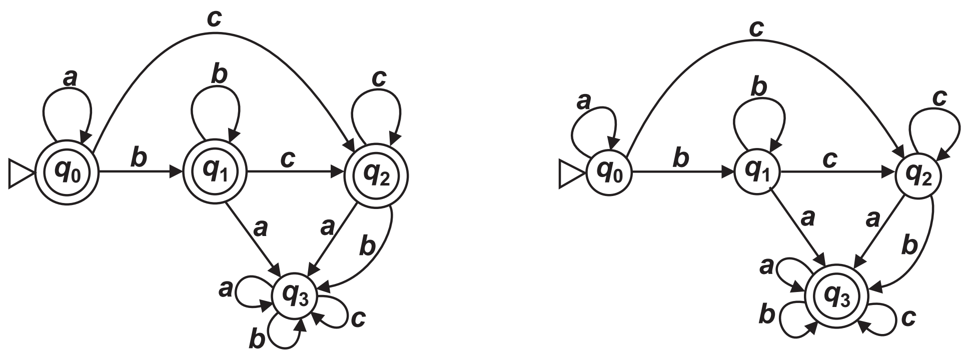

Рис. 3.35. Детерміновані автомати, що допускають мови $\{a^n b^m c^k \colon n \geqslant 0, m \geqslant 0, k \geqslant 0\}$ та $\overline{\{a^n b^m c^k \colon n \geqslant 0, m \geqslant 0, k \geqslant 0\}}$

Зауваження 3.19. Замкненість класу регулярних мов відносно перетину можна довести, безпосередньо побудувавши відповідний скінченний автомат: якщо мови L_1 та L_2 допускаються скінченними автоматами $\langle Q_1, T, \Delta_1, I_1, F_1\rangle$ та $\langle Q_2, T, \Delta_2, I_2, F_2\rangle$ без ε-переходів, то перетин $L_1 \cap L_2$ допускає автомат

$$\langle Q_1 \times Q_2, T, \Delta, I_1 \times I_2, F_1 \times F_2\rangle,$$

де $\Delta = \{((p_1, p_2), a, (q_1, q_2)) \colon (p_1, a, q_1) \in \Delta_1, (p_2, a, q_2) \in \Delta_2\}$.

3.8.2. Видалення ε-продукцій у регулярних граматиках

Теорема 3.11. *Будь-яка регулярна мова L породжується такою регулярною граматикою $G = \langle V, T, P, S\rangle$, що для всіх $A \in V$, $B \in V$, $x \in T$ виконуються умови:*

1) якщо $(A \to \varepsilon) \in P$, то $A = S$ (множина P містить не більш ніж одну ε-продукцію – $S \to \varepsilon$);

2) якщо $(A \to xB) \in P$, то $B \neq S$ (джерело S не входить до правої частини жодної продукції).

Доведення. Нехай мова L породжується деякою регулярною граматикою $G_0 = \langle V_0, T, P_0, S_0 \rangle$. Легко перевірити, що граматика G_0 еквівалентна граматиці $G_1 = \langle V, T, P_1, S \rangle$, де $V = V_0 \cup \{S\}$, $S \notin V_0$, $P_1 = P_0 \cup \{S \to \alpha \colon (S_0 \to \alpha) \in P_0\}$. Зазначимо, що P_1 не містить жодної продукції, права частина якої містила б нове джерело S. Однак множина P_1 може містити ε-продукції; більше того, P_1 може містити додаткову ε-продукцію $S \to \varepsilon$, якщо $(S_0 \to \varepsilon) \in P_0$.

Далі введемо $P_2 = P_1 \cup \{B \to x : (B \to xA) \in P_1, (A \to \varepsilon) \in P_1\}$. Граматика $G_2 = \langle V, T, P_2, S \rangle$ еквівалентна граматикам G_0 та G_1, оскільки застосування «нової» продукції $B \to x$ можна замінити послідовним застосуванням продукцій $B \to xA$ та $A \to \varepsilon$ із P_1.

Нарешті, можемо видалити всі ε-продукції, окрім $S \to \varepsilon$, оскільки застосування продукцій $B \to xA$ та $A \to \varepsilon$ ($x \in T$, $A \in V \setminus \{S\}$, $B \in V$) можна замінити застосуванням введеної продукції $B \to x$. Отже, отримуємо граматику $G = \langle V, T, P, S \rangle$, де $P = P_2 \setminus \{A \to \varepsilon \colon A \neq S\}$. Побудована граматика еквівалентна вихідній граматиці G, не містить ε-продукцій, окрім (можливо) продукції $S \to \varepsilon$, і праві частини продукцій множини P не містять джерела S. □

Наслідок. *Якщо мова L регулярна, то мова $L \setminus \{\varepsilon\}$ породжується регулярною граматикою без жодної ε-продукції.*

Зауваження 3.20. Нехай L – регулярна мова, яка породжена граматикою $\langle V, T, P, S \rangle$, що задовольняє умови теореми 3.11. Тоді мова $L \setminus \{\varepsilon\}$ породжується граматикою $\langle V, T, P \setminus \{S \to \varepsilon\}, S \rangle$. Зазначимо, що множина продукцій $P \setminus \{S \to \varepsilon\}$ справді не містить жодної ε-продукції.

Приклад 3.45. Розглянемо регулярну граматику

$$G_0 = \langle \{S_0, B\}, \{a, b\}, P_0, S_0 \rangle,$$

де $P_0 = \{S_0 \to aS_0 | bB | \varepsilon, B \to bB | \varepsilon\}$. Легко перевірити, що G_0 породжує мову $L = \{a^n b^m : n, m \geqslant 0\}$. Позбудемося ε-продукцій у граматиці G_0, використовуючи метод із доведення теореми 3.11 (єдино можлива ε-продукція, якщо залишиться, у лівій частині містить джерело, і джерело не може входити до правої частини жодної продукції нової граматики).

Ввівши новий нетермінальний символ S (джерело нової граматики), сформуємо множину P_1:

$$P_1 = P_0 \cup \{S \to \alpha \colon (S_0 \to \alpha) \in P_0\} = P_0 \cup \{S \to aS_0 | bB | \varepsilon\}.$$

Легко перевірити, що граматика G_0 справді еквівалентна граматиці $G_1 = \langle \{S_0, B, S\}, \{a, b\}, P_1, S\rangle$.

Далі, побудуємо множину продукцій P_2, ввівши «обхідні» продукції для кожної ε-продукції із P_1:

$$\begin{aligned} P_2 = P_1 \cup \{A_2 \to x \colon (A_2 \to xA_1) \in P_1, (A_1 \to \varepsilon) \in P_1\} = \\ = P_1 \cup \{S_0 \to a, S \to a, S_0 \to b, B \to b, S \to b\} = \\ = \{S_0 \to aS_0|bB|a|b|\varepsilon, B \to bB|b|\varepsilon, S \to aS_0|bB|a|b|\varepsilon\}. \end{aligned}$$

Нарешті, видаливши ε-продукції (окрім $S \to \varepsilon$) із множини P_2, отримуємо шукану граматику $G = \langle \{S_0, B, S\}, \{a, b\}, P, S\rangle$, де

$$P = \{S_0 \to aS_0|bB|a|b, B \to bB|b, S \to aS_0|bB|a|b|\varepsilon\}.$$

Легко переконатись, що побудована граматика G справді еквівалентна вихідній граматиці G_0, тобто породжує мову $L = \{a^n b^m \colon n, m \geqslant 0\}$.

Вправа 3.22. Позбутись ε-продукцій у граматиках із прикладів 1.16 та 1.17, використовуючи метод із доведення теореми 3.11.

3.8.3. Лема про розростання для регулярних мов

Теорема 3.12 (Лема про розростання для регулярних мов)**.** *Нехай L – довільна регулярна мова над алфавітом T. Тоді існує така константа $n \geqslant 1$, що будь-яке слово $w \in L$ довжиною $|w| \geqslant n$ можна зобразити у вигляді $w = xuy$, де $x, y \in T^*$, $u \in T^+$, $|xu| \leqslant n$, так, що для всіх $i \geqslant 0$ виконується умова $xu^iy \in L$.*

Доведення. Згідно з теоремою 3.11 регулярна мова L породжується такою регулярною граматикою $\langle V, T, P, S\rangle$, що множина P не містить продукцій вигляду $A \to \varepsilon$ $(A \neq S)$ та $A \to aS$ $(A \in V,\ a \in T)$.

Зафіксуємо $n = |V| + 1$. Будь-яке слово $w = a_1a_2 \ldots a_N \in L$ $(a_k \in T, 1 \leqslant k \leqslant N)$ довжиною $|w| = N \geqslant n$ є непорожнім і може бути виведене із джерела S за N кроків:

$$A_0 \Rightarrow a_1A_1 \Rightarrow a_1a_2A_2 \Rightarrow \cdots \Rightarrow a_1a_2 \ldots a_{N-1}A_{N-1} \Rightarrow a_1a_2 \ldots a_N = w, \tag{3.18}$$

де $A_0 = S$, і множина P містить продукції $A_k \to a_{k+1}A_{k+1}$ $(0 \leqslant k \leqslant N-2)$ та $A_{N-1} \to a_N$. Оскільки виведення (3.18) містить $N \geqslant n = |V| + 1$

нетермінальних символів, принаймні два із перших n нетермінальних символів $A_0, A_1, \ldots, A_{n-1}$ мають збігатися, тобто $A_i = A_j = A$ для деяких $0 \leqslant i < j \leqslant n-1$. Таким чином, виведення (3.18) набуває вигляду

$$A_0 \overset{*}{\Rightarrow} a_1a_2\ldots a_iA \overset{*}{\Rightarrow} a_1a_2\ldots a_ia_{i+1}\ldots a_jA \overset{*}{\Rightarrow} w.$$

Позначивши $x = a_1a_2\ldots a_i$, $u = a_{i+1}a_{i+2}\ldots a_j$, $y = a_{j+1}a_{j+2}\ldots a_N$, перепишемо отримане виведення у вигляді

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xuy = w. \tag{3.19}$$

Із виведення (3.19) отримуємо наявність виведень $A \overset{*}{\Rightarrow} uA$ та $A \overset{*}{\Rightarrow} y$, звідки, з урахуванням виведення $A_0 \overset{*}{\Rightarrow} xA$, маємо виведення

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xu^2A \overset{*}{\Rightarrow} \cdots \overset{*}{\Rightarrow} xu^iA \overset{*}{\Rightarrow} xu^iy$$

для будь-якого $i \geqslant 0$. Оскільки за побудовою $|xu| = j \leqslant n-1 < n$, теорему повністю доведено. □

Зауваження 3.21. У формулюванні теореми 3.12 умову $|xu| \leqslant n$ можна замінити умовою $|uy| \leqslant n$, виділяючи у доведенні повторення $A_i = A_j$ серед n останніх кроків виведення (3.18), тобто за умови $N - n \leqslant i < j \leqslant N - 1$.

Зауваження 3.22. В літературі лему про розростання часто доводять за допомогою скінченного автомата (див., наприклад, [4, 9, 14]).

Приклад 3.46. Нехай L – довільна скінченна мова. Оскільки будь-яка скінченна мова є регулярною (див. вправу 1.3), для формальної мови L має виконуватися твердження теореми 3.12. Справді, фіксуючи $n = \max\{|w| \colon w \in L\} + 1$, отримуємо правдивість твердження теореми завдяки відсутності слів $w \in L$ довжиною $|w| \geqslant n$.

Одне з основних застосувань леми про розростання (теореми 3.12) – доведення нерегулярності заданої формальної мови.

Приклад 3.47. Доведемо, що мова $L = \{a^mb^m \colon m \geqslant 0\}$ нерегулярна. Припустимо, що L – регулярна мова, і n – додатна константа, існування якої постулюється лемою про розростання. Тоді для слова $a^nb^n \in L$ мають існувати такі $x, u, y \in \{a, b\}^*$, $u \neq \varepsilon$, що $w = xuy$, $xu^iy \in L$ для всіх

$i \geqslant 0$, та $|xu| \leqslant n$. Але тоді u містить лише символи a, тобто $|u|_b = 0$. Між тим $|xuy|_a = |xuy|_b$ та $|xu^2y|_a = |xu^2y|_b$, тобто

$$|x|_a + |u|_a + |y|_a = |x|_b + |u|_b + |y|_b, \;\; |x|_a + 2|u|_a + |y|_a = |x|_b + 2|u|_b + |y|_b,$$

звідки маємо рівність $|u|_a = |u|_b$. Отже, $|u| = |u|_a + |u|_b = 0$, тобто $u = \varepsilon$. Отримана суперечність доводить, що мова L нерегулярна.

Наголосимо, що лема про розростання надає лише необхідну, але не достатню умову регулярності, тобто твердження теореми 3.12 може виконуватися й для нерегулярних мов.

Приклад 3.48. Для формальної мови $L = L_0 \cup L_1 \cup L_2 \cup L_3$, де

$$L_0 = \{a^{j+1}b^m c^m d^{k+1} \colon j, m, k \geqslant 0\}, \;\; L_1 = \{a^{j+1}b^{m_1}c^{m_2} \colon j, m_1, m_2 \geqslant 0\},$$
$$L_2 = \{b^{m_1}c^{m_2}d^{k+1} \colon m_1, m_2, k \geqslant 0\}, \; L_3 = \{b^{m_1}c^{m_2} \colon m_1, m_2 \geqslant 0\},$$

можна вибрати константу $n = 1$; довільне $w \in L$ довжиною $|w| \geqslant n = 1$ можна зобразити у вигляді $w = xuy$ із $x = \varepsilon$, $|u| = 1$, тобто за u взяти перший символ слова w. Легко перевірити, що цей розклад в усіх чотирьох випадках $w \in L_s$ $(s = 0, 1, 2, 3)$ відповідає твердженню теореми 3.12:

1) якщо $w = aa^j b^m c^m d^{k+1} = \varepsilon \cdot a \cdot a^j b^m c^m d^{k+1} \in L_0$, то для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot a^i \cdot a^j b^m c^m d^{k+1} \in L_0 \cup L_2$;

2) якщо $w = aa^j b^{m_1} c^{m_2} = \varepsilon \cdot a \cdot a^j b^{m_1} c^{m_2} \in L_1$, то для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot a^i \cdot a^j b^{m_1} c^{m_2} \in L_1 \cup L_3$;

3) нехай $w = b^{m_1}c^{m_2}d^{k+1} \in L_2$; якщо $m_1 \geqslant 1$, то $w = \varepsilon \cdot b \cdot b^{m_1-1}c^{m_2}d^{k+1}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot b^i \cdot b^{m_1-1}c^{m_2}d^{k+1} \in L_2$; якщо $m_1 = 0$, $m_2 \geqslant 1$, то $w = \varepsilon \cdot c \cdot c^{m_2-1}d^{k+1}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot c^i \cdot c^{m_2-1}d^{k+1} \in L_2$; якщо $m_1 = m_2 = 0$, то $w = \varepsilon \cdot d \cdot d^k$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot d^i \cdot d^k \in L_2$;

4) нехай $w = b^{m_1}c^{m_2} \in L_3$; якщо $m_1 \geqslant 1$, то $w = \varepsilon \cdot b \cdot b^{m_1-1}c^{m_2}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot b^i \cdot b^{m_1-1}c^{m_2} \in L_3$; якщо $m_1 = 0$, $m_2 \geqslant 1$, то $w = \varepsilon \cdot c \cdot c^{m_2-1}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot c^i \cdot c^{m_2-1} \in L_2$.

Зазначимо, що в разі заміни умови $|xu| \leqslant n$ умовою $|uy| \leqslant n$ (див. зауваження 3.21) необхідний розклад $w = xuy$ для будь-якого $w \in L$ довжиною $|w| \geqslant 1$ також існує: достатньо взяти $y = \varepsilon$, $|u| = 1$, тобто u є останнім символом слова w.

Отже, задана мова L задовольняє всі умови леми про розростання, однак ця мова нерегулярна. Справді, якщо б L була регулярною мовою, то, оскільки мова $\{ab^{m_1}c^{m_2}d : m_1, m_2 \geqslant 1\}$ регулярна, мова $L \cap \{ab^{m_1}c^{m_2}d\colon m_1, m_2 \geqslant 1\} = \{ab^m c^m d\colon m \geqslant 1\}$ згідно з теоремою 3.10 також була б регулярною. Однак нерегулярність мови $\{ab^m c^m d\colon m \geqslant 1\}$ неважко встановити лемою про розростання, аналогічно доведенню нерегулярності мови $\{a^m b^m\colon m \geqslant 0\}$ (див. приклад 3.47).

Приклад 3.49. Доведемо, що мова $L = \{a^q\colon q$ – просте число$\}$ нерегулярна. Припустимо, що L – регулярна мова і n – додатна константа, існування якої постулюється лемою про розростання. Виберемо слово $w = a^n$; тоді мають існувати такі $x, u, y \in \{a\}^*$, $u \neq \varepsilon$, що $w = xuy$, $xu^i y \in L$ для всіх $i \geqslant 0$, та $|xu| \leqslant n$. Розглянемо два випадки.

Перший випадок: $n = 1$. Оскільки $u \neq \varepsilon$, то $u = a$, звідки $x = \varepsilon$, $y = \varepsilon$. Покладаємо $i = 4$: $xu^i y = a^4 \in L$, що суперечить простоті числа 4.

Другий випадок: $n \geqslant 2$. Тоді $w = a^n = a^{|x|}a^{|u|}a^{n-|x|-|u|}$, де $|u| \geqslant 1$. Покладаємо $i = n + 1$, тоді

$$xu^i y = a^{|x|}(a^{|u|})^{n+1}a^{n-|x|-|u|} = a^{|x|+|u|(n+1)+n-|x|-|u|} = a^{n(|u|+1)} \in L.$$

Але число $n(|u| + 1)$ складене, оскільки $n \geqslant 2$ та $|u| + 1 \geqslant 2$. Отримана суперечність доводить нерегулярність мови L. Зазначимо без доведення, що мова L не контекстно-вільна (зауваження 4.17), але контекстно-залежна (вправа 5.4).

Приклад 3.50. Доведемо, що мова $L = \{a^{m^2}\colon m \geqslant 1\}$ нерегулярна. Припустимо, що L – регулярна мова і n – додатна константа, існування якої постулюється лемою про розростання. Виберемо слово $w = a^{n^2}$; тоді мають існувати такі $x, u, y \in \{a\}^*$, $u \neq \varepsilon$, що $w = xuy$, $|xu| \leqslant n$ та $\widetilde{w} = xu^3 y \in L$. Тоді $|x| + |u| + |y| = n^2$, а довжина слова

$$|\widetilde{w}| = |x| + 3|u| + |y| = |x| + |u| + |y| + 2|u| = n^2 + 2|u|$$

має бути квадратом деякого числа. Але з нерівності $1 \leqslant |u| \leqslant n$ випливає оцінка $n^2 < n^2 + 2|u| \leqslant n^2 + 2n < (n + 1)^2$, тобто між числами n^2 та $(n + 1)^2$ міститься квадрат деякого числа. Отримана суперечність доводить нерегулярність мови L. Зазначимо без доведення, що мова L не контекстно-вільна (зауваження 4.17), але контекстно-залежна (вправа 5.5).

Використовуючи метод доведення леми про розростання, доведемо корисний критерій скінченності мови, породженої регулярною граматикою.

Лема 3.13. *Нехай формальна мова L породжена регулярною граматикою $\langle V, T, P, S\rangle$, такою, що множина P не містить продукцій вигляду $A \to \varepsilon$ ($A \neq S$) та $A \to aS$ ($A \in V$, $a \in T$). Тоді необхідною і достатньою умовою нескінченності мови L є існування такого слова $w \in L$, що $|V| + 1 \leqslant |w| \leqslant 2|V|$.*

Доведення. **Необхідність.** Нехай мова L містить слово w довжиною $|V| + 1 \leqslant |w| \leqslant 2|V|$, виведене із джерела $S = A_0$ за N кроків:

$$A_0 \Rightarrow a_1 A_1 \Rightarrow \cdots \Rightarrow a_1 a_2 \dots a_{N-1} A_{N-1} \Rightarrow a_1 a_2 \dots a_N = w. \tag{3.20}$$

Оскільки $N = |w| \geqslant |V| + 1$, принаймні два з нетермінальних символів $A_0, A_1, \dots, A_{N-1}$ мають збігатися, тобто $A_i = A_j = A$ для деяких $0 \leqslant i < j$. Таким чином, виведення (3.20) набуває вигляду

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xuy = w, \tag{3.21}$$

де $x = a_1 a_2 \dots a_i$, $u = a_{i+1} a_{i+2} \dots a_j \neq \varepsilon$, $y = a_{j+1} a_{j+2} \dots a_N$. Із виведення (3.21) отримуємо наявність виведень $A \overset{*}{\Rightarrow} uA$ та $A \overset{*}{\Rightarrow} y$, звідки, з урахуванням виведення $A_0 \overset{*}{\Rightarrow} xA$, маємо виведення

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xu^2 A \overset{*}{\Rightarrow} \cdots \overset{*}{\Rightarrow} xu^i A \overset{*}{\Rightarrow} xu^i y$$

для будь-якого $i \geqslant 0$. Таким чином, L містить всі слова $xu^i y$ ($i \geqslant 0$) і тому є нескінченною.

Достатність. Нехай мова L нескінченна. Враховуючи нескінченність мови L, можемо вибрати $w \in L$ довжиною $|w| \geqslant |V| + 1$. Нехай w виведене із джерела $S = A_0$ за N кроків:

$$A_0 \Rightarrow a_1 A_1 \Rightarrow \cdots \Rightarrow a_1 a_2 \dots a_{N-1} A_{N-1} \Rightarrow a_1 a_2 \dots a_N = w. \tag{3.22}$$

Оскільки $N = |w| \geqslant |V| + 1$, принаймні два з перших $|V| + 1$ нетермінальних символів $A_0, A_1, \dots, A_{|V|}$ мають збігатися, тобто $A_i = A_j = A$ для деяких $0 \leqslant i < j \leqslant |V|$ ($i \neq 0$, оскільки $S = A_0$ не входить до правої частини жодної продукції). Таким чином, виведення (3.21) набуває вигляду

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xuy = w, \tag{3.23}$$

де $x = a_1a_2 \ldots a_i$, $u = a_{i+1}a_{i+2} \ldots a_j \neq \varepsilon$, $y = a_{j+1}a_{j+2} \ldots a_N$. Із виведення (3.23), вилучивши частину $xA \overset{*}{\Rightarrow} xuA$, отримуємо виведення

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} w_1 = xy,$$

тобто $w_1 = xy \in L$, до того ж $|w| - |V| \leqslant |w_1| < |w|$.

Зазначимо, що отримане слово w_1 непорожнє: справді, множина P не містить ε-продукцій, окрім, можливо, $S \to \varepsilon$, а джерело S не входить до правої частини жодної продукції, тобто $A_k \neq S$ для жодного $k \geqslant 1$.

Далі, якщо $|w_1| \geqslant |V|+1$, у виведенні $S \overset{*}{\Rightarrow} w_1$ принаймні два з перших $|V| + 1$ нетермінальних символів мають збігатися. Вилучивши відповідну частину виведення, отримуємо виведення $S \overset{*}{\Rightarrow} w_2$, тобто $w_2 \in L$, причому $|w_1| - |V| \leqslant |w_2| < |w_1|$. Продовжуючи процес, отримуємо у мові L послідовність непорожніх слів $w_0 = w$, w_1, w_2, $\ldots$, w_k, $\ldots$, де $|w_k| - |V| \leqslant |w_{k+1}| < |w_k|$. Очевидно, що послідовність має містити таке слово $w_m \in L$, що $|w_m| \leqslant |V|$, але $|w_{m-1}| \geqslant |V| + 1$; інакше кажучи, $m = \min\{k \in \mathbb{N} : |w_k| \leqslant |V|\}$. Тоді, оскільки $|w_{m-1}| \leqslant |w_m| + |V|$, для довжини слова w_{m-1} отримуємо оцінку $|V| + 1 \leqslant |w_{m-1}| \leqslant 2|V|$. □

Приклад 3.51. Регулярна граматика $\langle\{S, A, B\}, \{a, b, c, d\}, P, S\rangle$, де $P = \{S \to aA, A \to bB|c, B \to bA|d\}$, породжує формальну мову $L = \{ab^{2n}c : n \geqslant 0\} \cup \{ab^{2n+1}d : n \geqslant 0\}$. Оскільки ця мова нескінченна, має існувати принаймні одне слово $w \in L$ довжиною $3 + 1 \leqslant |w| \leqslant 2 \cdot 3$. Справді, мова містить три таких слова: ab^2c, ab^4c та ab^3d.

Якщо вилучити із множини P продукцію $B \to bA$, отримаємо граматику, яка породжує скінченну мову $\{ac, abd\}$; ця мова не містить жодного слова довжиною $4 \leqslant |w| \leqslant 6$, що узгоджується з твердженням леми 3.13.

Лема про розростання дає можливість встановити цікавий факт щодо побудови множини довжин слів фіксованої регулярної мови. Наведемо відповідне твердження без доведення (див. також [9, 23]).

Лема 3.14. *Для будь-якої регулярної мови L існує скінченний набір невід'ємних цілих чисел a_1, a_2,…, a_n, d_1, d_2,…, d_n, такий, що*

$$\{|w| : w \in L\} = \bigcup_{k=1}^{n} X_{a_k,d_k},$$

де $X_{a,d} = \{a + jd : j \in \mathbb{N} \cup \{0\}\}$.

Фактично лема 3.14 стверджує, що довжини слів регулярної мови утворюють множину, яка є об'єднанням скінченної кількості арифметичних прогресій. Зазначимо, що скінченна множина також є об'єднанням арифметичних прогресій, оскільки $X_{a,0} = \{a\}$.

Приклад 3.52. Для регулярної мови $\{a^{4n}b^{6m} : n \geqslant 0, m \geqslant 0\}$ отримуємо множину довжин

$$\{4n + 6m \colon n \geqslant 0, m \geqslant 0\} = \{0\} \cup \{4 + 2k \colon k \geqslant 0\}.$$

Зазначимо, що лема 3.14 не є критерієм регулярності, оскільки може справджуватися і для нерегулярних мов.

Приклад 3.53. Для нерегулярної мови $\{a^m b^m : m \geqslant 0\}$ (див. приклад 3.47) множина довжин $\{2m \colon m \geqslant 0\}$ є арифметичною прогресією.

3.9. Регулярні вирази. Теорема Кліні

3.9.1. Алгебра регулярних виразів

Нехай T – фіксований термінальний алфавіт. Множину *регулярних виразів* над T (або просто *регулярних виразів*) визначають рекурсивно трьома умовами:

1. Символи 0, 1 та довільний символ $a \in T$ є регулярними виразами над T.

2. Якщо r та s – регулярні вирази, то $(r + s)$, $(r \cdot s)$, r^* – регулярні вирази над T.

3. Інших регулярних виразів над T немає.

Множину всіх регулярних виразів над T позначатимемо як $R[T]$.

Приклад 3.54. Якщо $T = \{a, b, c\}$, то записи b, $(a^* + (b \cdot c))$, $((a^* + b) \cdot c)$, $(0 + (a \cdot 1))$ є регулярними виразами. Записи $(a+)$, $(\cdot b)$, $(a + c$ не є регулярними виразами.

Надалі, щоб спростити запис, у виразах опускатимемо зовнішні дужки, що не містять додаткової інформації, проте неминуче з'являються, якщо у виразі є хоча б один символ «+» або «·». Так, замість виразу $(a + b)$ писатимемо $a + b$. Крім того, зважаючи на традиційну в арифметиці домовленість про пріоритети операцій, дужки у виразах вигляду $r + (s \cdot t)$ будемо опускати, тобто замість $r + (s \cdot t)$ писатимемо $r + s \cdot t$.

Нарешті, символ «$\cdot$», якщо це не викликає непорозумінь, взагалі опускатимемо, тобто замість $r + s \cdot t$ писатимемо $r + st$.

Із кожним регулярним виразом $r \in R[T]$ пов'язують формальну мову $L[r] \subset T^*$ за таким рекурсивним визначенням:

1. $L[0] = \varnothing$, $L[1] = \{\varepsilon\}$, $L[a] = \{a\}$ для $a \in T$.

2. $L[r + s] = (L[r]) \cup (L[s])$, $L[rs] = (L[r]) \cdot (L[s])$, $L[r^*] = (L[r])^*$ для $r, s \in R[T]$.

Приклад 3.55. Якщо $T = \{a, b, c\}$, то

$$L[a + bc] = \{a\} \cup (\{b\} \cdot \{c\}) = \{a, bc\};$$
$$L[a(b + c)^*] = \{a\} \cdot \{b, c\}^* = \{aw \colon w \in \{b, c\}^*\};$$
$$L[(aa)^*] = \{a^2\}^* = \{a^{2n} \colon n \geqslant 0\}.$$

Регулярні вирази $r, s \in R[T]$ називають *еквівалентними*, якщо відповідні формальні мови збігаються:

$$(r = s) \Leftrightarrow (L[r] = L[s]).$$

Наведемо декілька простих тотожностей (еквівалентностей) для регулярних виразів:

1) $r + s = s + r$;
2) $r + (s + t) = (r + s) + t$;
3) $r \cdot (s \cdot t) = (r \cdot s) \cdot t$;
4) $r \cdot (s + t) = r \cdot s + r \cdot t$, $(s + t) \cdot r = s \cdot r + t \cdot r$;
5) $0 + r = r + 0 = r$;
6) $0 \cdot r = r \cdot 0 = 0$;
7) $1 \cdot r = r \cdot 1 = r$.

Тотожності 1-3 очевидні. Тотожність $r \cdot (s + t) = r \cdot s + r \cdot t$ випливає із рівності відповідних формальних мов:

$$L[r \cdot (s + t)] = \{uw \colon u \in L[r], w \in (L[s] \cup L[t])\} =$$
$$= \{uw \colon u \in L[r], w \in L[s]\} \cup \{uw \colon u \in L[r], w \in L[t]\},$$
$$L[r \cdot s + r \cdot t] = \{uw \colon u \in L[r], w \in L[s]\} \cup \{uw \colon u \in L[r], w \in L[t]\}.$$

Тотожність $(s + t) \cdot r = s \cdot r + t \cdot r$ можна довести аналогічно.

Вправа 3.23. Довести тотожності 5-7 самостійно.

Вправа 3.24. Довести тотожність $0^* = 1$.

Зауваження 3.23. В алгебрі регулярних виразів часто вводять додаткову операцію $r^+ = r \cdot r^*$. Очевидно, що $L[r^+] = (L[r])^+$.

Із властивостей 1-7 випливає, що регулярні вирази утворюють ідемпотентне півкільце з одиницею. Детальніше про властивості регулярних виразів див., наприклад, [4, 9, 14]. Про методи доведення тотожностей в алгебрі регулярних виразів див., зокрема, [14].

Лема 3.15. *Мова $L[r]$ є регулярною над алфавітом T для будь-якого $r \in R[T]$.*

Доведення. Формальні мови $\{a\}$ для довільного $a \in T$, мова $\{\varepsilon\}$ та порожня мова $\varnothing$ є, очевидно, регулярними. Далі, якщо $r_1, r_2 \in R[T]$ і мови $L[r_1]$ та $L[r_2]$ регулярні, то мови $L[r_1 + r_2]$, $L[r_1 \cdot r_2]$ та $L[r_1^*]$ також регулярні за теоремою 3.10. Отже, регулярність мови $L[r]$ для довільного регулярного виразу $r \in R[T]$ можна встановити індукцією за загальною кількістю операцій «$+$», «$\cdot$» та «*» у виразі r. □

3.9.2. Узагальнений скінченний автомат. Теорема Кліні про регулярні вирази

Узагальненим скінченним автоматом називають впорядкований набір $M = \langle Q, T, \Delta, I, F\rangle$, де множина станів Q та алфавіт T – скінченні непорожні множини, відношення (множина) переходів $\Delta \subset (Q \times R[T] \times Q)$, $I \subset Q$ та $F \subset Q$ – множини початкових та допускаючих станів відповідно. Конфігурацією узагальненого автомата M назвемо довільний набір $(q, w) \in (Q \times T^*)$. На множині конфігурацій узагальненого автомата M визначимо бінарне відношення такту «$\vdash_M$»:

$$((q_1, w) \vdash_M (q_2, u)) \Leftrightarrow \exists\, r \in R[T] \colon \begin{cases} w = xu, \; x \in L[r]; \\ (q_1, r, q_2) \in \Delta, \end{cases}$$

де $q_1, q_2 \in Q$, $w \in T^*$, $u \in T^*$. Інакше кажучи, відношення «$\vdash_M$» містить пари конфігурацій вигляду $((q_1, xu), (q_2, u))$, де $x \in L[r]$ для деякого виразу $r \in R[T]$, такого, що $(q_1, r, q_2) \in \Delta$. Автомат M допускає (сприймає) слово $w \in T^*$, якщо $(q_0, w) \vdash_M^* (q, \varepsilon)$ для деяких $q_0 \in I$ та $q \in F$. Множину слів $L[M]$, які допускає узагальнений автомат M, називають формальною мовою, яку допускає (сприймає, розпізнає) узагальнений автомат M. Очевидно, що скінченний автомат, визначений в означенні 3.1, а також скінченний автомат з ε-переходами (див. означення 3.6), можна розглядати як узагальнені скінченні автомати: будь-який символ $a \in T$ за визначенням є регулярним виразом $a \in R[T]$, порожньому слову ε відповідає регулярний вираз 1.

Зазначимо, що два переходи $(q, r_1, p), (q, r_2, p) \in \Delta$ узагальненого автомата M можна об'єднати в один перехід $(q, r_1 + r_2, p)$, не змінюючи мову $L[M]$; надалі такі переходи називатимемо *паралельними*. Між тим, додавання переходу $(q, 0, p)$ також не змінить мову $L[M]$; надалі переходи вигляду $(q, 0, p)$ називатимемо *порожніми*. Отже, без втрати загальності можна вважати, що для будь-якої пари станів $q, p \in Q$ множина переходів Δ узагальненого автомата M містить в точності один перехід (q, r_{qp}, p). Підкреслимо, що переходи (q, r_{qp}, p) та (p, r_{pq}, q) не паралельні, і їх неможливо об'єднати.

Узагальнені скінченні автомати, аналогічно скінченним автоматам, визначеним в означенні 3.1, зручно зображати у вигляді графів.

Приклад 3.56. Зображені на рис. 3.36 та 3.37 узагальнені скінченні автомати над алфавітом $\{a, b, c\}$ допускають ту саму формальну мову $\{a(ab)^k,\ b(ab)^k : k \geqslant 0\} \cup \{c\}$, другий автомат отриманий з першого об'єднанням паралельних переходів (q_0, a, q_1) та (q_0, b, q_1) в один перехід $(q_0, a + b, q_1)$; нагадаємо, що регулярному виразу 1 відповідає порожнє слово (ε-перехід (q_1, ε, q_2)).

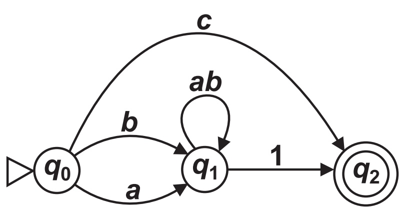

Рис. 3.36

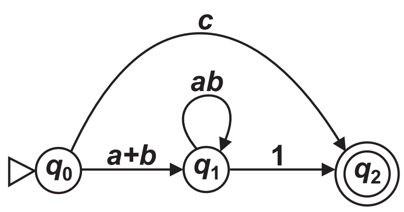

Рис. 3.37

Зазначимо, що мова $\{a(ab)^k,\ b(ab)^k : k \geqslant 0\} \cup \{c\} = (\{a, b\} \cdot \{ab\}^*) \cup \{c\}$ відповідає регулярному виразу $(a + b)(ab)^* + c$.

Лема 3.16. *Для будь-якої регулярної мови L над алфавітом T існує регулярний вираз $r \in R[T]$, такий, що $L = L[r]$.*

Доведення. Нехай L – регулярна мова над алфавітом T. Згідно з лемою 3.12 можна вважати, що L допускається скінченним автоматом $M = \langle Q, T, \Delta, \{q_0\}, \{q_f\} \rangle$ з ε-переходами, та початковий стан q_0 не збігається з допускаючим станом q_f. Скінченний автомат M розглядатимемо як узагальнений, який для кожної пари станів $q, p \in Q$ містить в точності один перехід (q, r_{qp}, p). Щоб знайти регулярний вираз, якому відповідає мова L, спростимо автомат M, видаливши із множини Q всі стани, окрім станів q_0 та q_f.

Зафіксуємо стан $t \in Q \setminus \{q_0, q_f\}$. Для кожної пари станів $q, p \in Q$ додамо до множини Δ перехід $(q, r_{qt}r_{tt}^* r_{tp}, p)$ і видалимо стан t (разом з усіма переходами, які містять t); легко зрозуміти, що мова $L[M]$ не зміниться, оскільки доданий перехід дублює застосування переходів (q, r_{qt}, t), (t, r_{tt}, t) та (t, r_{tp}, p) при переведенні автомата зі стану q у стан p через стан t. Об'єднавши паралельні переходи та повторивши описану процедуру для кожного стану $t \in Q \setminus \{q_0, q_f\}$, отримуємо автомат з двома станами q_0 і q_f та чотирма переходами (q_0, r_{q_0,q_0}, q_0), (q_0, r_{q_0,q_f}, q_f), (q_f, r_{q_f,q_f}, q_f), (q_f, r_{q_f,q_0}, q_0). Очевидно, що отриманий автомат допускає мову, яка відповідає регулярному виразу

$$r = (r_{q_0,q_0}^* r_{q_0,q_f} r_{q_f,q_f}^* r_{q_f,q_0})^* r_{q_0,q_0}^* r_{q_0,q_f} r_{q_f,q_f}^*, \tag{3.24}$$

тобто $L[M] = L[r]$. Зазначимо, що завжди можна знайти різні еквівалентні регулярні вирази для мови L. Зазвичай у практичних випадках легше спочатку видаляти стани t з порожніми петлями $(t, 0, t)$, а лише потім – з непорожніми. □

Теорема 3.13 (С. К. Кліні, не пізніше 1956 р.)**.** *Клас регулярних мов над алфавітом T збігається з класом мов $\{L[r] : r \in R[T]\}$.*

Доведення. Твердження теореми негайно випливає з лем 3.15 та 3.16. □

Наслідок. *Клас регулярних мов над алфавітом T є найменшою за відношенням вкладення («$\subset$») підмножиною множини 2^{T^*}, яка містить всі мови $\{a\}$ для довільного $a \in T$, мову $\{\varepsilon\}$, порожню мову $\varnothing$ та замкнена відносно об'єднання, конкатенації та замикання Кліні.*

Метод видалення станів, описаний у доведенні леми 3.16, можна використовувати для визначення регулярного виразу, якому відповідає задана регулярна мова. Зазначимо, що наявність порожніх переходів може суттєво спростити вираз (3.24).

Приклад 3.57. Користуючись методом видалення станів, визначимо регулярний вираз, якому відповідає регулярна мова

$$L = \{w \in \{a, b\}^* : |w|_a - |w|_b = 4k + 1,\ k \in \mathbb{Z}\};$$

термінальним алфавітом вважатимемо $T = \{a, b\}$. Для різниці $|w|_a - |w|_b$ можливі варіанти $4k$, $4k+1$, $4k+2$ та $4k+3$, де $k \in \mathbb{Z}$; автомат, який допускає мову L, має допустити слова з різницею $4k+1$ та не допустити слова з різницями $4k$, $4k+2$ та $4k+3$. Отже, для побудови автомата достатньо чотирьох станів q_0 (початковий), q_1, q_2 та q_3; $(q_0, w) \models^* (q_i, \varepsilon)$ тоді й тільки тоді, коли $|w|_a - |w|_b = 4k+i$ $(i = 0, 1, 2, 3)$. Граф такого автомата показано на рис. 3.38; його можна розглядати як узагальнений, порожні переходи з метою спрощення на рисунку не наведені.

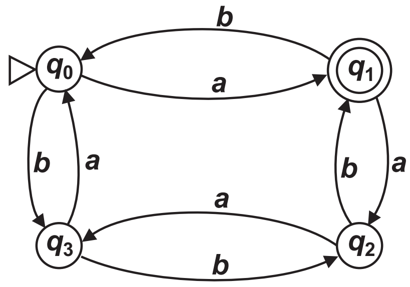

Рис. 3.38

Для видалення стану q_2 необхідно додати чотири переходи: (q_1, aa, q_3), (q_3, bb, q_1), (q_1, ab, q_1) та (q_3, ba, q_3) (враховуючи, що переходи $(q_2, 0, q_2)$, $(q_0, 0, q_2)$ та $(q_2, 0, q_0)$ порожні). Граф отриманого автомата наведено на рис. 3.39.

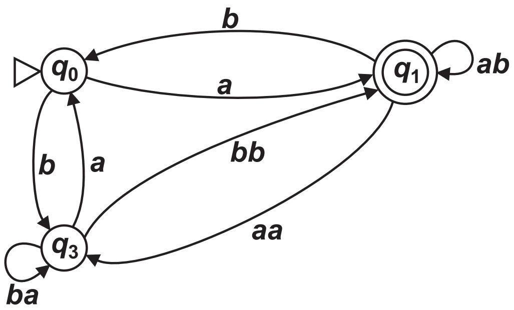

Рис. 3.39

Аналогічно, видаляємо стан q_3, додаючи переходи

$$(q_0, b(ba)^*a, q_0), (q_1, aa(ba)^*bb, q_1),$$
$$(q_0, b(ba)^*bb, q_1), (q_1, aa(ba)^*a, q_0).$$

Нарешті, об'єднуючи три останні переходи з існуючими паралельними (q_1, ab, q_1), (q_0, a, q_1) та (q_1, b, q_0) відповідно, отримуємо три переходи

$$(q_1, aa(ba)^*bb + ab, q_1),$$
$$(q_0, b(ba)^*bb + a, q_1),$$
$$(q_1, aa(ba)^*a + b, q_0).$$

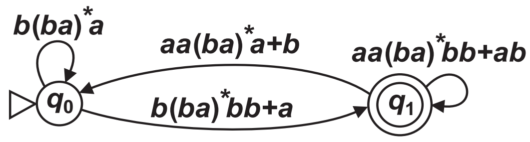

Рис. 3.40

Граф отриманого автомата зображено на рис. 3.40.

Користуючись формулою (3.24), можемо знайти регулярний вираз, якому відповідає мова L:

$$r_1 = \big((b(ba)^*a)^*(b(ba)^*bb + a)(aa(ba)^*bb + ab)^*(aa(ba)^*a + b)\big)^* \cdot$$
$$\cdot (b(ba)^*a)^*(b(ba)^*bb + a)(aa(ba)^*bb + ab)^*. \qquad (3.25)$$

Зауваження 3.24. В автоматі на рис. 3.39 можна замінити переходи (q_3, a, q_0) й (q_1, b, q_0) на чотири переходи (q_3, ab, q_3), (q_1, ba, q_1), (q_3, aa, q_1) й (q_1, bb, q_3) та об'єднати їх з існуючими паралельними переходами

(q_3, ba, q_3), (q_1, ab, q_1), (q_3, bb, q_1) та (q_1, aa, q_3) відповідно (рис. 3.41), а лише після цього видалити стан q_3 (рис. 3.42).

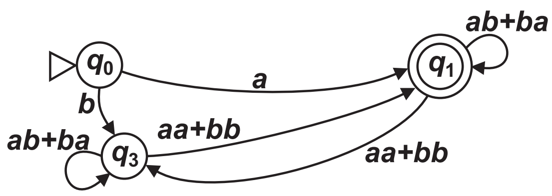

Рис. 3.41

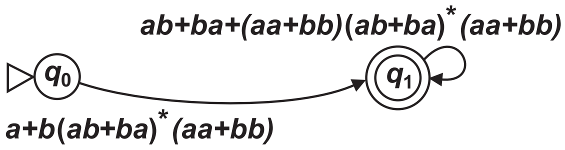

Рис. 3.42

У такому випадку отримуємо простіший регулярний вираз, якому відповідає мова L:

$$r_2 = \big(a+b(ab+ba)^*(aa+bb)\big)\big(ab+ba+(aa+bb)(ab+ba)^*(aa+bb)\big)^*. \quad (3.26)$$

Вправа 3.25. Визначити регулярні вирази, яким відповідають регулярні мови із прикладів 3.8 та 3.9.

Інший алгоритм знаходження регулярного виразу за скінченним автоматом методом рекурентних формул викладений, наприклад, у [24].

Зауваження 3.25. Для регулярного виразу r розглядають *висоту ітерації*, або *зіркову висоту* $\operatorname{sh} r$ – максимальну глибину вкладеності операцій замикання Кліні «*». Аналогічно, для регулярної мови L *висота ітерації (зіркова висота)* $\operatorname{sh} L$ – це найменша висота ітерації серед регулярних виразів, яким відповідає мова L (детальніше див. [25, 26]). У прикладі 3.57 висота регулярного виразу (3.25) $\operatorname{sh} r_1 = 3$, висота регулярного виразу (3.26) $\operatorname{sh} r_2 = 2$, а висота регулярної мови $\operatorname{sh} L = 2$, оскільки не існує регулярного виразу висотою 1, якому відповідає мова L (див. [26]). Зазначимо, що методи пошуку зіркової висоти регулярної мови, тобто мінімального за кількістю вкладень операції «*» регулярного виразу, відрізняються від методів пошуку мінімального за кількістю станів детермінованого скінченного автомата.

Запитання та завдання для самоконтролю

1. Навести неформальний опис та принцип роботи скінченного автомата. Які дії виконує скінченний автомат для переходу (q_1, α, q_2)?

2. Навести формальне означення скінченного автомата.

3. Навести означення часткового детермінованого скінченного автомата та детермінованого скінченного автомата. Яка, на вашу думку, перевага таких автоматів порівняно з недетермінованими, і навпаки?

4. Навести означення конфігурації та такту роботи скінченного автомата. Як вводять відношення «$\vdash_M^*$»? Яку конфігурацію називають тупиковою?

5. Дати визначення поняттю «автомат допускає формальну мову». Яку формальну мову називають автоматною? Які автомати називають еквівалентними?

6. Навести означення функції переходів та розширеної функції переходів детермінованого автомата, вказати відповідні приклади.

7. Сформулювати теорему про існування еквівалентного детермінованого автомата та навести відповідний метод побудови.

8. Чи збігаються класи автоматних та регулярних мов? Навести метод побудови скінченного автомата, який допускає мову, породжену регулярною граматикою; навести метод побудови регулярної граматики, яка породжує автоматну мову.

9. Навести означення скінченного автомата з ε-переходами. Яка, на вашу думку, перевага таких автоматів порівняно з автоматами без ε-переходів, і навпаки?

10. Навести метод видалення ε-переходів для скінченного автомата.

11. Навести означення множини правих контекстів слова відносно формальної мови.

12. Навести означення еквівалентності слів за формальною мовою та еквівалентності слів за скінченним автоматом. Як вводять індекси i_L та i_M, який зв'язок між ними? Вказати приклад автомата, для якого $i_L < i_M$.

13. Сформулювати теорему Майхілла–Нерода, навести відповідний метод побудови мінімального за кількістю станів скінченного детермінованого автомата, який допускає формальну мову. Яка, на вашу думку, перевага таких автоматів?

14. Які стани детермінованого скінченного автомата називають еквівалентними? Навести означення мінімального автомата, отриманого злиттям еквівалентних станів.

15. Навести алгоритм послідовної побудови відношень k-еквівалентності та алгоритм послідовного заповнення таблиці нееквівалентних ста-

нів для мінімізації скінченних автоматів.

16. Сформулювати властивості замкненості класу регулярних мов, навести методи побудови відповідних автоматів.

17. Сформулювати лему про розростання для регулярних мов.

18. Навести означення регулярного виразу. Описати метод видалення станів узагальненого скінченного автомата для знаходження регулярного виразу.

Розділ 4

Магазинні автомати і контекстно-вільні граматики

4.1. Дерева розбору. Однозначні контекстно-вільні мови

4.1.1. Дерева розбору

Нехай $G = \langle V, T, P, S\rangle$ – КВ-граматика (див. підрозд. 1.4). Структура продукцій у граматиках цього типу – $A \to \alpha$, де $A \in V$, $\alpha \in (V \cup T)^*$, – передбачає можливість заміни входження нетермінального символа A на слово α у будь-якому слові uAv незалежно від оточуючого контексту $u \in (V \cup T)^*$ та $v \in (V \cup T)^*$. Це означає, що символ A, який породжує слово $w \in T^*$ ($A \overset{*}{\underset{G}{\Rightarrow}} w$), може «розгортатися» в w незалежно від оточуючого контексту і не впливаючи на розгортання інших нетермінальних символів та інших входжень A.

Приклад 4.1. Розглянемо граматику $G = \langle\{S, A, B\}, \{a, b, c\}, P, S\rangle$ із множиною продукцій $P = \{S \to AAB, A \to aAc|\varepsilon, B \to bBc|\varepsilon\}$. Використовуючи техніку, описану в прикладі 1.13, п. 3 та 5, легко зрозуміти, що ця граматика породжує мову $L[G] = \{a^{k_1}c^{k_1}a^{k_2}c^{k_2}b^nc^n : k_1, k_2, n \geqslant 0\}$. Так, слово $w = a^2c^2acbc$ можна отримати із джерела S таким чином:

$$\begin{aligned} S \Rightarrow AAB \Rightarrow aAcAB \Rightarrow a^2Ac^2AB \Rightarrow a^2c^2AB \Rightarrow \\ \Rightarrow a^2c^2aAcB \Rightarrow a^2c^2acB \Rightarrow a^2c^2acbBc \Rightarrow a^2c^2acbc. \end{aligned} \tag{4.1}$$

Легко перевірити, що існує $\frac{7!}{3!2!2!} = 210$ варіантів виведення $S \overset{*}{\Rightarrow} w$, наведемо ще два:

$$S \Rightarrow AAB \Rightarrow AAbBc \Rightarrow AAbc \Rightarrow AaAcbc \Rightarrow \\ \Rightarrow Aacbc \Rightarrow aAcacbc \Rightarrow a^2Ac^2acbc \Rightarrow a^2c^2acbc; \tag{4.2}$$

$$S \Rightarrow AAB \Rightarrow aAcAB \Rightarrow aAcaAcB \Rightarrow aAcaAcbBc \Rightarrow \\ \Rightarrow a^2Ac^2aAcbBc \Rightarrow a^2c^2aAcbBc \Rightarrow a^2c^2aAcbc \Rightarrow a^2c^2acbc. \tag{4.3}$$

Проте всі варіанти виведення $S \overset{*}{\Rightarrow} w = a^2c^2acbc$ влаштовані за однаковою схемою: три нетермінальні символи, які складають отримане на першому кроці слово AAB (два входження символа A та символ B), розгортаються у слова a^2c^2, ac та bc відповідно:

$S \Rightarrow AAB$; $\quad A \Rightarrow aAc \Rightarrow aaAcc \Rightarrow aacc$;

$A \Rightarrow aAc \Rightarrow ac$;

$B \Rightarrow aBc \Rightarrow bc$.

Схему розгортання S в a^2c^2acbc наочно зображують у формі дерева розбору (рис. 4.1, формальне визначення див. далі в цьому підрозділі).

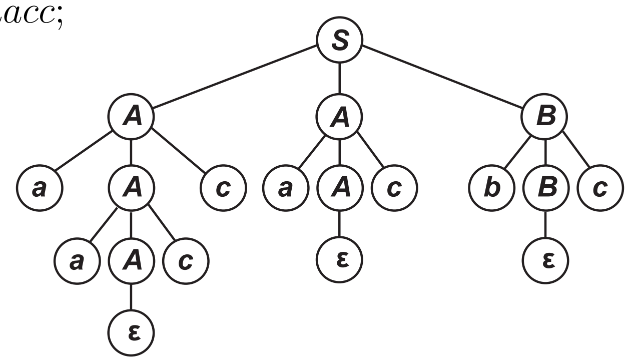

Рис. 4.1

Означення 4.1. Нехай $G = \langle V, T, P, S \rangle$ – КВ-граматика, $w \in T^*$, $A \in V$, та існує виведення $A \underset{G}{\overset{*}{\Rightarrow}} w$ із словом $\alpha \in (V \cup T)^*$, отриманим на першому кроці:

$$A \underset{G}{\Rightarrow} \alpha \underset{G}{\overset{*}{\Rightarrow}} w. \tag{4.4}$$

Деревом розбору (деревом виведення) слова w, що відповідає виведенню (4.4), називають мічене кореневе дерево, кожна вершина якого помічена символом із $V \cup T$ або міткою ε, і виконуються такі умови:

1) корінь дерева помічений символом A;

2) якщо $\alpha = \xi_1\xi_2 \ldots \xi_n \overset{*}{\Rightarrow} w$, де $\xi_j \in (V \cup T)$ $(j = 1, 2, \ldots, n, n \geqslant 1)$:

2.1) корінь має n безпосередніх нащадків;

2.2) безпосередні нащадки кореня помічені символами ξ_1, ξ_2, …, ξ_n, лінійно впорядковані згідно з порядком входження міток у слово α і розміщені на рисунку зліва направо згідно з цим впорядкуванням;

2.3) кожний безпосередній нащадок кореня, помічений нетермінальним символом $\xi_j = A_j \in V$, є коренем піддерева, яке відповідає розгортанню $A_j \overset{*}{\Rightarrow} w_j \in T^*$ у виведенні (4.4);

3) якщо $\alpha = \varepsilon$:

3.1) корінь має одного безпосереднього нащадка, який помічений міткою ε.

Відношення порядку для безпосередніх нащадків фіксованої вершини, встановлене у п. 2.2 означення 4.1, поширимо на всю множину вершин: якщо вершина u передує вершині v, кожний нащадок u передує кожному нащадку вершини v. Легко зрозуміти, що порівнянними є ті й тільки ті пари вершин, які не лежать на одній гілці дерева розбору. Зокрема, попарно порівнянними є усі вершини, які помічені термінальним символом або міткою ε – мітки цих вершин, розміщені згідно з побудованим відношенням порядку, утворюють виведене слово w. Так, для дерева розбору слова a^2c^2acbc у граматиці з прикладу 4.1 (рис. 4.1) отримуємо:

$$aa \cdot \varepsilon \cdot cca \cdot \varepsilon \cdot cb \cdot \varepsilon \cdot c = aaccacbc = a^2c^2acbc.$$

Означення 4.2. Виведення $S = \alpha_0 \underset{G}{\Rightarrow} \alpha_1 \underset{G}{\Rightarrow} \alpha_2 \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} \alpha_n = w$ слова w у КВ-граматиці G називають лівостороннім (правостороннім), якщо на кожному кроці $\alpha_i \underset{G}{\Rightarrow} \alpha_{i+1}$ $(i = 0, 1, \ldots, n-1)$ відбувається заміна першого зліва (справа) нетермінального символа у слові α_i.

Очевидно, що для будь-якого слова $w \in L[G]$ існує принаймні одне лівостороннє і принаймні одне правостороннє виведення, однак кожному дереву розбору відповідає в точності одне лівостороннє (і в точності одне правостороннє) виведення. Так, для слова a^2c^2acbc у граматиці з прикладу 4.1 виведення (4.1) лівостороннє, виведення (4.2) правостороннє, виведення (4.3) не є ані ліво-, ані правостороннім; усім цим трьом виведенням відповідає одне дерево розбору, зображене на рис. 4.1.

Зазначимо, що в загальному випадку для заданого слова може існувати декілька лівосторонніх (аналогічно – правосторонніх) виведень і, відповідно, декілька різних дерев розбору.

4.1.2. Однозначні КВ-граматики та КВ-мови

Означення 4.3. Контекстно-вільну граматику $G = \langle V, T, P, S\rangle$ називають однозначною, якщо для кожного слова $w \in L[G]$ існує в точності одне лівостороннє виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$.

Приклад 4.2. Контекстно-вільна граматика G з прикладу 4.1 однозначна, оскільки після першого обов'язкового кроку $S \Rightarrow AAB$ обидва входження символа A та символ B єдиним чином розгортаються відповідно в $a^{k_1}c^{k_1}$, $a^{k_2}c^{k_2}$ та b^nc^n $(k_1, k_2, n \geqslant 0)$, складаючи задане слово $a^{k_1}c^{k_1}a^{k_2}c^{k_2}b^nc^n \in L[G]$.

Приклад 4.3. Розглянемо граматику $G = \langle\{S\}, \{a, b\}, P, S\rangle$ із множиною продукцій $P = \{S \to aSbS|bSaS|\varepsilon\}$. Можна довести, що $L[G] = \{w \in \{a, b\}^* : |w|_a = |w|_b\}$. Очевидно, що для $aabb \in L$ існує єдине лівостороннє виведення із S:

$$S \Rightarrow aSbS \Rightarrow aaSbSbS \Rightarrow$$
$$\Rightarrow aabSbS \Rightarrow aabbS \Rightarrow aabb,$$

відповідне дерево розбору див. на рис. 4.2.

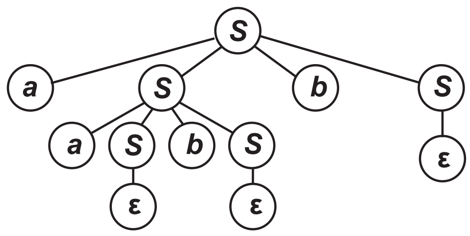

Рис. 4.2

Однак слово $abab$ можна отримати із S двома різними лівосторонніми виведеннями

$$S \Rightarrow aSbS \Rightarrow abSaSbS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab;$$
$$S \Rightarrow aSbS \Rightarrow abS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab,$$

яким відповідають різні дерева розбору (рис. 4.3 та 4.4 відповідно).

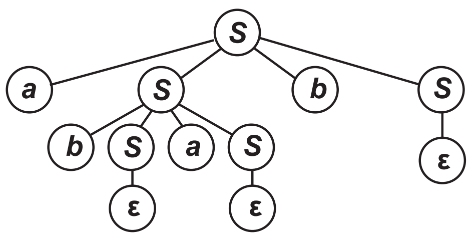

Рис. 4.3

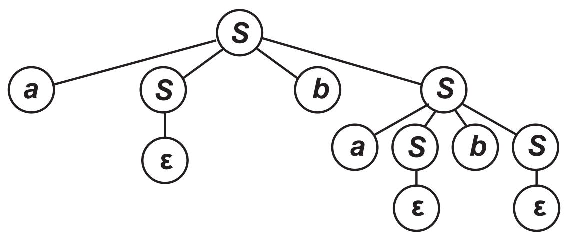

Рис. 4.4

Отже, КВ-граматика G неоднозначна, оскільки слово $abab$ можна отримати із S різними лівосторонніми виведеннями.

Вправа 4.1. Довести, що граматика G із прикладу 4.3 справді породжує мову $\{w \in \{a,b\}^*\colon |w|_a = |w|_b\}$.

Вказівка. Для довільного слова $w \in \{a,b\}^*$ довести логічні наслідки

$$(S \underset{G}{\overset{*}{\Rightarrow}} w) \Rightarrow (|w|_a = |w|_b),$$
$$(|w|_a = |w|_b) \Rightarrow (S \underset{G}{\overset{*}{\Rightarrow}} w);$$

в обох випадках скористатися індукцією за $|w| \geqslant 0$.

Вправа 4.2. Для граматики G із прикладу 4.3 та слова $aababb$ виписати всі лівосторонні виведення і зобразити відповідні дерева розбору.

Означення 4.4. Контекстно-вільну мову $L \subset T^*$ називають однозначною, якщо вона може бути породжена однозначною КВ-граматикою $G = \langle V, T, P, S\rangle$. Мову $L \subset T^*$ називають суттєво неоднозначною, якщо L не може бути породжена жодною однозначною КВ-граматикою $G = \langle V, T, P, S\rangle$.

Так, КВ-мова $L = \{a^{k_1}c^{k_1}a^{k_2}c^{k_2}b^n c^n : k_1, k_2, n \geqslant 0\} \subset \{a,b,c\}^*$ однозначна, оскільки породжена однозначною КВ-граматикою із прикладу 4.1.

Зазначимо, що будь-яку (навіть однозначну) КВ-мову можна породити неоднозначною КВ-граматикою – для «псування» однозначності у граматиці $\langle V, T, P, S\rangle$ достатньо додати до множини P продукцію $S \to S$. Проте згідно з означенням 4.4 однозначність КВ-мови означає існування принаймні однієї однозначної КВ-граматики, що породжує цю мову. Саме тому важливою практичною задачею є пошук однозначної КВ-граматики, яка породжувала б задану КВ-мову. Зазвичай КВ-мова задана деякою (неоднозначною) КВ-граматикою, і тоді задача полягає у модифікації цієї граматики з метою позбутись її неоднозначності.

Приклад 4.4. Розглянемо КВ-граматику $G = \langle\{E, I\}, \{+, a, b\}, P, E\rangle$ із множиною продукцій $P = \{E \to E + E | I, I \to a|b\}$. Очевидно, що $L[G]$ містить «правильні» арифметичні вирази з бінарною операцією «$+$» та двома ідентифікаторами «a» і «b». Так, граматика породжує слова a, b, $a + a$, $a + b$ і т. д., але не породжує слово $+a$. Граматика G неоднозначна, оскільки будь-яке слово з двома та більше входженнями символа «$+$» можна отримати із джерела E більш ніж одним лівостороннім виведенням. Так, для слова $a+a+b$ існує два лівосторонніх виведення:

$$E \Rightarrow E+E \Rightarrow E+E+E \Rightarrow I+E+E \Rightarrow a+E+E \Rightarrow$$
$$\Rightarrow a+I+E \Rightarrow a+a+E \Rightarrow a+a+I \Rightarrow a+a+b;$$

$$E \Rightarrow E+E \Rightarrow I+E \Rightarrow a+E \Rightarrow a+E+E \Rightarrow$$
$$\Rightarrow a+I+E \Rightarrow a+a+E \Rightarrow a+a+I \Rightarrow a+a+b;$$

відповідні дерева розбору показано на рис. 4.5 та 4.6.

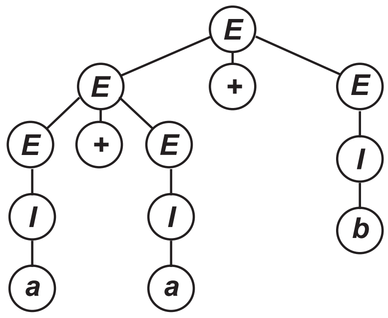

Рис. 4.5

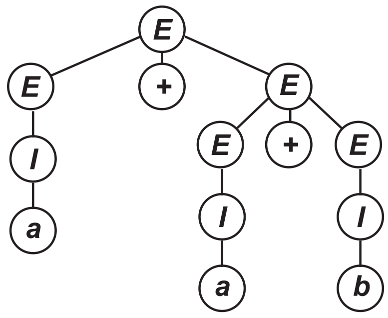

Рис. 4.6

Зазначимо, що така неоднозначність становить проблему при синтаксичному аналізі, навіть зважаючи на звичайну в математиці асоціативність операції «+» – принаймні через побічні ефекти, якщо a та b у реальній програмі означають виклик функцій. Проте в цьому разі неоднозначність легко усунути, конкретизувавши порядок виконання операції «+» – зазвичай зліва направо: модифікована граматика із множиною продукцій $P = \{E \to E+I|I, I \to a|b\}$ однозначна. Так, для розглянутого слова $a+a+b$ тепер існує єдине лівостороннє виведення

$$E \Rightarrow E+I \Rightarrow E+I+I \Rightarrow I+I+I \Rightarrow$$
$$\Rightarrow a+I+I \Rightarrow a+a+I \Rightarrow a+a+b,$$

яке відповідає порядку обчислення зліва направо. Відповідне дерево розбору зображено на рис. 4.7.

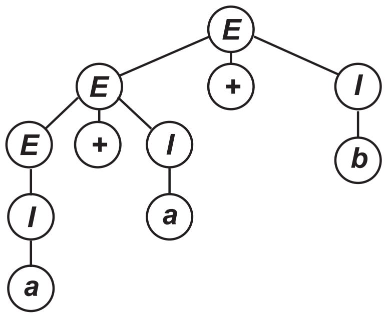

Рис. 4.7

Зауваження 4.1. Техніку із прикладу 4.4 можна узагальнити на випадок двох бінарних операцій з різними пріоритетами. Так, формальна граматика $G = \langle\{E, M, I\}, \{+, *, a, b\}, P, E\rangle$ із множиною продукцій $P = \{E \to E + M|M, M \to M * I|I, I \to a|b\}$ породжує арифметичні вирази з бінарними операціями «+» і «∗» та двома ідентифікаторами «a» і «b». Однозначність цієї граматики забезпечено введенням нетермінального символа M: аргументи операції «∗» не можуть містити «+», що гарантує пріоритетність «∗». Подальші узагальнення побудови однозначних граматик для арифметичних виразів, зокрема для виразів з використанням дужок, див., наприклад, у [4, 14].

Приклад 4.5. Розглянемо хрестоматійний приклад неоднозначної КВ-граматики, відомої як «стрибаюче **else**»:

$$G = \langle\{S\}, \{\mathbf{if}, \mathbf{then}, \mathbf{else}, a, b\}, P, S\rangle,$$
$$P = \{S \to \mathbf{if}\ b\ \mathbf{then}\ S \,|\, \mathbf{if}\ b\ \mathbf{then}\ S\ \mathbf{else}\ S \,|\, a\}.$$

Очевидно, що формальна КВ-граматика G породжує всі «правильні» синтаксичні конструкції, пов'язані з умовним оператором **if-then-else** («довга форма») або **if-then** («коротка форма»). Такі або аналогічні конструкції є в усіх (чи майже в усіх) сучасних алгоритмічних мовах програмування високого рівня. Граматика G неоднозначна, і ця неоднозначність добре відома у термінах синтаксичних конструкцій: при «вкладенні» умовних операторів, серед яких є принаймні один «довгий» і принаймні один «короткий», іноді неможливо встановити, до якого з операторів **if-then** належить секція **else**. Так, слово **if** b **then if** b **then** a **else** a можна породити із S двома лівосторонніми виведеннями:

$$S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ S\ \mathbf{else}\ S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ S\ \mathbf{else}\ S \Rightarrow$$
$$\Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ a\ \mathbf{else}\ S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ a\ \mathbf{else}\ a;$$
$$S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ S\ \mathbf{else}\ S \Rightarrow$$
$$\Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ a\ \mathbf{else}\ S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ a\ \mathbf{else}\ a;$$

відповідні дерева розбору зображено на рис. 4.8 та 4.9.

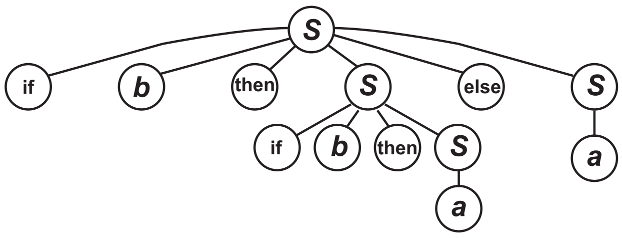

Рис. 4.8

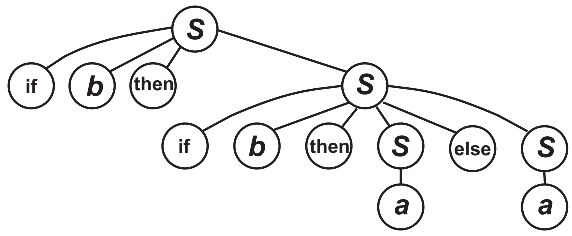

Рис. 4.9

Такої неоднозначності можна позбутися, заборонивши появу «короткої» форми оператора **if-then** перед **else** у «довгій» формі:

$$G = \langle\{S, S_0\}, \{\mathbf{if}, \mathbf{then}, \mathbf{else}, a, b\}, P, S\rangle,$$

$$P = \{S \to \mathbf{if}\ b\ \mathbf{then}\ S \,|\, \mathbf{if}\ b\ \mathbf{then}\ S_0\ \mathbf{else}\ S \,|\, a,$$
$$S_0 \to \mathbf{if}\ b\ \mathbf{then}\ S_0\ \mathbf{else}\ S_0 \,|\, a\}.$$

Так, слово **if** b **then if** b **then** a **else** a тепер можна отримати із S єдиним лівостороннім виведенням:

$$S \Rightarrow \mathbf{if}\ b\ \mathbf{then}\ S \Rightarrow$$
$$\Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ S_0\ \mathbf{else}\ S \Rightarrow$$
$$\Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ a\ \mathbf{else}\ S \Rightarrow$$
$$\Rightarrow \mathbf{if}\ b\ \mathbf{then}\ \mathbf{if}\ b\ \mathbf{then}\ a\ \mathbf{else}\ a;$$

відповідне дерево розбору показано на рис. 4.10.

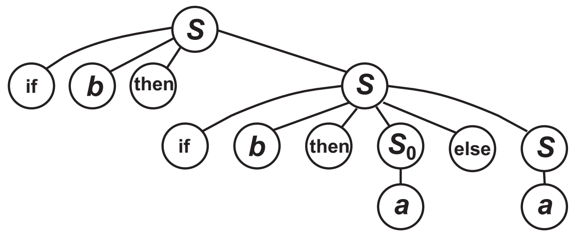

Рис. 4.10

Зазначимо, що обраний варіант позбування неоднозначності відповідає прийнятому у програмуванні правилу: «стрибаюче **else**» належить найближчому оператору **if**, якому під час синтаксичного розбору ще не приписано секції **else**.

Вправа 4.3. Позбутися неоднозначності у граматиці з прикладу 4.3.

Приклад 4.6. Розглянемо КВ-граматику $G = \langle\{S\}, T, P, S\rangle$ з термінальним алфавітом $T = \{a_1, b_1, a_2, b_2, \ldots, a_n, b_n\}$ (усі символи попарно різні) і множиною продукцій $P = \{S \to a_1Sb_1S|a_2Sb_2S|\ldots|a_nSb_nS|\varepsilon\}$. Нескладно довести, що ця граматика породжує «правильні» вирази з n парами дужок та є однозначною. Так, у випадку $n = 3$ три пари символів

a_1–b_1, a_2–b_2, a_3–b_3 можна розглядати як три пари дужок «(»–«)», «[»–«]», «{»–«}» відповідно; слово $a_3b_3a_1a_2b_2b_1$, яке в цьому випадку відповідає «дужковому виразу» $\{\}([\,])$, можна отримати із S єдиним лівостороннім виведенням:

$$S \Rightarrow a_3Sb_3S \Rightarrow a_3b_3S \Rightarrow a_3b_3a_1Sb_1S \Rightarrow a_3b_3a_1a_2Sb_2Sb_1S \Rightarrow \\ \Rightarrow a_3b_3a_1a_2b_2Sb_1S \Rightarrow a_3b_3a_1a_2b_2b_1S \Rightarrow a_3b_3a_1a_2b_2b_1. \tag{4.5}$$

Формальну мову, породжену цією граматикою, називають *мовою Діка*[1]. Детальніше про мову Діка див., наприклад, [10].

Вправа 4.4. Зобразити дерево розбору для виведення (4.5).

Вправа 4.5. Довести, що мова Діка є нерегулярною.

Вказівка. Скористатися лемою про розростання для регулярних мов (теорема 3.12).

Приклад 4.7. Розглянемо КВ-граматику $G = \langle\{S\}, T, P, S\rangle$ з термінальним алфавітом $T = \{a_0, a_1, \ldots, a_n\}$ (усі символи попарно різні) і множиною продукцій $P = \{S \to a_0|a_1S|a_2SS|\ldots a_nS^n\}$. Очевидно, що ця граматика є однозначною та породжує бездужкові вирази із префіксним записом операторів $a_0, a_1, \ldots, a_n$ арності від 0 до n. Так, слово $a_2a_2a_0a_1a_0a_0$ допускає єдине лівостороннє виведення із S:

$$S \Rightarrow a_2SS \Rightarrow a_2a_2SSS \Rightarrow a_2a_2a_0SS \Rightarrow a_2a_2a_0a_1SS \Rightarrow \\ \Rightarrow a_2a_2a_0a_1a_0S \Rightarrow a_2a_2a_0a_1a_0a_0. \tag{4.6}$$

Формальну мову, породжену цією граматикою, називають *мовою Лукасевича*[2].

Вправа 4.6. Зобразити дерево розбору для виведення (4.6).

Вправа 4.7. За яких $n \geqslant 0$ мова Лукасевича є регулярною?

Зауваження 4.2. Граматика з прикладу 4.7 описує загальну техніку для бездужкових виразів із префіксною формою запису. Так, арифметичний вираз $x * (y + z)$ у префіксній формі має вигляд $*x{+}yz$. Описана техніка була запропонована Яном Лукасевичем у 1924 р. і нині відома

[1] Дік Валтер, Валтер Франц Антон фон Дік (1856–1934) – німецький вчений, відомий, зокрема, як засновник комбінаторної теорії груп.

[2] Лукасевич (Лукашевич) Ян (1878–1956) – польський філософ і логік, запропонував аксіоматизовану тризначну логіку, нині відому як логіка Лукашевича.

як *префіксна* або *польська нотація*. У системах програмування з польською нотацією виразів (на базі мови типу LISP), враховуючи відсутність дужок, суттєво простіше проводити синтаксичний аналіз, що спрощує розробку трансляторів.

Наведемо приклад суттєво неоднозначної контекстно-вільної мови.

Приклад 4.8. Розглянемо формальну мову $L = \{a^m b^m c^n \colon m, n \geqslant 0\} \cup$ $\cup \{a^m b^n c^n : m, n \geqslant 0\}$ над термінальним алфавітом $\{a, b, c\}$. Очевидно, L є контекстно-вільною – породжувальну КВ-граматику легко побудувати, використовуючи техніку, описану в прикладі 1.13, п. 3–5. Однак мова L суттєво неоднозначна, тобто її неможливо породити жодною однозначною КВ-граматикою – можна довести (див. [4]), що слово $a^i b^i c^i$ для достатньо великого $i \geqslant 0$ можна отримати із джерела різними лівосторонніми виведеннями.

Вправа 4.8. Побудувати КВ-граматику, що породжує формальну мову із прикладу 4.8.

Зауваження 4.3. Інший, трохи складніший приклад суттєво неоднозначної мови див. у [14].

Зауваження 4.4. У багатьох простих випадках (див. приклади 4.3–4.5 та вправу 4.3) неоднозначність граматики можна легко усунути; у теорії синтаксичного аналізу відомі типові конструкції, які приводять до неоднозначності граматики. Проте у загальному випадку не існує алгоритму перевірки довільної КВ-граматики на неоднозначність, тобто проблема перевірки неоднозначності КВ-граматики є алгоритмічно нерозв'язною (детальніше див. [4, 9, 14]).

4.2. Нормальна форма Хомського

4.2.1. Видалення ε-продукцій у КВ-граматиках

Теорема 4.1. *Будь-яка КВ-мова L породжується такою КВ-граматикою $G = \langle V, T, P, S\rangle$, що для всіх $A \in V$ та $\alpha \in (V \cup T)^*$ виконуються умови:*

1) якщо $(A \to \varepsilon) \in P$, то $A = S$ (множина P містить не більш ніж одну ε-продукцію – $S \to \varepsilon$);

2) якщо $(A \to \alpha) \in P$, *то* $|\alpha|_S = 0$ *(джерело* S *не входить до правої частини жодної продукції).*

Доведення. Нехай КВ-мова L породжена деякою КВ-граматикою $G_0 = \langle V_0, T, P_0, S_0 \rangle$. Легко перевірити, що граматика G_0 еквівалентна граматиці $G_1 = \langle V, T, P_1, S \rangle$, де $V = V_0 \cup \{S\}$, $S \notin V_0$, $P_1 = P_0 \cup \{S \to S_0\}$. Зазначимо, що P_1 не містить жодної продукції, права частина якої містила б нове джерело S.

Далі введемо $P_2 = P_1 \cup \{A \to \alpha_1\alpha_2 \colon (A \to \alpha_1 B \alpha_2) \in P_1, (B \to \varepsilon) \in P_1\}$. Граматика $G_2 = \langle V, T, P_2, S \rangle$ еквівалентна граматикам G_0 та G_1, оскільки застосування «нової» продукції $A \to \alpha_1\alpha_2$ можна замінити послідовним застосуванням продукцій $A \to \alpha_1 B \alpha_2$ та $B \to \varepsilon$ із P_1.

Аналогічно, для кожної множини продукцій P_n $(n \geqslant 1)$ введемо множину $P_{n+1} = P_n \cup \{A \to \alpha_1\alpha_2 : (A \to \alpha_1 B \alpha_2), (B \to \varepsilon) \in P_n\}$; очевидно, $G_n = \langle V, T, P_n, S \rangle \sim G_0$ $(n \geqslant 1)$. За побудовою кожна множина P_n скінченна та $P_n \subset P_{n+1}$ $(n \geqslant 1)$. Оскільки права частина кожної «нової» продукції $A \to \alpha_1\alpha_2$ коротша за праву частину першої «породжувальної» продукції $A \to \alpha_1 B \alpha_2$, справджується оцінка

$$|P_n| \leqslant n_0 = |V| \left(\sum_{k=0}^{\max\{|\alpha| : (A \to \alpha) \in P_0\}} (|V| + |T|)^k \right).$$

Враховуючи до того ж, що $(P_n = P_{n+1}) \Rightarrow (P_{n+1} = P_{n+2})$, послідовність множин P_n $(n \geqslant 1)$ «стабілізується» не пізніше ніж за n_0 кроків: $P_n = P_{n_0}$ $(n \geqslant n_0)$.

Нарешті, можемо видалити із P_{n_0} всі ε-продукції, окрім $S \to \varepsilon$, оскільки будь-яка ε-продукція $B \to \varepsilon$ за умови $B \neq S$ може бути застосована лише після застосування деякої продукції вигляду $A \to \alpha_1 B \alpha_2$, але послідовне застосування продукцій $A \to \alpha_1 B \alpha_2$ та $B \to \varepsilon$ можна замінити застосуванням введеної продукції $A \to \alpha_1\alpha_2$. Отже, отримуємо граматику $G = \langle V, T, P, S \rangle$, де $P = P_{n_0} \setminus \{B \to \varepsilon \colon B \neq S\}$. Побудована граматика еквівалентна вихідній граматиці G_0, не містить ε-продукцій, окрім (можливо) продукції $S \to \varepsilon$, і праві частини продукцій множини P не містять джерела S. □

Наслідок. *Якщо мова* L *контекстно-вільна, то мова* $L \setminus \{\varepsilon\}$ *породжується контекстно-вільною граматикою без жодної* ε*-продукції.*

Зауваження 4.5. Нехай L – контекстно-вільна мова, породжена граматикою $\langle V, T, P, S\rangle$, що задовольняє умови теореми 4.1. Тоді мова $L\setminus\{\varepsilon\}$ породжується граматикою $\langle V, T, P\setminus\{S\to\varepsilon\}, S\rangle$. Зазначимо, що множина продукцій $P\setminus\{S\to\varepsilon\}$ справді не містить жодної ε-продукції.

Приклад 4.9. Використовуючи метод із доведення теореми 4.1, позбудемось ε-продукцій у КВ-граматиці $G_0 = \langle\{S_0, A, B\}, \{a, b\}, P_0, S_0\rangle$ із множиною продукцій $P_0 = \{S_0 \to AB, A \to a|\varepsilon, B \to b|\varepsilon\}$. Введемо нове джерело S та нову продукцію $S \to S_0$, отримуючи еквівалентну граматику $G_1 = \langle\{S, S_0, A, B\}, \{a, b\}, P_1, S\rangle$ із множиною продукцій $P_1 = \{S \to S_0, S_0 \to AB, A \to a|\varepsilon, B \to b|\varepsilon\}$.

Далі, враховуючи наявність продукцій $S_0 \to AB$, $A \to \varepsilon$ та $B \to \varepsilon$, введемо продукції $S_0 \to A$ та $S_0 \to B$, отримуючи множину продукцій $P_2 = \{S \to S_0, S_0 \to AB|A|B, A \to a|\varepsilon, B \to b|\varepsilon\}$. Аналогічно, враховуючи появу продукцій $S_0 \to A$ та $S_0 \to B$ та наявність продукцій $A \to \varepsilon$ та $B \to \varepsilon$, введемо продукцію $S_0 \to \varepsilon$, отримуючи множину продукцій $P_3 = \{S \to S_0, S_0 \to AB|A|B|\varepsilon, A \to a|\varepsilon, B \to b|\varepsilon\}$. Далі, враховуючи появу продукції $S_0 \to \varepsilon$ та наявність продукції $S \to S_0$, введемо продукцію $S \to \varepsilon$, отримуючи множину продукцій $P_4 = \{S \to S_0|\varepsilon, S_0 \to AB|A|B|\varepsilon, A \to a|\varepsilon, B \to b|\varepsilon\}$. Легко перевірити, що подальше застосування описаного механізму не приводить до появи нових продукцій.

Нарешті, можемо видалити із P_4 усі ε-продукції, окрім $S \to \varepsilon$, отримавши остаточно граматику

$$G = \langle\{S, S_0, A, B\}, \{a, b\}, P, S\rangle,$$
$$P = \{S \to S_0|\varepsilon, S_0 \to AB|A|B, A \to a, B \to b\}.$$

Легко пересвідчитися, що побудована граматика G еквівалентна заданій граматиці G_0 – справді, обидві породжують ту саму формальну мову $L[G] = \{\varepsilon, a, b, ab\}$. Підкреслимо, що множина продукцій P побудованої граматики містить лише одну ε-продукцію $S \to \varepsilon$, і джерело S не входить до правої частини жодної продукції – ці дві обставини гарантують, що граматика $\langle\{S, S_0, A, B\}, \{a, b\}, P\setminus\{S\to\varepsilon\}, S\rangle$ породжує мову $L[G]\setminus\{\varepsilon\}$.

Приклад 4.10. Використовуючи метод із доведення теореми 4.1, позбудемось ε-продукцій у КВ-граматиці $G_0 = \langle\{S_0\}, \{a, b\}, P_0, S_0\rangle$ із множиною продукцій $P_0 = \{S_0 \to aS_0b|\varepsilon\}$. Введемо нове джерело

S та нову продукцію $S \to S_0$, отримуючи еквівалентну граматику $G_1 = \langle\{S, S_0\}, \{a, b\}, P_1, S\rangle$, де $P_1 = \{S \to S_0, S_0 \to aS_0b|\varepsilon\}$.

Далі, враховуючи наявність продукцій $S \to S_0$, $S_0 \to aS_0b$ та $S_0 \to \varepsilon$, введемо продукції $S \to \varepsilon$ та $S_0 \to ab$, отримуючи множину продукцій $P_2 = \{S \to S_0|\varepsilon, S_0 \to aS_0b|ab|\varepsilon\}$. Очевидно, що подальше застосування механізму $P_{n+1} = P_n \cup \{A \to \alpha_1\alpha_2 : (A \to \alpha_1 B\alpha_2), (B \to \varepsilon) \in P_n\}$ не приводить до появи нових продукцій.

Нарешті, можемо видалити із P_2 усі ε-продукції, окрім $S \to \varepsilon$, отримавши остаточно граматику

$$G = \langle\{S, S_0\}, \{a, b\}, P, S\rangle,$$
$$P = \{S \to S_0|\varepsilon, S_0 \to aS_0b|ab\}.$$

Очевидно, що побудована граматика G еквівалентна заданій граматиці G_0 – справді, обидві породжують ту саму мову $L[G] = \{a^n b^n : n \geqslant 0\}$. Підкреслимо, що множина продукцій P побудованої граматики містить лише одну ε-продукцію $S \to \varepsilon$, і джерело S не входить до правої частини жодної продукції – ці дві обставини гарантують, що граматика $\langle\{S, S_0\}, \{a, b\}, P \setminus \{S \to \varepsilon\}, S\rangle$ породжує мову $L[G] \setminus \{\varepsilon\}$.

Вправа 4.9. Використовуючи метод із доведення теореми 4.1, позбутися ε-продукцій в КВ-граматиках із прикладів 4.1, 4.3, 4.6.

4.2.2. Нормальна форма Хомського

Означення 4.5. Контекстно-вільну граматику $G = \langle V, T, P, S\rangle$ називають граматикою у нормальній формі Хомського, якщо P містить лише продукції вигляду:

1) $S \to \varepsilon$;
2) $A \to a$, де $A \in V$, $a \in T$;
3) $A \to BC$, де $A \in V$, $B \in V \setminus \{S\}$, $C \in V \setminus \{S\}$.

Теорема 4.2. *Будь-яка КВ-мова породжується деякою КВ-граматикою у нормальній формі Хомського.*

Доведення. Нехай формальна мова $L \subset T^*$ породжується КВ-граматикою $G = \langle V, T, P, S\rangle$. З огляду на твердження теореми 4.1 можна вважати, що P не містить ε-продукцій, окрім, можливо, $S \to \varepsilon$, і джерело S не міститься у правій частині жодної продукції із P. Проведемо над граматикою G низку перетворень, зберігаючи еквівалентність вихідній граматиці G.

1. Для кожного термінального символа $a \in T$ введемо новий нетермінальний символ T_a, замінимо у правій частині кожної продукції всі входження a на T_a і додамо до множини P нові продукції $T_a \to a$ $(a \in T)$. Тут і далі збережемо для нового (розширеного) нетермінального алфавіту і нової множини продукцій позначення V та P відповідно. Отримуємо еквівалентну граматику, продукції якої мають вигляд $A \to \alpha$ $(A \in V, \alpha \in V^*)$ або $T_a \to a$ $(a \in T)$.

2. Кожну продукцію $(A \to B_1B_2 \ldots B_n) \in P$ $(n \geqslant 3)$ замінимо на $n-1$ продукцію $A \to B_1C_1$, $C_1 \to B_2C_2$, $\ldots$, $C_{n-3} \to B_{n-2}C_{n-2}$, $C_{n-2} \to B_{n-1}B_n$, де C_i $(1 \leqslant i \leqslant n-2)$ – нові нетермінальні символи. Отримуємо еквівалентну граматику з продукціями $A \to a$, $A \to BC$ та $A \to B$, де $A \in V$, $B \in V \setminus \{S\}$, $C \in V \setminus \{S\}$.

3. Для кожної пари продукцій $(A \to B) \in P$ (такі продукції називають *ланцюговими*) та $(B \to \alpha) \in P$ $(A \in V, B \in V \setminus \{S\}, \alpha \notin V)$ введемо продукцію $A \to \alpha$; процес повторюємо доти, доки можливо, тобто доки з'являються нові продукції. Підкреслимо, що завдяки умові $\alpha \notin V$ нові продукції не можуть мати вигляд $A \to B$, $A, B \in V$. Оскільки на цьому етапі можуть з'являтися нові продукції лише з існуючими правими частинами, тобто число $n_0 = |\{\alpha : \exists A \in V : (A \to \alpha) \in P\}|$ не зростає, загальна кількість продукцій на цьому етапі $|P| \leqslant n_0|V|$. Очевидно, нові продукції, додані до P на цьому етапі, не приводять до породження нових слів, а отже отримуємо еквівалентну граматику.

4. Видалимо із множини P всі продукції вигляду $A \to B$, $A, B \in V$. Очевидно, застосування ланцюгової продукції $A \to B$ може привести до породження нових слів над T лише після застосувань продукцій $B \to B_1$, $B_1 \to B_2$, $\ldots$, $B_{m-1} \to B_m$, $B_m \to \alpha$ $(\alpha \notin V)$. Проте, якщо наведені продукції містяться в P, тоді на етапі 3 до P за m кроків була додана продукція $B \to \alpha$, а на наступному кроці – продукція $A \to \alpha$. Отже, застосування ланцюгової продукції $A \to B$ з подальшим породженням слова $\alpha \notin V$ можна замінити застосуванням неланцюгової продукції $A \to \alpha$. Таким чином, отримана граматика еквівалентна заданій і за побудовою є граматикою у нормальній формі Хомського. □

Приклад 4.11. Використовуючи метод із доведення теореми 4.2, зведемо до нормальної форми Хомського КВ-граматику із прикладу 4.10.

Зазначимо, що, після перетворень, зроблених у прикладі 4.10, множина P не містить ε-продукцій, окрім $S \to \varepsilon$, і джерело S не входить до правої частини жодної продукції із P:

$$G = \langle \{S, S_0\}, \{a, b\}, P, S \rangle,$$
$$P = \{S \to S_0 | \varepsilon, S_0 \to aS_0b | ab\}.$$

Під час подальших перетворень збережемо для нових граматик і відповідних множин продукцій позначення G та P відповідно.

1. Введемо нетермінальні символи T_a і T_b, замінимо у правій частині кожної продукції всі входження a на T_a і b на T_b та додамо до множини P продукції $T_a \to a$ і $T_b \to b$. Отримуємо еквівалентну граматику без термінальних символів у правих частинах продукцій, окрім $T_a \to a$ і $T_b \to b$:

$$G = \langle \{S, S_0, T_a, T_b\}, \{a, b\}, P, S \rangle,$$
$$P = \{S \to S_0 | \varepsilon, S_0 \to T_aS_0T_b | T_aT_b, T_a \to a, T_b \to b\}.$$

2. Замінимо продукцію $S_0 \to T_aS_0T_b$ продукціями $S_0 \to T_aC$ і $C \to S_0T_b$, ввівши новий нетермінальний символ C. Отримуємо еквівалентну граматику з правими частинами довжиною не більше ніж 2:

$$G = \langle \{S, S_0, T_a, T_b, C\}, \{a, b\}, P, S \rangle,$$
$$P = \{S \to S_0 | \varepsilon, S_0 \to T_aC | T_aT_b, C \to S_0T_b, T_a \to a, T_b \to b\}.$$

3. Враховуючи наявність ланцюгової продукції $S \to S_0$ та продукцій $S_0 \to T_aC | T_aT_b$, введемо продукції $S \to T_aC | T_aT_b$. Очевидно, подальше застосування цього механізму не приводить до появи нових продукцій. Отримуємо еквівалентну граматику з множиною продукцій

$$P = \{S \to S_0 | T_aC | T_aT_b | \varepsilon, S_0 \to T_aC | T_aT_b, C \to S_0T_b, T_a \to a, T_b \to b\}.$$

4. Видалимо із множини P ланцюгову продукцію $S \to S_0$, отримуючи еквівалентну граматику у нормальній формі Хомського:

$$G = \langle \{S, S_0, T_a, T_b, C\}, \{a, b\}, P, S \rangle,$$
$$P = \{S \to T_aC | T_aT_b | \varepsilon, S_0 \to T_aC | T_aT_b, C \to S_0T_b, T_a \to a, T_b \to b\}.$$

Із щойно наведеного прикладу видно, що у практичних випадках, зводячи КВ-граматику до нормальної форми Хомського, деякі кроки можна пропускати. Так, у наведеному прикладі були пропущені кроки, пов'язані із введенням нового джерела та з видаленням ε-продукцій.

Наведемо дещо штучний приклад з ланцюговими продукціями, які замінюють нетермінальні символи «за циклом».

Приклад 4.12. Використовуючи метод із доведення теореми 4.2, зведемо до нормальної форми Хомського КВ-граматику

$$G = \langle \{S, A, B, C\}, \{a\}, \{S \to A, A \to B, B \to C, C \to A, B \to a\}, S \rangle.$$

Зазначимо, що множина P не містить ε-продукцій, і джерело S не входить до правої частини жодної продукції із P. Крім того, термінальні символи не містяться у правих частинах продукцій із P, окрім продукції $B \to a$, яка має вигляд, дозволений у нормальній формі Хомського. Таким чином, для зведення граматики до нормальної форми Хомського достатньо лише позбутися ланцюгових продукцій, що відповідає п. 3 і 4 наведеної процедури.

Враховуючи наявність ланцюгової продукції $A \to B$ і продукції $B \to a$, введемо продукцію $A \to a$ (підкреслимо, що процедура не передбачає введення нових ланцюгових продукцій). Далі, враховуючи наявність ланцюгових продукцій $C \to A$ і $S \to A$ та продукції $A \to a$, введемо продукції $C \to a$ і $S \to a$. Очевидно, подальше застосування цього механізму не приводить до появи нових продукцій. Нарешті, видаливши 4 ланцюгові продукції, отримуємо еквівалентну граматику у нормальній формі Хомського:

$$G = \langle \{S, A, B, C\}, \{a\}, \{B \to a, A \to a, C \to a, S \to a\}, S \rangle.$$

Очевидно, що із множини P можна видалити нетермінальні символи A, B і C разом із продукціями $A \to a$, $B \to a$, $C \to a$, залишивши лише нетермінальний символ S і продукцію $S \to a$. Поява «зайвих» продукцій і нетермінальних символів є досить типовим явищем при перетвореннях КВ-граматик – див., наприклад, [4, 14].

Вправа 4.10. Використовуючи метод із доведення теореми 4.2, звести до нормальної форми Хомського КВ-граматики із прикладів 4.1, 4.3, 4.4, 4.6, 4.7. 4.9.

Зауваження 4.6. Нормальну форму Хомського часто використовують у синтаксичному аналізі. Наприклад, алгоритм Кока–Янгера–Касамі, який перевіряє, чи може слово бути породжене контекстно-вільною граматикою, вимагає, щоб граматика була записана саме у нормальній формі Хомського (див., наприклад, [4, 14, 27]).

Зауваження 4.7. Зазначимо, що, окрім нормальної форми Хомського, є інші зображення контекстно-вільних граматик, зокрема, нормальна форма Грейбах[1]. КВ-граматику $G = \langle V, T, P, S\rangle$ називають граматикою у нормальній формі Грейбах, якщо множина продукцій P містить лише продукції виду $S \to \varepsilon$ та $A \to a\alpha$, де $A \in V$, $a \in T$, $\alpha \in V^*$ [4, 14]. Кожна контекстно-вільна мова L породжується граматикою у нормальній формі Грейбах, доведення (див., наприклад, [4]) конструктивне, тобто надає конкретний метод побудови шуканої граматики. Зауважимо, що іноді дозволяють продукцію $A \to \varepsilon$ і вимагають $|\alpha| \leqslant 2$ (наприклад, [9]).

4.3. Автомати з магазинною пам'яттю

Автомат з магазинною пам'яттю (МП-автомат, магазинний автомат, стековий автомат) можна неформально подати як абстрактний пристрій, який містить:

1) необмежену в обидва боки стрічку, кожна комірка якої може містити один символ *вхідного алфавіту* T або не містити жодного (містить порожній символ $\Lambda \notin T$);

2) курсор, який в кожний момент часу вказує на певну комірку і може пересуватися вздовж стрічки зліва направо; комірку називають *поточною*, символ у поточній комірці також називають *поточним*;

3) *магазин* (*стек*), який містить нескінченну кількість комірок, кожна з яких може містити один символ *алфавіту магазинної пам'яті* (*стекового алфавіту*) Γ або не містити жодного.

Заповнення стеку (запис) та зчитування зі стеку (видалення) організовані за принципом «останнім зайшов, першим вийшов»[2]. Єдиний символ стеку, до якого є доступ, називають вершиною стеку. Вважатимемо,

[1] Грейбах Шейла Адель (народ. у 1939 р.) – видатний американський вчений, отримала вагомі результати у галузі формальних мов, граматик та МП-автоматів; довела нерозв'язність деяких алгоритмічних проблем, пов'язаних з КВ-мовами.

[2] Акронім LIFO – last in, first out (англ.)

що стек «зростає» справа наліво. Таким чином, вершиною є перший зліва символ у стеку.

МП-автомат працює покроково, за дискретними моментами часу $t = 0, 1, 2, \ldots$. На кожному кроці він перебуває в одному зі своїх *станів*; стан МП-автомата на конкретному кроці називають *поточним.* На початку роботи, тобто у момент $t = 0$, на стрічці записане вхідне слово скінченної довжини w. Якщо слово w непорожнє, курсор вказує на перший зліва символ вхідного слова. МП-автомат перебуває в одному із *початкових станів* $q_0 \in I$, стек порожній. На кожному кроці МП-автомат за поточним станом $q_1 \in Q$, поточним символом або порожнім словом $\xi \in (T \cup \{\varepsilon\})$ та вершиною стеку $\alpha \in (\Gamma \cup \{\varepsilon\})$ шукає перехід $((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta$ ($q_2 \in Q$, $\beta \in \Gamma^*$) із множини *переходів* $\Delta \subset ((Q \times (T \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})) \times (Q \times \Gamma^*))$ і виконує такі дії:

1) видаляє поточний символ (записує у поточну комірку порожній символ Λ) і переміщує курсор на одну комірку праворуч; якщо $\xi = \varepsilon$, видалення поточного символа та переміщення курсора не відбувається – такий перехід називають ε-переходом за аналогією до скінченних автоматів з ε-переходами (див. означення 3.6);

2) зчитує та одночасно видаляє з вершини стеку слово α, після чого записує у стек слово β (перший символ слова β стає вершиною стеку); зокрема, у разі $\alpha = \varepsilon$ зчитування немає, у разі $\beta = \varepsilon$ запису немає, у разі $\alpha = \beta = \varepsilon$ жодних дій зі стеком немає;

3) змінює поточний стан на стан $q_2 \in Q$ (можливий випадок $q_2 = q_1$).

Таким чином, МП-автомат зчитує зі стеку не більше одного символа, проте записує у стек слово довільної довжини. Якщо дії МП-автомата визначені однозначно, і до того ж МП-автомат має єдиний початковий стан $q_0 \in I$, МП-автомат називатимемо *детермінованим.* МП-автомат без обмежень щодо детермінованості називатимемо *недетермінованим.*[1]

Якщо вхідне слово прочитане повністю (поточним символом є Λ), МП-автомат може виконувати лише ε-переходи.

Надамо формальний опис автомата з магазинною пам'яттю.

Означення 4.6. Автоматом з магазинною пам'яттю називають впорядкований набір $\langle Q, T, \Gamma, \Delta, I, F\rangle$, де Q, T, Γ – скінченні непорожні множини, $\Delta \subset ((Q \times (T \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})) \times (Q \times \Gamma^*))$, $I \subset Q$, $F \subset Q$. Множину Q називають множиною станів, T – вхідним алфавітом, Γ – алфавітом

[1] Формальне визначення детермінованого МП-автомата див., наприклад, у [4, 9, 14].

магазинної пам'яті (стековим алфавітом), Δ – відношенням (множиною) переходів, I – множиною початкових станів, F – множиною допускаючих станів.

Конфігурацією МП-автомата $\langle Q, T, \Gamma, \Delta, I, F\rangle$ називають довільний набір $(q, w, \gamma) \in (Q \times T^* \times \Gamma^*)$.

Зауваження 4.8. Порожній символ Λ у неформальному описі МП-автомата використовують лише для описання ситуації, коли вхідне слово прочитане повністю, і курсор вказує саме на Λ. Для формального описання МП-автомата, що не передбачає введення стрічки та курсора, порожній символ не потрібний.

На множині конфігурацій МП-автомата $M = \langle Q, T, \Gamma, \Delta, I, F\rangle$ визначимо бінарне відношення такту роботи «$\vdash_M$»:

$$((q_1, w_1, \gamma_1) \vdash_M (q_2, w_2, \gamma_2)) \Leftrightarrow \begin{cases} w_1 = \xi w_2,\ \xi \in (T \cup \{\varepsilon\}); \\ \gamma_1 = \alpha\gamma, \gamma_2 = \beta\gamma,\ \alpha \in (\Gamma \cup \{\varepsilon\}), \beta \in \Gamma^*; \\ ((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta, \end{cases}$$

де $q_1, q_2 \in Q$, $w_1, w_2 \in T^*$, $\gamma \in \Gamma^*$. Конфігурацію $c \in (Q \times T^* \times \Gamma^*)$ назвемо тупиковою, якщо не існує жодної конфігурації $\tilde{c}$, такої, що $c \vdash_M \tilde{c}$.

Нагадаємо, що «$\vdash_M^*$» – транзитивно-рефлексивне замикання відношення «$\vdash_M$» (див. означення 1.4); якщо із контексту зрозуміло, що йдеться саме про МП-автомат M, замість «$\vdash_M$» та «$\vdash_M^*$» писатимемо відповідно «$\vdash$» та «$\vdash^*$».

Кажуть, що МП-автомат $M = \langle Q, T, \Gamma, \Delta, I, F\rangle$ допускає (сприймає) слово $w \in T^*$, якщо $(q_0, w, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$ для деяких $q_0 \in I$ та $q \in F$. Це означає, що МП-автомат має повністю зчитати слово w, опинившись в одному з допускаючих станів при порожньому стеку. Множину слів $L[M]$, які допускає МП-автомат M, називають формальною мовою, яку допускає (сприймає) МП-автомат M. Буде доведено (підрозд. 4.4), що клас мов, які допускають МП-автомати, збігається з класом КВ-мов.

МП-автомати M_1 та M_2 називають еквівалентними, якщо M_1 та M_2 задані над спільним вхідним алфавітом та допускають ту саму мову: $(M_1 \sim M_2) \Leftrightarrow (L[M_1] = L[M_2])$.

Зауваження 4.9. Іноді в літературі (наприклад, [4, 11, 14]) під МП-автоматом розуміють набір $\langle Q, T, \Gamma, \Delta, I, Z_0, F\rangle$, де Z_0 – початковий символ або маркер дна стеку, який перебуває у стеку (магазині) на початку роботи, тобто у момент часу $t = 0$. Такий підхід еквівалентний розглянутому та не змінює множину слів $L[M]$, які допускає МП-автомат.

Зауваження 4.10. Розглядають МП-автомати, які мають вихідний потік (перетворювачі з магазинною пам'яттю), детальніше див. [4, 23].

На МП-автомати поширюють техніку задання за допомогою графів, описану в підрозд. 3.1.3: для кожного переходу $((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta$ до відповідного графа включають ребро з міткою $\xi, \alpha/\beta$, що веде від вершини з міткою q_1 до вершини з міткою q_2.

Приклад 4.13. Над вхідним алфавітом $T = \{a, b\}$ розглянемо МП-автомат $\langle Q, T, \Gamma, \Delta, I, F \rangle$ із множиною станів $Q = \{q_0, q_1\}$, стековим алфавітом $\Gamma = \{A\}$, множиною початкових станів $I = \{q_0\}$, множиною допускаючих станів $F = \{q_0, q_1\}$ та множиною переходів

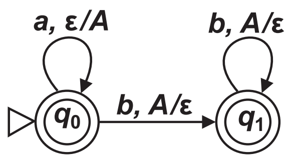

Рис. 4.11

$$\Delta = \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, b, A), (q_1, \varepsilon)), \\ ((q_1, b, A), (q_1, \varepsilon))\}.$$

Граф МП-автомата зображено на рис. 4.11.

Оскільки ε-переходів немає, МП-автомат обов'язково має зчитувати один символ на кожному кроці. Перебуваючи в початковому стані q_0, МП-автомат може зчитати як символ a, виконавши перехід $((q_0, a, \varepsilon), (q_0, A))$, залишаючись у стані q_0 і записуючи у стек A, так і символ b, виконавши перехід $((q_0, b, A), (q_1, \varepsilon))$, переходячи у стан q_1 і видаляючи символ A з вершини стеку. Перебуваючи у стані q_1, МП-автомат може зчитати лише символ b, виконавши перехід $((q_1, b, A), (q_1, \varepsilon))$, залишаючись у стані q_1 і видаляючи символ A з вершини стеку. Отже, МП-автомат, перебуваючи у стані q_0, зчитує послідовність символів a, на кожному кроці записуючи у стек A; з появою першого символа b, МП-автомат переходить і надалі залишається у стані q_1, зчитує послідовність символів b, на кожному кроці видаляючи A з вершини стеку.

Таким чином, МП-автомат може зчитувати вхідні слова вигляду $a^n b^m$ $(n \geqslant m \geqslant 0)$, залишивши у стеку після зчитування $n - m$ символів A, тобто, враховуючи, що $F = Q = \{q_0, q_1\}$, МП-автомат допускає КВ-мову $\{a^n b^n : n \geqslant 0\}$; зазначимо, що нерегулярність цієї мови доведена у прикладах 3.28 та 3.47.

Цей МП-автомат є детермінованим, оскільки має один початковий стан, і дії МП-автомата на кожному кроці визначені однозначно.

Продемонструємо роботу цього МП-автомата у термінах відношення «$\vdash$» (нагадаємо, що вершиною стеку є перший зліва символ):

$$(q_0, a^2b^2, \varepsilon) \vdash (q_0, ab^2, A) \vdash (q_0, b^2, A^2) \vdash (q_1, b, A) \vdash (q_1, \varepsilon, \varepsilon);$$
$$(q_0, a^2b^3, \varepsilon) \vdash (q_0, ab^3, A) \vdash (q_0, b^3, A^2) \vdash (q_1, b^2, A) \vdash (q_1, b, \varepsilon);$$
$$(q_0, a^2b, \varepsilon) \vdash (q_0, ab, A) \vdash (q_0, b, A^2) \vdash (q_1, \varepsilon, A);$$
$$(q_0, aba, \varepsilon) \vdash (q_0, ba, A) \vdash (q_1, a, \varepsilon).$$

Слово a^2b^2 МП-автомат зчитає повністю у допускаючому стані q_1 з порожнім стеком, тобто МП-автомат допускає слово a^2b^2. Слово a^2b^3 МП-автомат не зчитує повністю, оскільки для поточного символа b немає відповідного переходу з поточним станом q_1, який не зчитує зі стеку жодного символа; слово a^2b МП-автомат зчитує повністю та потрапляє у тупикову конфігурацію у допускаючому стані q_1, але стек непорожній – містить символ A; слово aba не буде зчитане повністю, оскільки для поточного символа a немає відповідного переходу з поточним станом q_1. Отже, цей МП-автомат не допускає слова a^2b^3, a^2b та aba.

Окремо зазначимо, що МП-автомат може завершити роботу (потрапити в одну з тупикових конфігурацій) у стані q_0, якщо вхідне слово починається з b або є порожнім. Так, МП-автомат допускає слово ε та не допускає b^2 (як і будь-яке слово з першим символом b) – в обох випадках МП-автомат залишається в початковому стані q_0, не зробивши жодного такту.

Зауважимо, що можна побудувати еквівалентний МП-автомат, увівши додатковий ε-перехід $((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon))$ та залишивши лише один допускаючий стан q_1:

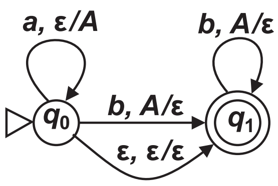

Рис. 4.12

$$\langle \{q_0, q_1\}, \{a, b\}, \{A\}, \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, b, A), (q_1, \varepsilon)), ((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon)), ((q_1, b, A), (q_1, \varepsilon))\}, \{q_0\}, \{q_1\}\rangle,$$

граф цього МП-автомата зображено на рис. 4.12. Зазначимо, що цей МП-автомат недетермінований. Так, вхідним словом a^2b^2 із q_0 можна потрапити у три різні тупикові конфігурації:

$$(q_0, a^2b^2, \varepsilon) \vdash (q_1, a^2b^2, \varepsilon);$$
$$(q_0, a^2b^2, \varepsilon) \vdash (q_0, ab^2, A) \vdash (q_1, ab^2, A);$$

$$(q_0, a^2b^2, \varepsilon) \vdash (q_0, ab^2, A) \vdash (q_0, b^2, A^2) \vdash (q_1, b, A) \vdash (q_1, \varepsilon, \varepsilon);$$
$$(q_0, a^2b^2, \varepsilon) \vdash (q_0, ab^2, A) \vdash (q_0, b^2, A^2) \vdash (q_1, b^2, A^2) \vdash (q_1, b, A) \vdash (q_1, \varepsilon, \varepsilon).$$

Отже, МП-автомат допускає слово a^2b^2, оскільки $(q_0, a^2b^2, \varepsilon) \vdash^{*} (q_1, \varepsilon, \varepsilon)$, де $q_0 \in I$, $q_1 \in F$.

МП-автомат, граф якого зображено на рис. 4.12, можна дещо спростити, видаливши перехід $((q_0, b, A), (q_1, \varepsilon))$, дію якого можна замінити двома послідовними переходами $((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon))$ і $((q_1, b, A), (q_1, \varepsilon))$ (рис. 4.13). Очевидно, це спрощення залишає МП-автомат недетермінованим.

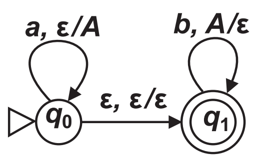

Рис. 4.13

Приклад 4.14. Над вхідним алфавітом $T = \{a, b\}$ розглянемо МП-автомат $\langle \{q_0, q_1\}, T, \{A\}, \Delta, \{q_0\}, \{q_1\} \rangle$ із множиною переходів

$$\Delta = \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon)),$$
$$((q_1, b, A), (q_1, \varepsilon)), ((q_1, b, \varepsilon), (q_1, \varepsilon))\},$$

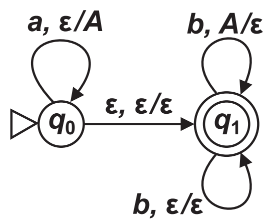

Рис. 4.14

граф МП-автомата зображено на рис. 4.14.

Цей МП-автомат недетермінований, оскільки у конфігурації з початковим станом q_0 та поточним символом a, незалежно від вершини стеку, може бути виконаний як ε-перехід $((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon))$, так і перехід $((q_0, a, \varepsilon), (q_0, A))$. Аналогічно, у конфігурації з поточним станом q_1, поточним символом b та вершиною стеку A може бути виконаний як перехід $((q_1, b, \varepsilon), (q_1, \varepsilon))$, так і перехід $((q_1, b, A), (q_1, \varepsilon))$.

Поки МП-автомат перебуває в q_0, до стеку буде записано стільки символів A, скільки буде зчитано символів a; з переходом в q_1 при зчитуванні символа b з вершини стеку може бути видалений символ A, або стек може не змінитись. Отже, для спорожнення стеку МП-автомат має зчитати символів b не менше ніж a, тобто допускає мову $L = \{a^m b^n : n \geqslant m \geqslant 0\}$. Покажемо можливі дії МП-автомата для вхідного слова ab^2:

$$(q_0, ab^2, \varepsilon) \vdash (q_1, ab^2, \varepsilon);$$
$$(q_0, ab^2, \varepsilon) \vdash (q_0, b^2, A) \vdash (q_1, b^2, A) \vdash (q_1, b, \varepsilon) \vdash (q_1, \varepsilon, \varepsilon);$$
$$(q_0, ab^2, \varepsilon) \vdash (q_0, b^2, A) \vdash (q_1, b^2, A) \vdash (q_1, b, A) \vdash (q_1, \varepsilon, \varepsilon);$$
$$(q_0, ab^2, \varepsilon) \vdash (q_0, b^2, A) \vdash (q_1, b^2, A) \vdash (q_1, b, A) \vdash (q_1, \varepsilon, A).$$

Маємо три різні тупикові конфігурації (q_1, ab^2, ε), $(q_1, \varepsilon, \varepsilon)$ та (q_1, ε, A). Отже, МП-автомат допускає слово ab^2, оскільки $(q_0, ab^2, \varepsilon) \overset{*}{\vdash} (q_1, \varepsilon, \varepsilon)$, де $q_0 \in I$, $q_1 \in F$.

Аналогічно прикладу 4.13, можна замінити ε-перехід $((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon))$ на два переходи $((q_0, b, A), (q_1, \varepsilon))$ і $((q_0, b, \varepsilon), (q_1, \varepsilon))$ та зробити стан q_0 допускаючим, однак отриманий МП-автомат теж є недетермінованим.

Вправа 4.11. Побудувати для мови з прикладу 4.14 породжувальну КВ-граматику та перевірити її на однозначність.

Приклад 4.15. Побудуємо МП-автомат, який допускає формальну мову $L = \{ww^R \colon w \in T^*\}$ над алфавітом $T = \{a, b\}$. Як і в прикладах 4.13 та 4.14, МП-автомат має розділити вхідне слово на дві частини, занести у стек певну інформацію щодо першої частини та перевірити відповідність другої частини вхідного слова даним у стеку.

На відміну від МП-автоматів із прикладів 4.13 та 4.14, у цьому випадку необхідно занести у стек не тільки кількість символів a чи b, а й порядок їх розміщення – по суті, необхідно запам'ятати у стеку слово w. Таким чином, під час зчитування символів a та b у першій частині вхідного слова необхідно записувати у стек різні символи, тобто стековий алфавіт має містити принаймні два символи. Крім того, на відміну від формальних мов із прикладів 4.13 та 4.14, у слові вигляду ww^R $(w \in T^*)$ під час зчитування зліва направо неможливо визначити, де закінчується перша частина вхідного слова, у цьому випадку – слово w, тобто половина вхідного слова ww^R. Таким чином, перехід у стан q_1 для зчитування другої половини вхідного слова необхідно здійснювати, передбачивши альтернативу – залишитися в q_0 і продовжити зчитування першої половини, – що несумісно з детермінованістю. За аналогією до МП-автоматів із прикладів 4.13 та 4.14, перехід із q_0 в q_1 найпростіше здійснити ε-переходом, отримуючи недетермінований МП-автомат.

Отже, вводимо МП-автомат $\langle \{q_0, q_1\}, T, \{A, B\}, \Delta, \{q_0\}, \{q_1\} \rangle$ із множиною переходів

$$\Delta = \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, b, \varepsilon), (q_0, B)), ((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon)),$$
$$((q_1, a, A), (q_1, \varepsilon)), ((q_1, b, B), (q_1, \varepsilon))\},$$

граф МП-автомата зображено на рис. 4.15.

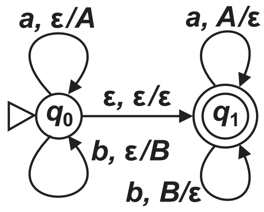

Рис. 4.15

Перебуваючи у початковому стані q_0, МП-автомат має зчитувати першу половину вхідного слова, на кожному кроці записуючи у стек символ A, зчитуючи поточний символ a (перехід $((q_0, a, \varepsilon), (q_0, A))$), або символ B, зчитуючи поточний символ b (перехід $((q_0, b, \varepsilon), (q_0, B))$). Перебуваючи в q_0, МП-автомат на будь-якому кроці, незалежно від поточного символу і вершини стеку, може виконати ε-перехід $((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon))$, переходячи у стан q_1. Перебуваючи у стані q_1, МП-автомат має зчитувати другу половину вхідного слова, на кожному кроці видаляючи зі стеку символ A, зчитуючи поточний символ a (перехід $((q_1, a, A), (q_1, \varepsilon))$), або B, зчитуючи поточний символ b (перехід $((q_1, b, \varepsilon), (q_1, B))$). Таким чином, МП-автомат має зчитати у стані q_0 у точності першу половину вхідного слова ww^R, тобто слово w, записавши у стек $|w|$ символів, а у стані q_1 – у точності другу половину вхідного слова ww^R, тобто слово w^R, видаливши зі стеку $|w^R| = |w|$ символів; лише в цьому випадку МП-автомат повністю зчитає слово ww^R, спорожнивши стек на останньому кроці.

Покажемо можливі дії МП-автомата для вхідного слова $abba \in L$:

$$(q_0, abba, \varepsilon) \vdash (q_1, abba, \varepsilon); \quad (q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_1, bba, A);$$
$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, BA) \vdash (q_1, ba, BA) \vdash (q_1, a, A) \vdash (q_1, \varepsilon, \varepsilon);$$
$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, BA) \vdash (q_0, a, BBA) \vdash (q_1, a, BBA);$$
$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, BA) \vdash (q_0, a, BBA) \vdash$$
$$\vdash (q_0, \varepsilon, ABBA) \vdash (q_1, \varepsilon, ABBA).$$

Маємо п'ять різних тупикових конфігурацій. МП-автомат допускає слово $abba$, оскільки $(q_0, abba, \varepsilon) \overset{*}{\vdash} (q_1, \varepsilon, \varepsilon)$, де $q_0 \in I$, $q_1 \in F$. Підкреслимо: якщо ε-перехід виконано до того, як зчитана перша половина вхідного слова ww^R (занадто рано), вхідне слово ww^R не буде зчитане повністю; якщо ε-перехід виконано після того, як був зчитаний хоча б один символ другої половини вхідного слова ww^R (занадто пізно), у тупиковій конфігурації стек виявиться непорожнім.

Покажемо роботу МП-автомата для вхідного слова $ab^2 \notin L$:

$$(q_0, ab^2, \varepsilon) \vdash (q_1, ab^2, \varepsilon);$$
$$(q_0, ab^2, \varepsilon) \vdash (q_0, b^2, A) \vdash (q_1, b^2, A);$$

$$(q_0, ab^2, \varepsilon) \vdash (q_0, b^2, A) \vdash (q_0, b, BA) \vdash (q_1, b, BA) \vdash (q_1, \varepsilon, A);$$
$$(q_0, ab^2, \varepsilon) \vdash (q_0, b^2, A) \vdash (q_0, b, BA) \vdash (q_0, \varepsilon, BBA) \vdash (q_1, \varepsilon, BBA).$$

Маємо чотири різні тупикові конфігурації, серед яких немає $(q_1, \varepsilon, \varepsilon)$; отже, МП-автомат не допускає слово ab^2.

Зауваження 4.11. На відміну від детермінованих скінченних автоматів, які допускають той самий клас мов, що й недетерміновані (див. теорему 3.1), детерміновані МП-автомати допускають вужчий клас мов, ніж недетерміновані. Наприклад, мову $L = \{ww^R : w \in T^*\}$ у разі $|T| \geqslant 2$ (див. приклад 4.15) справді не може допустити жодний детермінований МП-автомат – необхідні теоретичні відомості для доведення недетермінованості цієї мови див., наприклад, у [13]; власне доведення недетермінованості мови $L = \{ww^R : w \in T^*\}$ та деяких інших мов автори залишили як вправу.

Вправа 4.12. Побудувати детермінований МП-автомат, який допускає мову $L = \{wcw^R : w \in \{a, b\}^*\}$ над алфавітом $T = \{a, b, c\}$.

Вправа 4.13. Побудувати для мов із прикладу 4.15 та вправи 4.12 породжувальні КВ-граматики та перевірити їх на однозначність.

Вправа 4.14. Побудувати МП-автомат, який допускає мову Діка з n парами дужок (див. приклад 4.6), та продемонструвати роботу цього МП-автомата для вхідних слів $a_1a_1b_1b_1a_1b_1a_1b_1$ та $a_3b_3a_1a_2b_2b_1$.

Вказівка. Для $n = 1$ цей МП-автомат має вигляд

$$\langle \{q_0\}, \{a, b\}, \{C\}, \{((q_0, a, \varepsilon), (q_0, C)), ((q_0, b, C), (q_0, \varepsilon))\}, \{q_0\}, \{q_0\} \rangle.$$

Вправа 4.15. Побудувати МП-автомат, який допускає мову Лукасевича (див. приклад 4.7).

Приклад 4.16. Побудуємо МП-автомат, який допускає формальну мову $L = \{w \in T^* : |w|_a = |w|_b\}$ над алфавітом $T = \{a, b\}$, породжувальна граматика для якої розглянута у прикладі 4.3 (див. також вправу 4.1). МП-автомат має контролювати співвідношення кількості зчитаних символів a та кількості зчитаних b – вхідне слово має бути допущеним тоді й тільки тоді, коли загальна кількість зчитаних символів a та кількість зчитаних b збігаються. Очевидно, що достатньо контролювати різницю між кількістю зчитаних символів a та b, і це слід робити за допомогою стеку – адже ця різниця може бути як завгодно великою, а кількість станів МП-автомата скінченна і фіксована.

Очевидно, різниця між кількістю зчитаних символів a та b може бути як додатна, так і від'ємна. Для зберігання інформації про знак достатньо передбачити два різні символи стекового алфавіту: додатну різницю зберігати у стеку як кількість символів A, від'ємну – як кількість B. На початку роботи у момент $t = 0$ різниця між кількістю зчитаних символів a та b дорівнює нулю. Якщо на поточному кроці зчитано символ a, до попереднього значення різниці додаємо одиницю, якщо символ b – віднімаємо одиницю. Якщо на вершині стеку міститься A, додавання одиниці (зчитано a) означає запис A у стек, віднімання одиниці (зчитано b) означає видалення A зі стеку. Якщо на вершині стеку міститься B, додавання одиниці (зчитано a) означає видалення B зі стеку, віднімання одиниці (зчитано b) означає запис B у стек. Якщо стек на поточному кроці порожній, додавання та віднімання одиниці можна реалізувати тільки записом у стек відповідно A або B. Отже, мову L допустить, наприклад, такий МП-автомат M з одним станом:

$$M = \langle \{q_0\}, T, \{A, B\}, \Delta, \{q_0\}, \{q_0\} \rangle,$$
$$\Delta = \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, a, B), (q_0, \varepsilon)),$$
$$((q_0, b, A), (q_0, \varepsilon)), ((q_0, b, \varepsilon), (q_0, B))\}.$$

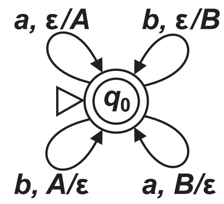

Рис. 4.16

Граф МП-автомата M зображено на рис. 4.16.

МП-автомат M недетермінований. Справді, для вхідного слова $abba$ можливі такі дії:

$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, \varepsilon) \vdash (q_0, a, B) \vdash (q_0, \varepsilon, \varepsilon);$$
$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, \varepsilon) \vdash (q_0, a, B) \vdash (q_0, \varepsilon, AB);$$
$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, BA) \vdash (q_0, a, BBA) \vdash (q_0, \varepsilon, BA);$$
$$(q_0, abba, \varepsilon) \vdash (q_0, bba, A) \vdash (q_0, ba, BA) \vdash (q_0, a, BBA) \vdash (q_0, \varepsilon, ABBA).$$

Маємо чотири різні тупикові конфігурації, серед яких є $(q_0, \varepsilon, \varepsilon)$, тобто МП-автомат справді допускає слово $abba$.

Вправа 4.16. Використовуючи підхід, викладений у прикладі 4.16, побудувати МП-автомати, які допускають формальні мови над алфавітом $T = \{a, b\}$:

1) $\{w \in T^*\colon |w|_a > |w|_b\}$;
2) $\{w \in T^*\colon |w|_a < |w|_b\}$;
3) $\{w \in T^*\colon |w|_a \geqslant |w|_b\}$;
4) $\{w \in T^*\colon |w|_a \leqslant |w|_b\}$;
5) $\{w \in T^*\colon |w|_a \neq |w|_b\}$;
6) $\{w \in T^*\colon 2|w|_a = 3|w|_b\}$.

Вправа 4.17. Побудувати для мов із вправи 4.16 породжувальні КВ-граматики та перевірити їх на однозначність.

Вправа 4.18. Побудувати детермінований МП-автомат, який допускає формальну мову $\{w \in T^*\colon |w|_a = |w|_b\}$ над алфавітом $T = \{a, b\}$.

Вправа 4.19. Побудувати МП-автомат, який допускає формальну мову із прикладу 3.48, та перевірити його на детермінованість.

4.4. Характеризація класу контекстно-вільних мов через МП-автомати

Доведемо, що клас мов, які допускають автомати з магазинною пам'яттю, збігається з класом контекстно-вільних мов.

4.4.1. Побудова МП-автомата за КВ-граматикою

Теорема 4.3. *Нехай L – контекстно-вільна мова. Тоді існує МП-автомат M, який допускає мову L.*

Доведення. Нехай мову L породжує КВ-граматика $G = \langle V, T, P, S\rangle$. З огляду на теорему 4.2 можна вважати, що G є граматикою у нормальній формі Хомського, тобто містить лише продукції вигляду $S \to \varepsilon$, $A \to a$ та $A \to BC$, де $A \in V$, $a \in T$, $B \in V \setminus \{S\}$, $C \in V \setminus \{S\}$ (див. означення 4.5). Визначимо шуканий (взагалі кажучи, недетермінований) МП-автомат $M = \langle Q, T, \Gamma, \Delta, I, F\rangle$:

1) множина станів $Q = \{q_0, q_1\}$;
2) алфавіт магазинної пам'яті $\Gamma = V$;
3) множина переходів

$$\Delta = \{((q_0, \varepsilon, \varepsilon), (q_1, S))\} \cup \{((q_1, \varepsilon, A), (q_1, BC))\colon (A \to BC) \in P\} \cup$$
$$\cup \{((q_1, a, A), (q_1, \varepsilon))\colon (A \to a) \in P\} \cup \{((q_1, \varepsilon, S), (q_1, \varepsilon))\colon (S \to \varepsilon) \in P\};$$

4) множина початкових станів $I = \{q_0\}$;
5) множина допускаючих станів $F = \{q_1\}$.

Доведемо, що побудований МП-автомат M справді допускає слово $w \in T^*$ тоді й тільки тоді, коли існує лівостороннє виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$.

Зважаючи на вигляд продукцій для граматики у нормальній формі Хомського, легко зрозуміти, що лівосторонніми виведеннями із джерела S можна отримати лише слова вигляду $u\alpha$, де $u \in T^*$, $\alpha \in V^*$. Також зазначимо, що множина переходів Δ побудованого МП-автомата задає відповідність між продукціями граматики G та переходами МП-автомата M; для зручності зведемо цю відповідність у таблицю 4.1.

Таблиця 4.1

Продукція граматики	Перехід МП-автомата	Застосування продукції	Такт роботи МП-автомата
$S \to \varepsilon$	$((q_1, \varepsilon, S), (q_1, \varepsilon))$	$uS\alpha \underset{G}{\Rightarrow} u\alpha$	$(q_1, v, S\alpha) \vdash_M (q_1, v, \alpha)$
$A \to a$	$((q_1, a, A), (q_1, \varepsilon))$	$uA\alpha \underset{G}{\Rightarrow} ua\alpha$	$(q_1, av, A\alpha) \vdash_M (q_1, v, \alpha)$
$A \to BC$	$((q_1, \varepsilon, A), (q_1, BC))$	$uA\alpha \underset{G}{\Rightarrow} uBC\alpha$	$(q_1, v, A\alpha) \vdash_M (q_1, v, BC\alpha)$

I. Нехай $w \in T^*$ та $S \overset{*}{\underset{G}{\Rightarrow}} w$. Доведемо, що M допускає слово $w \in T^*$. Зафіксуємо лівостороннє виведення

$$S = u_1\alpha_1 \underset{G}{\Rightarrow} u_2\alpha_2 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} u_n\alpha_n = w, \tag{4.7}$$

де $u_k \in T^*$, $\alpha_k \in V^*$, $k = 1, 2, \ldots, n$; зокрема, з першого та останнього кроку виведення (4.7) видно, що $u_1 = \varepsilon$, $\alpha_1 = S$, $u_n = w$, $\alpha_n = \varepsilon$. Індукцією за $k = 1, 2, \ldots, n$ доведемо, що $(q_0, w, \varepsilon) \vdash_M^* (q_1, v_k, \alpha_k)$, де слово $v_k \in T^*$ визначене співвідношенням $u_k v_k = w$.

1. База індукції. Нехай $k = 1$. Тоді за побудовою $u_1 = \varepsilon$, $v_1 = w$, $\alpha_1 = S$. На початку роботи єдино можливим для виконання є ε-перехід $((q_0, \varepsilon, \varepsilon), (q_1, S))$ – МП-автомат має перейти з початкового стану q_0 у допускаючий стан q_1, записавши у стек символ S. Отже,

$$(q_0, w, \varepsilon) = (q_0, v_1, \varepsilon) \vdash_M (q_1, v_1, S) = (q_1, v_1, \alpha_1).$$

2. Припущення індукції. Нехай твердження індукції виконується для $k = m$: $(q_0, w, \varepsilon) \vdash_M^* (q_1, v_m, \alpha_m)$, де слово $v_m \in T^*$ визначене співвідношенням $u_m v_m = w$.

3. Крок індукції. Нехай $k = m + 1$. У виведенні (4.7) на кроці $u_m\alpha_m \underset{G}{\Rightarrow} u_{m+1}\alpha_{m+1}$ можуть бути застосовані такі продукції (табл. 4.1):

- $S \to \varepsilon$, якщо $u_m = u_{m+1}$, $\alpha_m = S\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $A \to a$, якщо $u_{m+1} = u_m a$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $A \to BC$, якщо $u_m = u_{m+1}$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = BC\tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$),

де $A \in V$, $B \in V \setminus \{S\}$, $C \in V \setminus \{S\}$, $a \in T$. Таким чином, у кожному з цих випадків для МП-автомата M у конфігурації (q_1, v_m, α_m) може бути виконано відповідно один з таких переходів (табл. 4.1):

- $((q_1, \varepsilon, S), (q_1, \varepsilon))$, $v_m = v_{m+1}$, $\alpha_m = S\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $((q_1, a, A), (q_1, \varepsilon))$, $v_m = av_{m+1}$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $((q_1, \varepsilon, A), (q_1, BC))$, $v_m = v_{m+1}$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = BC\tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$).

В усіх трьох випадках з урахуванням припущення індукції отримуємо: $u_{m+1}v_{m+1} = u_m v_m = w$, що доводить твердження індукції для $k = m + 1$:

$$(q_0, w, \varepsilon) \vdash^*_M (q_1, v_m, \alpha_m) \vdash_M (q_1, v_{m+1}, \alpha_{m+1}),$$

де слово $v_{m+1} \in T^*$ визначене співвідношенням $u_{m+1}v_{m+1} = w$.

Отже, твердження індукції доведено, звідки за $k = n$ отримуємо: $(q_0, w, \varepsilon) \vdash^*_M (q_1, v_n, \alpha_n) = (q_1, \varepsilon, \varepsilon)$, тобто МП-автомат M допускає слово w.

II. Нехай M допускає слово $w \in T^*$, тобто $(q_0, w, \varepsilon) \vdash^*_M (q_1, \varepsilon, \varepsilon)$. Доведемо, що $S \overset{*}{\underset{G}{\Rightarrow}} w$. Як вже зазначалося в п. I, на початку роботи єдино можливим для виконання є ε-перехід $((q_0, \varepsilon, \varepsilon), (q_1, S))$ – МП-автомат має перейти з початкового стану q_0 у допускаючий стан q_1, записавши у стек символ S. Надалі, оскільки жодний перехід не передбачає повернення в q_0, МП-автомат до кінця роботи залишається у стані q_1. Зафіксуємо послідовність конфігурацій, що дозволяє перевести МП-автомат із початкової конфігурації (q_0, w, ε) у конфігурацію $(q_1, \varepsilon, \varepsilon)$:

$$(q_0, w, \varepsilon) \vdash_M (q_1, w, S) = (q_1, v_1, \alpha_1) \vdash_M (q_1, v_2, \alpha_2) \vdash_M \cdots \vdash_M (q_1, v_n, \alpha_n) = (q_1, \varepsilon, \varepsilon), \tag{4.8}$$

де $v_k \in T^*$, $\alpha_k \in V^*$, $k = 1, 2, \ldots, n$. Індукцією за $k = 1, 2, \ldots, n$ доведемо, що $S \overset{*}{\underset{G}{\Rightarrow}} u_k\alpha_k$, де слово $u_k \in T^*$ визначене співвідношенням $u_k v_k = w$.

1. База індукції. Нехай $k = 1$. Очевидно, $v_1 = w$, $\alpha_1 = S$, $u_1 = \varepsilon$, і твердження $S \overset{*}{\underset{G}{\Rightarrow}} u_1\alpha_1 = S$ виконується.

2. Припущення індукції. Нехай твердження індукції виконується для $k = m$: $S \overset{*}{\underset{G}{\Rightarrow}} u_m\alpha_m$, де слово $u_m \in T^*$ визначене співвідношенням $u_m v_m = w$.

3. Крок індукції. Нехай $k = m + 1$. Зважаючи на вміст множини переходів Δ, на такті $(q_1, v_m, \alpha_m) \vdash_M (q_1, v_{m+1}, \alpha_{m+1})$ у послідовності (4.8) може бути виконано такі переходи (табл. 4.1):

- $((q_1, \varepsilon, S), (q_1, \varepsilon))$, якщо $v_m = v_{m+1}$, $\alpha_m = S\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $((q_1, a, A), (q_1, \varepsilon))$, якщо $v_m = av_{m+1}$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $((q_1, \varepsilon, A), (q_1, BC))$, якщо $v_m = v_{m+1}$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = BC\tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$),

де $A \in V$, $B \in V \setminus \{S\}$, $C \in V \setminus \{S\}$, $a \in T$. Таким чином, у кожному із цих випадків у виведенні в КВ-граматиці G до слова $u_m\alpha_m$ може бути застосована відповідно одна з таких продукцій (табл. 4.1):

- $S \to \varepsilon$, $u_m = u_{m+1}$, $\alpha_m = S\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $A \to a$, $u_{m+1} = u_m a$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = \tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$);
- $A \to BC$, $u_m = u_{m+1}$, $\alpha_m = A\tilde{\alpha}_m$, $\alpha_{m+1} = BC\tilde{\alpha}_m$ ($\tilde{\alpha}_m \in V^*$).

В усіх трьох випадках з урахуванням припущення індукції отримуємо: $u_{m+1}v_{m+1} = u_m v_m = w$, що доводить твердження індукції для $k = m + 1$: $S \overset{*}{\underset{G}{\Rightarrow}} u_{m+1}\alpha_{m+1}$, де $u_{m+1} \in T^*$ визначене співвідношенням $u_{m+1}v_{m+1} = w$.

Отже, твердження індукції доведено, звідки за $k = n$ отримуємо виведення $S \overset{*}{\underset{G}{\Rightarrow}} u_n\alpha_n = w$, тобто КВ-граматика G породжує слово w.

Таким чином, для будь-якого $w \in T^*$ твердження $(q_0, w, \varepsilon) \overset{*}{\vdash}_M (q_1, \varepsilon, \varepsilon)$ та $S \overset{*}{\underset{G}{\Rightarrow}} w$ еквівалентні, тобто МП-автомат M допускає w тоді й тільки тоді, коли w можна отримати із джерела S продукціями множини P граматики G. Теорему повністю доведено. $\square$

Зауваження 4.12. Доведення теореми 4.3 конструктивне, тобто надає конкретний метод побудови відповідного МП-автомата за заданою КВ-граматикою, однак цей метод потребує зведення граматики G до нормальної форми Хомського, що у практичних випадках може бути обтяжливим. Наведемо без доведення інший варіант побудови МП-автомата за довільною КВ-граматикою $G = \langle V, T, P, S \rangle$, який моделює лівосторонні виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$ (детальніше див. [4, 9, 11, 14]):

$$M = \langle \{q_0, q_1\}, T, V \cup T, \Delta, \{q_0\}, \{q_1\} \rangle,$$
$$\Delta = \{((q_0, \varepsilon, \varepsilon), (q_1, S))\} \cup \{((q_1, a, a), (q_1, \varepsilon)) \colon a \in T\} \cup$$
$$\cup \{((q_1, \varepsilon, A), (q_1, \beta)) \colon (A \to \beta) \in P\}.$$

Приклад 4.17. Побудуємо МП-автомат, який допускає КВ-мову $L = \{a^n b^n : n \geqslant 0\}$, використовуючи метод із доведення теореми 4.3. Породжувальна КВ-граматика G у нормальній формі Хомського має вигляд (див. приклади 4.10 та 4.11)

$$G = \langle \{S, S_0, T_a, T_b, C\}, \{a, b\}, P, S \rangle,$$
$$P = \{S \to T_a C | T_a T_b | \varepsilon, S_0 \to T_a C | T_a T_b, C \to S_0 T_b, T_a \to a, T_b \to b\}.$$

Шуканий МП-автомат має вигляд

$$M = \langle \{q_0, q_1\}, \{a, b\}, \{S, S_0, T_a, T_b, C\}, \Delta, \{q_0\}, \{q_1\} \rangle,$$
$$\Delta = \{((q_0, \varepsilon, \varepsilon), (q_1, S)), ((q_1, \varepsilon, S), (q_1, T_a C)), ((q_1, \varepsilon, S), (q_1, T_a T_b)),$$
$$((q_1, \varepsilon, S), (q_1, \varepsilon)), ((q_1, \varepsilon, S_0), (q_1, T_a C)), ((q_1, \varepsilon, S_0), (q_1, T_a T_b)),$$
$$((q_1, \varepsilon, C), (q_1, T_a T_b)), ((q_1, a, T_a), (q_1, \varepsilon)), ((q_1, b, T_b), (q_1, \varepsilon))\}.$$

Хоча МП-автомат M допускає мову L, існує еквівалентний простіший детермінований МП-автомат, який наведено у прикладі 4.13. Тобто зазвичай у практичних випадках для КВ-мови L можна навести МП-автомат, простіший за МП-автомат, побудований за методом теореми 4.3.

Вправа 4.20. Побудувати МП-автомат, який допускає КВ-мову $L = \{a^n b^n : n \geqslant 0\}$, використовуючи метод із зауваження 4.12. Порівняти з МП-автоматами прикладів 4.17 та 4.13, показати можливі дії МП-автомата для вхідних слів a^2b^2, a^2b^3, a^2b, aba.

Вправа 4.21. Побудувати МП-автомати, які допускають мови із прикладів 4.4–4.7.

4.4.2. Побудова КВ-граматики за МП-автоматом

Лема 4.1. *Будь-який МП-автомат* $M = \langle Q, T, \Gamma, \Delta, I, F \rangle$ *еквівалентний деякому МП-автомату* $M' = \langle Q', T, \Gamma, \Delta', I, F \rangle$*, такому, що для кожного переходу* $((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta'$ *справджується нерівність* $|\alpha| + |\beta| \leqslant 1$.

Доведення. Для кожного переходу $\delta = ((q_1, \xi, \alpha), (q_2, \beta_1\beta_2 \dots \beta_n)) \in \Delta$, $\alpha \in (\Gamma \cup \{\varepsilon\})$, $\beta_j \in \Gamma$ $(j = 1, 2, \dots, n)$, такого, що $|\alpha| + |\beta| > 1$, введемо нові стани $q_\delta^1 \notin Q$, $q_\delta^2 \notin Q$,..., $q_\delta^n \notin Q$ та замінимо перехід δ на $n+1$ переходів

$$((q_1, \xi, \alpha), (q_\delta^1, \varepsilon)), ((q_\delta^1, \varepsilon, \varepsilon), (q_\delta^2, \beta_1)), ((q_\delta^2, \varepsilon, \varepsilon), (q_\delta^3, \beta_2)), \dots,$$
$$((q_\delta^{n-1}, \varepsilon, \varepsilon), (q_\delta^n, \beta_{n-1})), ((q_\delta^n, \varepsilon, \varepsilon), (q_2, \beta_n)).$$

Очевидно, таке перетворення не змінює множину слів, які допускає МП-автомат, а отже $L[M] = L[M']$. □

Лема 4.2. *Будь-який МП-автомат* $M = \langle Q, T, \Gamma, \Delta, I, F \rangle$ *еквівалентний деякому МП-автомату* $M' = \langle Q', T, \Gamma, \Delta', \{q_0\}, \{q_f\} \rangle$*, який містить у точності один початковий стан* q_0 *та у точності один допускаючий стан* q_f*.*

Доведення. МП-автомат M еквівалентний МП-автомату

$$M' = \langle Q \cup \{q_0, q_f\}, T, \Gamma, \Delta', \{q_0\}, \{q_f\} \rangle,$$
$$\Delta' = \Delta \cup \{((q_0, \varepsilon, \varepsilon), (p_0, \varepsilon)) : p_0 \in I\} \cup \{((p_f, \varepsilon, \varepsilon), (q_f, \varepsilon)) : p_f \in F\},$$

де $q_0, q_f \notin Q$. □

Лема 4.3. *Будь-який МП-автомат* $M = \langle Q, T, \Gamma, \Delta, I, F \rangle$ *еквівалентний деякому МП-автомату* $M' = \langle Q', T, \Gamma', \Delta', \{q_0\}, \{q_f\} \rangle$*, який містить у точності один початковий стан* q_0*, у точності один допускаючий стан* q_f*, та для кожного переходу* $((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta'$ *справджується рівність* $|\alpha| + |\beta| = 1$.

Доведення. З урахуванням лем 4.1 та 4.2 можна без втрати загальності вважати, що МП-автомат M містить у точності один початковий стан q_0, у точності один допускаючий стан q_f, та для кожного переходу $((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta$ справджується рівність $|\alpha| + |\beta| \leqslant 1$. Для кожного переходу $\delta = ((q_1, \xi, \varepsilon), (q_2, \varepsilon)) \in \Delta$ (випадок $|\alpha| + |\beta| = 0$) введемо новий стан $q_\delta \notin Q$ і новий символ алфавіту магазинної пам'яті $\gamma_\delta \notin \Gamma$, й замінимо перехід δ двома переходами $((q_1, \xi, \varepsilon), (q_\delta, \gamma_\delta))$ та $((q_\delta, \varepsilon, \gamma_\delta), (q_2, \varepsilon))$. Очевидно, таке перетворення не змінює множину слів, які допускає МП-автомат, а отже $L[M] = L[M']$. □

Теорема 4.4. *Нехай формальну мову* L *над вхідним алфавітом* T *допускає деякий МП-автомат. Тоді мова* L *є контекстно-вільною.*

Доведення. Нехай формальну мову $L \subset T^*$ допускає МП-автомат $M = \langle Q, T, \Gamma, \Delta, I, F\rangle$. З урахуванням леми 4.3 можна вважати, що:

1) $I = \{q_0\}$, $F = \{q_f\}$;

2) для кожного переходу $((q_1, \xi, \alpha), (q_2, \beta)) \in \Delta$ справджується рівність $|\alpha| + |\beta| = 1$.

Визначимо шукану КВ-граматику $G = \langle V, T, P, S\rangle$:

- $V = \{A_{p,q} \colon p, q \in Q\}$;
- $P = \{A_{p,q} \to \xi_1 A_{p_2,q_1} \xi_2 A_{q_2,q} \colon$
  $((p, \xi_1, \varepsilon), (p_2, \alpha)), ((q_1, \xi_2, \alpha), (q_2, \varepsilon)) \in \Delta,\ q \in Q\} \cup$
  $\cup\, \{A_{q,q} \to \varepsilon \colon q \in Q\}$;
- $S = A_{q_0,q_f}$.

Доведемо, що для будь-якого слова $w \in T^*$ виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$ існує тоді й тільки тоді, коли МП-автомат M допускає w.

Нехай $u \in T^*$, $p, q \in Q$. Доведемо еквівалентність тверджень $(p, u, \varepsilon) \vdash^*_M (q, \varepsilon, \varepsilon)$ та $A_{p,q} \overset{*}{\underset{G}{\Rightarrow}} u$.

I. Для кожних $u \in T^*$, $p, q \in Q$, таких, що $(p, u, \varepsilon) \vdash^*_M (q, \varepsilon, \varepsilon)$, існує деяка послідовність конфігурацій, що переводить МП-автомат із конфігурації (p, u, ε) у конфігурацію $(q, \varepsilon, \varepsilon)$:

$$(p, u, \varepsilon) = (r_1, u_1, \alpha_1) \vdash_M (r_2, u_2, \alpha_2) \vdash_M \cdots$$
$$\cdots \vdash_M (r_{n_{u,p,q}-1}, u_{n_{u,p,q}-1}, \alpha_{n_{u,p,q}-1}) \vdash_M (r_{n_{u,p,q}}, u_{n_{u,p,q}}, \alpha_{n_{u,p,q}}) = (q, \varepsilon, \varepsilon), \quad (4.9)$$

де $r_k \in Q$, $u_k \in T^*$, $\alpha_k \in \Gamma^*$, $k = 1, 2, \ldots, n_{u,p,q}$. Індукцією за кількістю конфігурацій $n_{u,p,q} \geqslant 1$ у послідовності (4.9) доведемо: якщо $(p, u, \varepsilon) \vdash^*_M (q, \varepsilon, \varepsilon)$, то $A_{p,q} \overset{*}{\underset{G}{\Rightarrow}} u$.

1. База індукції. Нехай $n_{u,p,q} = 1$. Тоді, очевидно, $p = r_1 = q$ та $u = u_1 = \varepsilon$, звідки $(A_{p,q} \to u) = (A_{q,q} \to \varepsilon) \in P$.

2. Припущення індукції. Нехай $A_{p,q} \overset{*}{\underset{G}{\Rightarrow}} u$, якщо $(p, u, \varepsilon) \vdash^*_M (q, \varepsilon, \varepsilon)$ та $n_{u,p,q} \leqslant m$.

3. Крок індукції. Нехай $(p, u, \varepsilon) \vdash^*_M (q, \varepsilon, \varepsilon)$ та $n_{u,p,q} = m + 1$:

$$(p, u, \varepsilon) = (r_1, u_1, \alpha_1) \vdash_M (r_2, u_2, \alpha_2) \vdash_M \ldots$$
$$\ldots \vdash_M (r_m, u_m, \alpha_m) \vdash_M (r_{m+1}, u_{m+1}, \alpha_{m+1}) = (q, \varepsilon, \varepsilon). \quad (4.10)$$

Оскільки $|\alpha| + |\beta| = 1$ для кожного переходу $((q_1, \xi_1, \alpha), (q_2, \beta)) \in \Delta$, та $\alpha_1 = \varepsilon$, на першому такті має бути виконано перехід вигляду $((p, \xi_1, \varepsilon), (r_2, A)) \in \Delta$, де $A = \alpha_2 \in \Gamma$, $\xi_1 \in (T \cup \{\varepsilon\})$, $u_1 = \xi_1 u_2$. Далі, оскільки $\alpha_{m+1} = \varepsilon$, стек після другого такту принаймні один раз стає порожнім. Нехай $k = \min\limits_{3 \leqslant i \leqslant m+1} \{\alpha_i = \varepsilon\}$. Згідно з принципом «останнім зайшов, першим вийшов», на такті $(r_{k-1}, u_{k-1}, \alpha_{k-1}) \vdash_M (r_k, u_k, \alpha_k)$ має бути виконано перехід $((r_{k-1}, \xi_2, A), (r_k, \varepsilon))$. Отже, $u_{k-1} = \xi_2 u_k$ ($\xi_2 \in (T \cup \{\varepsilon\})$), $\alpha_{k-1} = A$, $\alpha_k = \varepsilon$, і послідовність (4.10) можна подати у вигляді

$$(p, u, \varepsilon) = (r_1, \xi_1 u_2, \varepsilon) \vdash_M (r_2, u_2, A) \vdash_M^* (r_{k-1}, \xi_2 u_k, A) \vdash_M (r_k, u_k, \varepsilon) \vdash_M^* \\ \vdash_M^* (r_m, u_m, \alpha_m) \vdash_M (r_{m+1}, u_{m+1}, \alpha_{m+1}) = (q, \varepsilon, \varepsilon).$$

Оскільки Δ містить переходи $((p, \xi_1, \varepsilon), (r_2, A))$ і $((r_{k-1}, \xi_2, A), (r_k, \varepsilon))$, P містить продукцію $A_{p,q} \to \xi_1 A_{r_2, r_{k-1}} \xi_2 A_{r_k, q}$. Далі із вибору номера $k = \min\limits_{3 \leqslant i \leqslant m+1} \{\alpha_i = \varepsilon\}$ випливає, що $\alpha_i = A\tilde{\alpha}_i$ ($\tilde{\alpha}_i \in \Gamma^*$) для всіх $2 \leqslant i \leqslant k-1$, тобто між тактами 2 та $k-1$ «на дні» стеку міститься символ A. Звідси з урахуванням $(r_2, u_2, A) = (r_2, u_{2,k} \xi_2 u_k, A) \vdash_M^* (r_{k-1}, \xi_2 u_k, A)$ отримуємо виведення

$$(r_2, u_{2,k}, \varepsilon) \vdash_M^* (r_{k-1}, \varepsilon, \varepsilon), \tag{4.11}$$

де слово $u_{2,k} \in T^*$ визначене співвідношенням $u_{2,k} \xi_2 u_k = u_2$. Із виведення (4.11) з урахуванням припущення індукції випливає існування виведення $A_{r_2, r_{k-1}} \overset{*}{\underset{G}{\Rightarrow}} u_{2,k}$. Крім того, із $(r_k, u_k, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$ з урахуванням припущення індукції випливає існування виведення $A_{r_k, q} \overset{*}{\underset{G}{\Rightarrow}} u_k$. Нарешті, із наявності продукції $A_{p,q} \to \xi_1 A_{r_2, r_{k-1}} \xi_2 A_{r_k, q}$ та існування виведень $A_{r_2, r_{k-1}} \overset{*}{\underset{G}{\Rightarrow}} u_{2,k}$ і $A_{r_k, q} \overset{*}{\underset{G}{\Rightarrow}} u_k$ випливає існування виведення

$$A_{p,q} \underset{G}{\Rightarrow} \xi_1 A_{r_2, r_{k-1}} \xi_2 A_{r_k, q} \overset{*}{\underset{G}{\Rightarrow}} \xi_1 u_{2,k} \xi_2 u_k = \xi_1 u_2 = u_1 = u.$$

Отже, твердження індукції для $n_{u,p,q} = m + 1$ доведено. Таким чином, доведено існування виведення $A_{p,q} \overset{*}{\underset{G}{\Rightarrow}} u$, якщо $(p, u, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$.

II. Для кожних $u \in T^*$, $p, q \in Q$, таких, що $A_{p,q} \overset{*}{\underset{G}{\Rightarrow}} u$, існує деяке виведення

$$A_{p,q} = \alpha_0 \underset{G}{\Rightarrow} \alpha_1 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} \alpha_{n_{u,p,q}} = u. \tag{4.12}$$

Індукцією за кількістю застосувань продукцій $n_{u,p,q} \geqslant 1$ у послідовності (4.12) доведемо: якщо $A_{p,q} \underset{G}{\overset{*}{\Rightarrow}} u$, то $(p, u, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$.

1. База індукції. Нехай $n_{u,p,q} = 1$. Тоді, очевидно, $A_{p,q} \underset{G}{\Rightarrow} u$, що, зважаючи на структуру множини продукцій P, можливо лише у випадку $p = q$ та $u = \varepsilon$ із застосуванням продукції $A_{q,q} \to \varepsilon$, звідки маємо виведення $(p, u, \varepsilon) = (q, \varepsilon, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$.

2. Припущення індукції. Нехай $(p, u, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$, якщо $A_{p,q} \underset{G}{\overset{*}{\Rightarrow}} u$ та $n_{u,p,q} \leqslant m$.

3. Крок індукції. Нехай $A_{p,q} \underset{G}{\overset{*}{\Rightarrow}} u$ та $n_{u,p,q} = m + 1$. Оскільки $n_{u,p,q} = m + 1 \geqslant 2$, слово α_1 має містити принаймні один нетермінальний символ. Отже, на першому кроці має бути застосована продукція вигляду $A_{p,q} \to \xi_1 A_{p_2,q_1} \xi_2 A_{q_2,q}$ $(\xi_1, \xi_2 \in (T \cup \{\varepsilon\}),\ A_{p_2,q_1}, A_{q_2,q} \in V)$, що дозволяє подати виведення (4.12) у вигляді

$$A_{p,q} = \alpha_0 \underset{G}{\Rightarrow} \xi_1 A_{p_2,q_1} \xi_2 A_{q_2,q} \underset{G}{\overset{*}{\Rightarrow}} u = \xi_1 u_{p_2,q_1} \xi_2 u_{q_2,q},$$

де A_{p_2,q_1} та $A_{q_2,q}$ розгортаються у слова $u_{p_2,q_1} \in T^*$ та $u_{q_2,q} \in T^*$ відповідно. Отже, існують виведення $A_{p_2,q_1} \underset{G}{\overset{*}{\Rightarrow}} u_{p_2,q_1}$ та $A_{q_2,q} \underset{G}{\overset{*}{\Rightarrow}} u_{q_2,q}$, і, до того ж, $n_{u_{p_2,q_1},p_2,q_1} + n_{u_{q_2,q},q_2,q} = m$. Звідси, враховуючи припущення індукції, отримуємо:

$$(p_2, u_{p_2,q_1}, \varepsilon) \vdash_M^* (q_1, \varepsilon, \varepsilon); \quad (q_2, u_{q_2,q}, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon). \tag{4.13}$$

Далі, оскільки P містить продукцію $A_{p,q} \to \xi_1 A_{p_2,q_1} \xi_2 A_{q_2,q}$, Δ містить переходи $((p, \xi_1, \varepsilon), (p_2, \gamma))$ та $((q_1, \xi_2, \gamma), (q_2, \varepsilon))$, де $\gamma \in \Gamma$. Отже, з урахуванням (4.13) отримуємо:

$$\begin{gathered}(p, u, \varepsilon) = (p, \xi_1 u_{p_2,q_1} \xi_2 u_{q_2,q}, \varepsilon) \vdash_M (p_2, u_{p_2,q_1} \xi_2 u_{q_2,q}, \gamma) \vdash_M^* \\ \vdash_M^* (q_1, \xi_2 u_{q_2,q}, \gamma) \vdash_M (q_2, u_{q_2,q}, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon).\end{gathered}$$

Отже, твердження індукції для $n_{u,p,q} = m + 1$ доведено. Таким чином, доведено: $(p, u, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$, якщо існує виведення $A_{p,q} \underset{G}{\overset{*}{\Rightarrow}} u$.

Враховуючи доведене у п. I та II, отримуємо еквівалентність тверджень $(p, u, \varepsilon) \vdash_M^* (q, \varepsilon, \varepsilon)$ та $A_{p,q} \underset{G}{\overset{*}{\Rightarrow}} u$. Звідси остаточно отримуємо: для будь-якого слова $w \in T^*$ виведення $S = A_{q_0,q_f} \underset{G}{\overset{*}{\Rightarrow}} w$ існує тоді й тільки тоді, коли МП-автомат M допускає w. Теорему повністю доведено. $\square$

Зауваження 4.13. Доведення теореми 4.4 конструктивне, тобто дає конкретний метод побудови породжувальної КВ-граматики за заданим МП-автоматом. Однак побудована КВ-граматика у більшості випадків є занадто складною, зокрема містить значну кількість «зайвих» нетермінальних символів. Так, якщо МП-автомат містить 5 станів, побудована за описаним методом КВ-граматика містить 25 нетермінальних символів і, ймовірно, допускає значне спрощення. Крім того, цей метод вимагає, щоб усі переходи МП-автомата задовольняли умови, які надає лема 4.3, що потребує перетворень за методом доведень лем 4.1–4.3 і в разі застосування збільшує кількість станів і переходів МП-автомата та, як результат, суттєво ускладнює шукану КВ-граматику.

Приклад 4.18. Використовуючи метод із доведення теореми 4.4, побудуємо породжувальну КВ-граматику для мови $L = \{ww^R\colon w \in \{a,b\}^*\}$. МП-автомат, що допускає L, побудовано у прикладі 4.15:

$$\begin{gathered} M = \langle \{q_0, q_1\}, \{a,b\}, \{A,B\}, \Delta, \{q_0\}, \{q_1\} \rangle, \\ \Delta = \{((q_0,a,\varepsilon),(q_0,A)), ((q_0,b,\varepsilon),(q_0,B)), ((q_0,\varepsilon,\varepsilon),(q_1,\varepsilon)), \\ ((q_1,a,A),(q_1,\varepsilon)), ((q_1,b,B),(q_1,\varepsilon))\}. \end{gathered}$$

Оскільки МП-автомат M містить перехід $((q_0,\varepsilon,\varepsilon),(q_1,\varepsilon))$, який не задовольняє одну з умов, що надає лема 4.3 (а саме $|\varepsilon| + |\varepsilon| = 0 \neq 1$), проведемо відповідні перетворення, ввівши новий стан q_c, новий символ магазинної пам'яті C та замінивши перехід $((q_0,\varepsilon,\varepsilon),(q_1,\varepsilon))$ двома переходами $((q_0,\varepsilon,\varepsilon),(q_c,C))$ і $((q_c,\varepsilon,C),(q_1,\varepsilon))$, отримуючи такий МП-автомат:

$$\begin{gathered} M' = \langle \{q_0, q_1, q_c\}, \{a,b\}, \{A,B,C\}, \Delta', \{q_0\}, \{q_1\} \rangle, \\ \Delta' = \{((q_0,a,\varepsilon),(q_0,A)), ((q_0,b,\varepsilon),(q_0,B)), ((q_0,\varepsilon,\varepsilon),(q_c,C)), \\ ((q_c,\varepsilon,C),(q_1,\varepsilon)), ((q_1,a,A),(q_1,\varepsilon)), ((q_1,b,B),(q_1,\varepsilon))\}. \end{gathered}$$

Далі, користуючись методом із доведення теореми 4.4, отримуємо КВ-граматику $\langle V, \{a,b\}, P, S \rangle$:

- $V = \{A_{q_0,q_0}, A_{q_0,q_1}, A_{q_0,q_c}, A_{q_1,q_0}, A_{q_1,q_1}, A_{q_1,q_c}, A_{q_c,q_0}, A_{q_c,q_1}, A_{q_c,q_c}\}$;
- $P = \{A_{q_0,q} \to aA_{q_0,q_1}aA_{q_1,q},\ A_{q_0,q} \to bA_{q_0,q_1}bA_{q_1,q},\ A_{q_0,q} \to A_{q_c,q_c}A_{q_1,q}\colon q \in \{q_0,q_1,q_c\}\} \cup \{A_{q,q} \to \varepsilon\colon q \in \{q_0,q_1,q_c\}\}$;
- $S = A_{q_0,q_1}$.

Очевидно, що з 9 нетермінальних символів лише 4 породжують слова над термінальним алфавітом $T = \{a, b\}$: A_{q_0,q_0}, A_{q_0,q_1}, A_{q_1,q_1}, A_{q_c,q_c}. Однак із джерела A_{q_0,q_1} неможливо вивести жодного слова, що містить символ A_{q_0,q_0}, отже символ A_{q_0,q_0} також є «зайвим». Видаливши «зайві» нетермінальні символи та продукції, які містять такі символи в лівій або правій частині, отримуємо еквівалентну граматику $\langle V', \{a, b\}, P', S\rangle$:

- $V' = \{A_{q_0,q_1}, A_{q_1,q_1}, A_{q_c,q_c}\}$;
- $P' = \{A_{q_0,q_1} \to aA_{q_0,q_1}aA_{q_1,q_1},\ A_{q_0,q_1} \to bA_{q_0,q_1}bA_{q_1,q_1},$
  $A_{q_0,q_1} \to A_{q_c,q_c}A_{q_1,q_1},\ A_{q_1,q_1} \to \varepsilon,\ A_{q_c,q_c} \to \varepsilon\}$;
- $S = A_{q_0,q_1}$.

Легко зрозуміти, що побудована граматика справді породжує формальну мову $L = \{ww^R \colon w \in \{a, b\}^*\}$. Проте цю граматику також можна дещо спростити: нетермінальний символ A_{q_1,q_1} міститься у лівій частині лише однієї продукції – $A_{q_1,q_1} \to \varepsilon$, а отже всі його входження можна замінити на ε; аналогічно, A_{q_c,q_c} міститься у лівій частині лише однієї продукції – $A_{q_c,q_c} \to \varepsilon$, і всі його входження також можна замінити на ε. Тепер символи A_{q_c,q_c} та A_{q_1,q_1} можна видалити разом із продукціями $A_{q_c,q_c} \to \varepsilon$ та $A_{q_1,q_1} \to \varepsilon$, отримуючи більш просту еквівалентну граматику $\langle V'', \{a, b\}, P'', S\rangle$:

- $V'' = \{A_{q_0,q_1}\}$;
- $P'' = \{A_{q_0,q_1} \to aA_{q_0,q_1}a,\ A_{q_0,q_1} \to bA_{q_0,q_1}b, A_{q_0,q_1} \to \varepsilon\}$;
- $S = A_{q_0,q_1}$.

Нарешті, залишивши для єдиного нетермінального символа традиційне для джерела позначення S, запишемо отриману граматику у вигляді

$$\langle\{S\}, \{a, b\}, \{S \to aSa|bSb|\varepsilon\}, S\rangle. \tag{4.14}$$

Приклад 4.19. Використовуючи метод із доведення теореми 4.4, побудуємо породжувальну КВ-граматику для мови $L \subset \{a, b\}^*$, що допускає такий МП-автомат:

$$M = \langle\{q_0\}, \{a, b\}, \{A, B\}, \Delta, \{q_0\}, \{q_0\}\rangle,$$
$$\Delta = \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, b, A), (q_0, \varepsilon)), ((q_0, b, \varepsilon), (q_0, B)),$$
$$((q_0, a, B), (q_0, \varepsilon)), ((q_0, \varepsilon, A), (q_0, \varepsilon))\}.$$

Умови, що надає лема 4.3, для наведеного МП-автомата виконуються. Користуючись методом із доведення теореми 4.4 і залишивши для єдиного нетермінального символа A_{q_0,q_0} традиційне для джерела позначення S, отримуємо КВ-граматику $\langle V, \{a,b\}, P, S\rangle$:

- $V = \{A_{q_0,q_0}\} = \{S\}$;
- $P = \{S \to aSbS,\ S \to bSaS,\ S \to aSS,\ S \to \varepsilon\}$.

Можна довести (див. приклад 4.3 і вправу 4.1, а також приклад 4.16 і вправу 4.16), що отримана граматика породжує ту саму мову, яку допускає вихідний МП-автомат, а саме мову $\{w \in \{a,b\}^* \colon |w|_a \geqslant |w|_b\}$.

Зауваження 4.14. Існує зв'язок між однозначними КВ-граматиками та детермінованими МП-автоматами. Для мови, яку допускає детермінований МП-автомат, існує породжувальна однозначна КВ-граматика (детальніше див., наприклад, [14]). У зворотний бік наслідку немає: наприклад, мову $\{ww^R : w \in \{a,b\}^*\}$ породжує однозначна КВ-граматика (4.14), проте її не допускає жодний детермінований МП-автомат (див. приклад 4.15 та зауваження 4.11).

Вправа 4.22. Використовуючи метод із доведення теореми 4.4, побудувати породжувальні КВ-граматики для мов із прикладів 4.13, 4.14 та вправ 4.12, 4.14.

4.5. Властивості замкнености класу контекстно-вільних мов

Теорема 4.5. *Нехай L_1 та L_2 – контекстно-вільні мови над алфавітом T. Тоді $L_1 \cup L_2$, $L_1 \cdot L_2$, L_1^* та L_1^R – контекстно-вільні мови над алфавітом T. (Клас контекстно-вільних мов замкнений відносно операцій об'єднання, конкатенації, замикання Кліні та обертання.)*

Доведення. **Об'єднання.** Нехай формальні мови L_1 та L_2 породжуються КВ-граматиками $\langle V_1, T, P_1, S_1\rangle$ та $\langle V_2, T, P_2, S_2\rangle$ відповідно, та $V_1 \cap V_2 = \varnothing$. Виберемо $S \notin (V_1 \cup V_2)$. Тоді мову $L_1 \cup L_2$ породжує така КВ-граматика (аналогічну техніку наведено в прикладі 1.13, п. 4):

$$\langle V_1 \cup V_2 \cup \{S\}, T, P_1 \cup P_2 \cup \{S \to S_1 | S_2\}, S\rangle.$$

Конкатенація. Нехай формальні мови L_1 та L_2 породжуються КВ-граматиками $\langle V_1, T, P_1, S_1\rangle$ та $\langle V_2, T, P_2, S_2\rangle$ відповідно, та $V_1 \cap V_2 = \varnothing$. Виберемо $S \notin (V_1 \cup V_2)$. Тоді мову $L_1 \cdot L_2$ породжує така КВ-граматика (див. також приклад 1.13, п. 5):

$$\langle V_1 \cup V_2 \cup \{S\}, T, P_1 \cup P_2 \cup \{S \to S_1 S_2\}, S\rangle.$$

Замикання Кліні. Нехай формальна мова L_1 породжується КВ-граматикою $\langle V_1, T, P_1, S_1\rangle$. Виберемо $S \notin V_1$. Тоді мову L_1^* породжує така КВ-граматика (див. також приклад 1.13, п. 6):

$$\langle V_1 \cup \{S\}, T, P_1 \cup \{S \to S_1 S | \varepsilon\}, S\rangle.$$

Обертання. Нехай формальна мова L_1 породжується КВ-граматикою $\langle V_1, T, P_1, S_1\rangle$. Тоді мову L_1^R породжує така КВ-граматика:

$$\langle V_1, T, \{A \to \alpha^R \colon A \in V, \alpha \in (V \cup T)^*, (A \to \alpha) \in P_1\}, S_1\rangle. \qquad \square$$

У прикладі 1.13 продемонстровано застосування техніки з доведення теореми 4.5 для побудови КВ-граматик, які породжують об'єднання, конкатенацію та замикання Кліні мов, породжених заданими КВ-граматиками.

Приклад 4.20. Формальну мову $L = \{a^n b^n : n \geqslant 0\}$ породжує КВ-граматика $\langle \{S\}, \{a, b\}, \{S \to aSb | \varepsilon\}, S\rangle$. Застосовуючи техніку з доведення теореми 4.5 для операції обертання, отримуємо КВ-граматику, що породжує мову $L^R = \{b^n a^n : n \geqslant 0\}$: $\langle \{S\}, \{a, b\}, \{S \to bSa | \varepsilon\}, S\rangle$.

Зазначимо, що клас КВ-мов незамкнений відносно операції перетину.

Приклад 4.21. Формальні мови $L_1 = \{a^m b^m c^n : m, n \geqslant 0\}$ та $L_2 = \{a^m b^n c^n : m, n \geqslant 0\}$ є контекстно-вільними (див. приклад 4.8). Але мова $L_1 \cap L_2 = \{a^n b^n c^n : n \geqslant 0\}$ не контекстно-вільна, що можна довести лемою про розростання (див. приклад 4.24 для мови $\{a^n b^n c^n : n \geqslant 1\}$).

Вправа 4.23. Довести, що клас контекстно-вільних мов незамкнений відносно операції доповнення.

Вказівка. Скориставшись законом де Моргана (див., наприклад, [7, 8]), отримати суперечність із незамкненістю класу КВ-мов відносно операції перетину.

Теорема 4.6. *Нехай L_1 – контекстно-вільна мова над алфавітом T, L_2 – регулярна мова над алфавітом T. Тоді мова $L_1 \cap L_2$ – контекстно-вільна мова над алфавітом T.*

Доведення. Для доведення теореми застосуємо техніку із зауваження 3.19. Нехай мову L_1 допускає автомат з магазинною пам'яттю $M_1 = \langle Q_1, T, \Gamma, \Delta_1, I_1, F_1 \rangle$, а мову L_2 допускає скінченний автомат $M_2 = \langle Q_2, T, \Delta_2, I_2, F_2 \rangle$ без ε-переходів. Тоді мову $L_1 \cap L_2$ допускає МП-автомат $M_3 = \langle Q_1 \times Q_2, T, \Gamma, \Delta' \cup \Delta'', I_1 \times I_2, F_1 \times F_2 \rangle$, де

$$\Delta' = \{(((p_1, p_2), a, \alpha), ((q_1, q_2), \beta)) \colon ((p_1, a, \alpha), (q_1, \beta)) \in \Delta_1, (p_2, a, q_2) \in \Delta_2\},$$
$$\Delta'' = \{(((p_1, p_2), \varepsilon, \alpha), ((q_1, p_2), \beta)) \colon ((p_1, \varepsilon, \alpha), (q_1, \beta)) \in \Delta_1\}.$$

Переходи із множини Δ' відповідають синхронній роботі автоматів M_1 та M_2 для поточного символа $a \in T$. Переходи із множини Δ'' відповідають виконанню МП-автоматом M_1 ε-переходу $((p_1, \varepsilon, \alpha), (q_1, \beta)) \in \Delta_1$ (зчитування поточного символа немає, проте дії зі стеком в загальному випадку можуть проводитись); скінченний автомат M_2 на поточному кроці залишається у стані p_2. □

Приклад 4.22. Нехай $L_1 = \{ww^R : w \in \{a, b\}^*\}$ (породжувальну КВ-граматику див. у прикладі 4.18), $L_2 = \{a^i b^j a^k \colon i, j, k \geqslant 0\} = L[a^* b^* a^*]$. Отже, L_1 та L_2 – відповідно контекстно-вільна та регулярна мови над алфавітом $\{a, b\}$. Мову L_1 допускає МП-автомат

$$M_1 = \langle \{q_0, q_1\}, \{a, b\}, \{A, B\}, \Delta_1, \{q_0\}, \{q_1\} \rangle,$$
$$\Delta_1 = \{((q_0, a, \varepsilon), (q_0, A)), ((q_0, b, \varepsilon), (q_0, B)), ((q_0, \varepsilon, \varepsilon), (q_1, \varepsilon)),$$
$$((q_1, a, A), (q_1, \varepsilon)), ((q_1, b, B), (q_1, \varepsilon))\}$$

(див. приклад 4.15), мову L_2 допускає скінченний автомат

$$M_2 = \langle \{p_0, p_1, p_2\}, \{a, b\}, \Delta_2, \{p_0\}, \{p_0, p_1, p_2\} \rangle,$$
$$\Delta_2 = \{(p_0, a, p_0), (p_0, b, p_1), (p_1, a, p_2),$$
$$(p_1, b, p_1), (p_2, a, p_2)\}$$

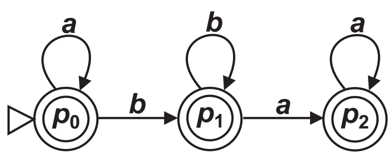

Рис. 4.17

без ε-переходів, граф якого показано на рис. 4.17.
Тоді за методом із доведення теореми 4.6 побудуємо МП-автомат, що допускає мову $L_1 \cap L_2$ (див. рис. 4.18):

$$M_3 = \langle \{(q_0, p_0), (q_0, p_1), (q_0, p_2), (q_1, p_0), (q_1, p_1), (q_1, p_2)\}, \{a, b\},$$
$$\{A, B\}, \Delta' \cup \Delta'', \{(q_0, p_0)\}, \{(q_1, p_0), (q_1, p_1), (q_1, p_2)\} \rangle.$$

Легко переконатись, що побудований МП-автомат M_3 справді допускає мову $L_1 \cap L_2 = \{a^i b^{2m} a^i \colon i, m \geqslant 0\}$.

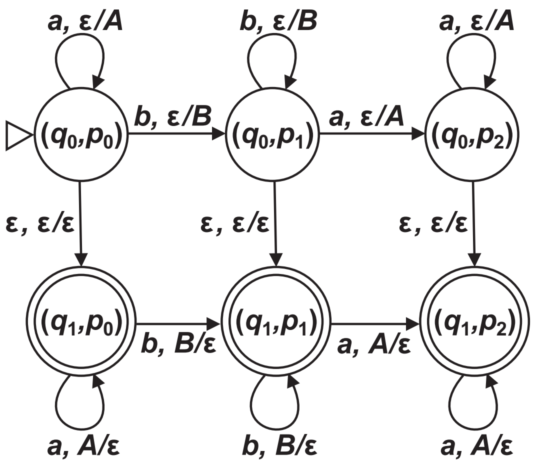

Рис. 4.18

Вправа 4.24. Для МП-автомата M_3 з прикладу 4.22 перевірити:

$$((q_0, p_0), a^{2i}, \varepsilon) \overset{*}{\vdash} ((q_1, p_0), \varepsilon, \varepsilon) \quad (i \geqslant 0),$$
$$((q_0, p_0), b^{2m}, \varepsilon) \overset{*}{\vdash} ((q_1, p_1), \varepsilon, \varepsilon) \quad (m \geqslant 1),$$
$$((q_0, p_0), a^i b^{2m} a^i, \varepsilon) \overset{*}{\vdash} ((q_1, p_2), \varepsilon, \varepsilon)$$
$$(i, m \geqslant 1).$$

Показати можливі дії МП-автомата M_3 для вхідних слів a^6, b^6 та $a^2b^4a^2$.

Зауваження 4.15. Теорему 4.6 також можна доводити, побудувавши відповідну породжувальну КВ-граматику (див., наприклад, [9]).

Зауваження 4.16. Теорема Хомського–Шютценберже[1] надає можливість зображувати довільну КВ-мову як гомоморфний образ перетину деякої мови Діка (див. приклад 4.6) та деякої регулярної мови (точне формулювання та доведення теореми наведено, наприклад, у [9, 23], детальніше про гомоморфізм алгебричних структур див., наприклад, [7, 9]).

4.6. Лема про розростання для контекстно-вільних мов

Нагадаємо деякі поняття, пов'язані з кореневими деревами (детальніше див., наприклад [7, 8]):
- гілка – шлях без повторень вершин, що з'єднує корінь дерева з одним із листів;
- довжина гілки – кількість ребер, які містить гілка;
- висота дерева – довжина найбільшої гілки.

Лема 4.4. *Нехай $G = \langle V, T, P, S\rangle$ – КВ-граматика у нормальній формі Хомського, $\mathcal{D}$ – дерево розбору для деякого виведення $A \overset{*}{\underset{G}{\Rightarrow}} w$ ($A \in V$, $w \in T^+$), $h_{\mathcal{D}}$ – висота $\mathcal{D}$, $n_{\mathcal{D}}$ – кількість листів у $\mathcal{D}$. Тоді справджується нерівність $n_{\mathcal{D}} \leqslant 2^{h_{\mathcal{D}}-1}$.*

[1]Шютценберже Марсель-Поль (1920–1996) – видатний французький вчений, отримав важливі результати у галузях формальної лінгвістики та комбінаторики.

Доведення. Твердження леми доведемо індукцією за $h_{\mathcal{D}} \geqslant 1$.

1. База індукції. Нехай $h_{\mathcal{D}} = 1$. Тоді з урахуванням виду продукцій для КВ-граматики у нормальній формі Хомського виведення $A \overset{*}{\underset{G}{\Rightarrow}} w$ має вигляд $A \underset{G}{\Rightarrow} w = a \in T$ із застосуванням єдиної продукції $A \to a$. Таким чином, $n_{\mathcal{D}} = 1 \leqslant 2^{h_{\mathcal{D}}-1} = 1$.

2. Припущення індукції. Нехай $n_{\mathcal{D}} \leqslant 2^{h_{\mathcal{D}}-1}$, якщо $h_{\mathcal{D}} \leqslant m$.

3. Крок індукції. Нехай $h_{\mathcal{D}} = m+1$. Тоді з урахуванням виду продукцій для КВ-граматики у нормальній формі Хомського виведення $A \overset{*}{\underset{G}{\Rightarrow}} w$ має вигляд $A \underset{G}{\Rightarrow} BC \overset{*}{\underset{G}{\Rightarrow}} w$ із застосуванням на першому кроці продукції $A \to BC$ ($B, C \in V \setminus \{S\}$). Отже, корінь дерева $\mathcal{D}$ має двох безпосередніх нащадків v_B та v_C з мітками B та C відповідно. Розглянемо піддерева $\mathcal{D}_B$ та $\mathcal{D}_C$ з коренями v_B та v_C відповідно. Очевидно, $\max\{h_{\mathcal{D}_B}, h_{\mathcal{D}_C}\} \leqslant h_{\mathcal{D}}-1 = m$. Далі, множина листів дерева $\mathcal{D}$ складається з листів піддерев $\mathcal{D}_B$ та $\mathcal{D}_C$, тобто $n_{\mathcal{D}} = n_{\mathcal{D}_B} + n_{\mathcal{D}_C}$. Отже, з урахуванням припущення індукції, отримуємо:

$$n_{\mathcal{D}} = n_{\mathcal{D}_B} + n_{\mathcal{D}_C} \leqslant 2^{h_{\mathcal{D}_B}-1} + 2^{h_{\mathcal{D}_C}-1} \leqslant 2 \cdot 2^{m-1} = 2^m.$$

Таким чином, твердження індукції для $h_{\mathcal{D}} = m+1$ доведено, що завершує доведення леми. □

Наслідок. *Нехай $G = \langle V, T, P, S\rangle$ – КВ-граматика у нормальній формі Хомського, $w \in L[G]$, $\mathcal{D}$ – дерево розбору для деякого виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$, $h_{\mathcal{D}}$ – висота $\mathcal{D}$. Тоді справджується нерівність $|w| \leqslant 2^{h_{\mathcal{D}}-1}$.*

Доведення. Достатньо зазначити, що у випадку $w \neq \varepsilon$ кожний лист дерева розбору $\mathcal{D}$ взаємно однозначно відповідає деякому символу у складі w, а отже $\mathcal{D}$ містить у точності $|w|$ листів. Випадок $w = \varepsilon$ є тривіальним, оскільки $|\varepsilon| = 0$. □

Теорема 4.7 (Лема про розростання для КВ-мов)**.** *Нехай L – довільна КВ-мова над алфавітом T. Тоді існує така константа $n \geqslant 1$, що будь-яке слово $w \in L$ довжиною $|w| \geqslant n$ можна зобразити у вигляді $w = xuyvz$, де $x, y, z, u, v \in T^*$, $|u| + |v| \geqslant 1$, $|uyv| \leqslant n$, так, що для всіх $i \geqslant 0$ виконується умова $xu^iyv^iz \in L$.*

Доведення. З урахуванням теореми 4.2 вважатимемо, що КВ-мова $L \subset T^*$ породжена КВ-граматикою $\langle V, T, P, S\rangle$ у нормальній формі Хомського.

Зафіксуємо $n = 2^{|V|}$, слово $w \in L$ довжиною $|w| \geqslant n$, деяке виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$ та відповідне дерево розбору $\mathcal{D}$ висотою $h_{\mathcal{D}}$. З урахуванням наслідку з леми 4.4 отримуємо нерівність

$$2^{h_{\mathcal{D}}-1} \geqslant |w| \geqslant n = 2^{|V|}. \tag{4.15}$$

Із (4.15) отримуємо оцінку на висоту дерева розбору: $h_{\mathcal{D}} > |V|$. Отже, дерево розбору $\mathcal{D}$ містить принаймні одну гілку $\mathcal{B}$ принаймні з $|V| + 1$ ребрами, або, що те ж саме, з $|V| + 2$ вершинами. Очевидно, що серед вершин будь-якої гілки дерева розбору $\mathcal{D}$ у точності одна вершина (лист) помічена термінальним символом – мітка ε у цьому разі неможлива, оскільки $w \neq \varepsilon$, і джерело S не входить до правих частин жодної продукції. Таким чином, гілка $\mathcal{B}$ містить принаймні $|V|+1$ вершин, які помічені нетермінальними символами, тобто принаймні дві вершини v_1 та v_2 серед останніх $|V| + 1$ вершин (починаючи від кореня) мають однакову мітку, яку позначимо через $A \in V$. З урахуванням повторення A на гілці $\mathcal{B}$, дереву розбору $\mathcal{D}$ поставимо у відповідність виведення

$$S \overset{*}{\underset{G}{\Rightarrow}} xAz \overset{*}{\underset{G}{\Rightarrow}} xuAvz \overset{*}{\underset{G}{\Rightarrow}} xuyvz, \tag{4.16}$$

де $x, y, z, u, v \in T^*$, і виведення $xAz \overset{*}{\underset{G}{\Rightarrow}} xuAvz$, як частина виведення (4.16), має містити застосування принаймні однієї продукції. Зазначимо, що виведення (4.16) може відрізнятися від вихідного виведення $S \overset{*}{\underset{G}{\Rightarrow}} w$, але обом цим виведенням відповідає одне дерево розбору $\mathcal{D}$. З урахуванням вигляду продукцій для граматики у нормальній формі Хомського першою продукцією у виведенні $xAz \overset{*}{\underset{G}{\Rightarrow}} xuAvz$ має бути продукція вигляду $A \to BC$ $(B, C \in V \setminus \{S\})$, звідки отримуємо, що u та v не можуть бути порожніми одночасно: $uv \neq \varepsilon$, або $|u| + |v| \geqslant 1$. Схематичне зображення дерева розбору для виведення (4.16) див. на рис. 4.19.

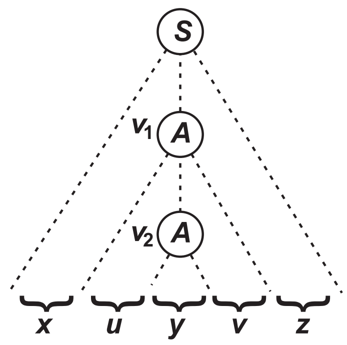

Рис. 4.19

Оскільки застосування продукцій при виведенні у КВ-граматиках не залежить від контексту, існування виведення $xAz \overset{*}{\underset{G}{\Rightarrow}} xuAvz$ означає наявність виведення $A \overset{*}{\underset{G}{\Rightarrow}} uAv$. Аналогічно, існування виведення

$xuAvz \overset{*}{\underset{G}{\Rightarrow}} xuyvz$ означає наявність виведення $A \overset{*}{\underset{G}{\Rightarrow}} y$. Повторюючи виведення $A \overset{*}{\underset{G}{\Rightarrow}} uAv$ (як реалізацію виведення $xAz \overset{*}{\underset{G}{\Rightarrow}} xuAvz$ у складі (4.16)) $i \geqslant 0$ разів, отримуємо:

$$S \overset{*}{\underset{G}{\Rightarrow}} xAz \overset{*}{\underset{G}{\Rightarrow}} xuAvz \overset{*}{\underset{G}{\Rightarrow}} xu^2Av^2z \overset{*}{\underset{G}{\Rightarrow}} \ldots \overset{*}{\underset{G}{\Rightarrow}} xu^iAv^iz \overset{*}{\underset{G}{\Rightarrow}} xu^iyv^iz. \qquad (4.17)$$

Зазначимо, що у випадку $i = 0$ виведення (4.17) набуває вигляду

$$S \overset{*}{\underset{G}{\Rightarrow}} xAz \overset{*}{\underset{G}{\Rightarrow}} xyz.$$

Нарешті, застосувавши лему 4.4 до піддерева $\mathcal{D}_{v_1}$ з коренем v_1, яке є деревом розбору для виведення $A \overset{*}{\underset{G}{\Rightarrow}} uyv$, отримуємо нерівність $|uyv| \leqslant 2^{h_{\mathcal{D}_{v_1}}-1}$, де $h_{\mathcal{D}_{v_1}}$ – висота дерева $\mathcal{D}_{v_1}$. Вершини v_1 та v_2 вибрані так, що $h_{\mathcal{D}_{v_1}} \leqslant |V| + 1$, звідки $|uyv| \leqslant 2^{|V|} = n$, що завершує доведення теореми. $\square$

Приклад 4.23. Нехай L – довільна скінченна мова. Оскільки будь-яка скінченна мова є регулярною (див. вправу 1.3), а отже й контекстно-вільною, для формальної мови L має виконуватися твердження теореми 4.7. Справді, як і у прикладі 3.46, виберемо $n = \max\{|w| \colon w \in L\} + 1$ та отримуємо правдивість твердження теореми завдяки відсутності слів $w \in L$ довжиною $|w| \geqslant n$.

Аналогічно лемі про розростання для регулярних мов (теорема 3.12), одне з основних застосувань леми про розростання для КВ-мов – доведення того, що задана мова не є контекстно-вільною.

Приклад 4.24. Доведемо, що мова $L = \{a^mb^mc^m : m \geqslant 1\}$ не є контекстно-вільною. Припустимо, що L контекстно-вільна, і n – додатна константа, існування якої постулюється лемою про розростання для КВ-мов. Тоді для слова $w = a^nb^nc^n$ мають існувати такі слова $x, y, z, u, v \in \{a, b, c\}^*$, $|u| + |v| \geqslant 1$, що $w = xuyvz$, $\widetilde{w} = xu^2yv^2z \in L$.

Доведемо, що слова u та v не містять ані ab, ані bc. Справді, якщо u (аналогічно – v) містить ab, тобто $u = a^{i+1}b^{j+1}c^k$ $(i, j, k \geqslant 0)$, тоді слово $u^2 = a^{i+1}b^{j+1}c^ka^{i+1}b^{j+1}c^k$ містить ba або ca, і $\widetilde{w} \notin L$; якщо u (аналогічно – v) містить bc, тобто $u = a^ib^{j+1}c^{k+1}$ $(i, j, k \geqslant 0)$, тоді слово $u^2 = a^ib^{j+1}c^{k+1}a^ib^{j+1}c^{k+1}$ містить ca або cb, і $\widetilde{w} \notin L$. Отже, кожне зі слів u та v містять або тільки a, або тільки b, або тільки c, тобто принаймні одна з літер a, b чи c не входить ані до u, ані до v: $|u|_a + |v|_a = 0$, або $|u|_b + |v|_b = 0$, або $|u|_c + |v|_c = 0$.

А між тим $|w|_a = |w|_b = |w|_c$ та $|\widetilde{w}|_a = |\widetilde{w}|_b = |\widetilde{w}|_c$, тобто

$$\begin{aligned}
|x|_a + |u|_a + |y|_a + |v|_a + |z|_a &= |x|_b + |u|_b + |y|_b + |v|_b + |z|_b = \\
&= |x|_c + |u|_c + |y|_c + |v|_c + |z|_c; \\
|x|_a + 2|u|_a + |y|_a + 2|v|_a + |z|_a &= |x|_b + 2|u|_b + |y|_b + 2|v|_b + |z|_b = \\
&= |x|_c + 2|u|_c + |y|_c + 2|v|_c + |z|_c,
\end{aligned}$$

звідки $|u|_a + |v|_a = |u|_b + |v|_b = |u|_c + |v|_c$.

Таким чином, $|u|_a + |v|_a = |u|_b + |v|_b = |u|_c + |v|_c = 0$, тобто $|u| = |v| = 0$, що суперечить умові $|u| + |v| \geqslant 1$. Отримана суперечність доводить, що мова L не є контекстно-вільною.

Приклад 4.25. Доведемо, що мова $L = \{w_0^2 : w_0 \in \{a,b\}^+\}$ не є контекстно-вільною. Припустимо, що L контекстно-вільна, і n – додатна константа, існування якої постулюється лемою про розростання для КВ-мов. Тоді для слова $w = a^{n+1}b^{n+1}a^{n+1}b^{n+1}$ мають існувати такі слова $x, y, z, u, v \in \{a,b\}^*$, $|u| + |v| \geqslant 1$, $|uyv| \leqslant n$, що $w = xuyvz$, $\widetilde{w} = xu^0yv^0z = xyz \in L$. Із нерівності $|uyv| \leqslant n$ випливає, що $\widetilde{w} = a^{n_1}b^{n_2}a^{n_3}b^{n_4}$ $(n_1, n_2, n_3, n_4 \geqslant 1)$. Оскільки $w = aw_1b$ $(w_1 \in \{a,b\}^*)$, для $\widetilde{w} \in L$ отримуємо: $\widetilde{w} = xyz = \widetilde{w}_0\widetilde{w}_0$, $\widetilde{w}_0 = a\widetilde{w}_1b$ $(\widetilde{w}_1 \in \{a,b\}^*)$. Враховуючи умову $|uyv| \leqslant n$, маємо такі випадки:

- $xyz = a^ib^ja^{n+1}b^{n+1}$ $(i, j \geqslant 1)$, $\widetilde{w}_0 = a^ib^j = a^{n+1}b^{n+1}$;
- $xyz = a^{n+1}b^ja^ib^{n+1}$ $(i, j \geqslant 1)$, $\widetilde{w}_0 = a^{n+1}b^j = a^ib^{n+1}$;
- $xyz = a^{n+1}b^{n+1}a^ib^j$ $(i, j \geqslant 1)$, $\widetilde{w}_0 = a^{n+1}b^{n+1} = a^ib^j$.

Отже, в усіх трьох випадках $xyz = a^{n+1}b^{n+1}a^{n+1}b^{n+1} = xuyvz$, звідки $u = v = \varepsilon$, що суперечить умові $|u| + |v| \geqslant 1$. Отримана суперечність доводить, що мова L не є контекстно-вільною.

Підкреслимо, що лема про розростання для КВ-мов надає лише необхідну, але не достатню умову, що задана мова є контекстно-вільною, тобто твердження теореми 4.7 може виконуватися й для мов, які не контекстно-вільні.

Приклад 4.26. Для формальної мови

$$L = \{a^jb^mc^md^m : j, m \geqslant 1\} \cup \{b^{m_1}c^{m_2}d^{m_3} : m_1, m_2, m_3 \geqslant 0\}$$

можна вибрати константу $n = 1$, і довільне $w \in L$ довжиною $|w| \geqslant 1$ має вигляд $aa^j b^m c^m d^m$ ($j \geqslant 0$, $m \geqslant 1$), або $b^{m_1} c^{m_2} d^{m_3}$ ($m_1 \geqslant 0$, $m_2 \geqslant 0$, $m_3 \geqslant 0$). В усіх випадках можна використати розклад $w = xuyvz$ із $x = y = v = \varepsilon$, $|u| = 1$, тобто за u взяти перший символ слова w. Легко перевірити, що цей розклад в усіх випадках відповідає твердженню теореми 4.7:

1) якщо $w = aa^j b^m c^m d^m = \varepsilon \cdot a \cdot \varepsilon \cdot \varepsilon \cdot a^j b^m c^m d^m$, то для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot a^i \cdot \varepsilon \cdot \varepsilon \cdot a^j b^m c^m d^m \in L$;

2) нехай $w = b^{m_1} c^{m_2} d^{m_3}$; якщо $m_1 \geqslant 1$, то $w = \varepsilon \cdot b \cdot \varepsilon \cdot \varepsilon \cdot b^{m_1 - 1} c^{m_2} d^{m_3}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot b^i \cdot \varepsilon \cdot \varepsilon \cdot b^{m_1 - 1} c^{m_2} d^{m_3} \in L$; якщо $m_1 = 0$, $m_2 \geqslant 1$, то $w = \varepsilon \cdot c \cdot \varepsilon \cdot \varepsilon \cdot c^{m_2 - 1} d^{m_3}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot c^i \cdot \varepsilon \cdot \varepsilon \cdot c^{m_2 - 1} d^{m_3} \in L$; якщо $m_1 = m_2 = 0$, то $w = \varepsilon \cdot d \cdot \varepsilon \cdot \varepsilon \cdot d^{m_3 - 1}$, і для всіх $i \geqslant 0$ виконується умова $\varepsilon \cdot d^i \cdot \varepsilon \cdot \varepsilon \cdot d^{m_3 - 1} \in L$.

Отже, задана мова L задовольняє твердження леми про розростання для КВ-мов, однак ця мова не контекстно-вільна. Справді, якщо б L була контекстно-вільною, то, оскільки мова $\{ab^{m_1} c^{m_2} d^{m_3} : m_1, m_2, m_3 \geqslant 1\}$ регулярна, мова $L \cap \{ab^{m_1} c^{m_2} d^{m_3} : m_1, m_2, m_3 \geqslant 1\} = \{ab^m c^m d^m : m \geqslant 1\}$ згідно з теоремою 4.6 була б контекстно-вільною. Однак формальна мова $\{ab^m c^m d^m : m \geqslant 1\}$ не є контекстно-вільною, що неважко встановити лемою про розростання, аналогічно випадку мови $\{a^m b^m c^m : m \geqslant 1\}$ (див. приклад 4.24).

Зауваження 4.17. У прикладах 3.49 та 3.50 було доведено нерегулярність мов $\{a^q : q$ – просте число$\}$ та $\{a^{m^2} : m \geqslant 1\}$ відповідно. Однак ці мови також не контекстно-вільні, оскільки кожна КВ-мова над односимвольним алфавітом є регулярною (див., наприклад, [9]).

Запитання та завдання для самоконтролю

1. Навести означення контекстно-вільної формальної граматики та контекстно-вільної формальної мови.

2. Навести означення дерева розбору слова, яке відповідає виведенню.

3. Яке виведення називають лівостороннім (правостороннім)?

4. Навести означення однозначної контекстно-вільної граматики, однозначної контекстно-вільної мови та суттєво неоднозначної контекстно-вільної мови, навести відповідні приклади.

5. Яку формальну мову називають мовою Діка?

6. Яку формальну мову називають мовою Лукасевича?

7. Навести означення нормальної форми Хомського та описати метод зведення контекстно-вільної граматики до нормальної форми Хомського.

8. Навести неформальний опис та принцип роботи автомата з магазинною пам'яттю (МП-автомата). Які дії виконує МП-автомат для переходу $((q_1, \xi, \alpha), (q_2, \beta))$?

9. Навести формальне означення МП-автомата.

10. Які МП-автомати називають детермінованими? Яка, на вашу думку, перевага таких автоматів порівняно з недетермінованими, і навпаки?

11. Наведіть означення конфігурації та такту роботи МП-автомата. Як вводять відношення «$\vdash^*_M$»? Яку конфігурацію називають тупиковою?

12. Дати визначення поняттю «МП-автомат допускає формальну мову». Які МП-автомати називають еквівалентними?

13. Чи допускають детерміновані МП-автомати та недетерміновані МП-автомати той самий клас мов? Якщо ні, навести відповідний приклад.

14. Сформулювати властивості замкненості класу КВ-мов, навести методи побудови відповідних граматик.

15. Навести схему побудови МП-автомата, який допускає перетин контекстно-вільної та регулярної мов.

16. Сформулювати лему про розростання для КВ-мов.

Розділ 5

Лінійно обмежені машини Тьюрінга та контекстно-залежні граматики

5.1. Невкорочувальні граматики

Нагадаємо, що формальну граматику $G = \langle V, T, P, S\rangle$ називають контекстно-залежною (КЗ-граматикою) або граматикою типу 1, якщо множина P містить лише продукції вигляду $\gamma_1 A \gamma_2 \to \gamma_1 \alpha \gamma_2$, де $A \in V$; $\alpha \in (V \cup T)^+$; $\gamma_1, \gamma_2 \in (V \cup T)^*$ (див. підрозд. 1.4). Іноді дозволяють появу однієї ε-продукції $S \to \varepsilon$, де S – джерело, за умови, що символ S не міститься у правій частині жодної продукції (див. зауваження 1.6).

Означення 5.1. Граматику $G = \langle V, T, P, S\rangle$ називають «невкорочувальною», якщо для кожної продукції вигляду $p = (\alpha \to \beta) \in P$ справджується нерівність $|\alpha| \leqslant |\beta|$.

Зауважимо, що граматики з прикладу 1.13, п. 3 та прикладу 1.15 є невкорочувальними. Продукції КЗ-граматик не зменшують довжину поточного слова, отже, кожна КЗ-граматика є невкорочувальною. Виявляється, що клас мов, породжених невкорочувальними граматиками, повністю вичерпується КЗ-мовами.

Теорема 5.1. *Нехай формальна мова L породжена невкорочувальною граматикою $G = \langle V, T, P, S \rangle$. Тоді L є КЗ-мовою.*

Доведення. Для кожного термінального символа $a \in T$, який міститься в лівій частині хоча б однієї продукції, введемо новий нетермінальний символ $T_a \notin V$, замінимо усі входження символа a в усіх продукціях на T_a та введемо нову продукцію $T_a \to a$. Таким чином позбуваємось термінальних символів у лівих частинах продукцій. Побудована граматика G_1 еквівалентна граматиці G.

Далі для кожної продукції $p_i = (A_1A_2 \dots A_m \to \beta_1\beta_2 \dots \beta_n) \in P$ $(n \geqslant m \geqslant 2)$ не дозволеного для КЗ-граматик вигляду введемо нові нетермінальні символи $X_k^i \notin V$ $(1 \leqslant k \leqslant m)$. Замінимо кожну продукцію p_i серією продукцій дозволеного вигляду:

$$\begin{aligned}
A_1A_2A_3 \dots A_{m-1}A_m &\to X_1^iA_2A_3 \dots A_{m-1}A_m,\\
X_1^iA_2A_3 \dots A_{m-1}A_m &\to X_1^iX_2^iA_3 \dots A_{m-1}A_m,\\
&\vdots\\
X_1^iX_2^iX_3^i \dots X_{m-1}^iA_m &\to X_1^iX_2^iX_3^i \dots X_{m-1}^iX_m^i,\\
X_1^iX_2^iX_3^i \dots X_{m-1}^iX_m^i &\to \beta_1X_2^iX_3^i \dots X_{m-1}^iX_m^i,\\
\beta_1X_2^iX_3^i \dots X_{m-1}^iX_m^i &\to \beta_1\beta_2X_3^i \dots X_{m-1}^iX_m^i,\\
&\vdots\\
\beta_1\beta_2 \dots \beta_{m-2}X_{m-1}^iX_m^i &\to \beta_1\beta_2 \dots \beta_{m-1}X_m^i,\\
\beta_1\beta_2\beta_3 \dots \beta_{m-1}X_m^i &\to \beta_1\beta_2\beta_3 \dots \beta_{m-1}\beta_m\beta_{m+1} \dots \beta_n.
\end{aligned}$$

Зазначимо, що символи A_j можуть міститись у різних продукціях, однак унікальність символів X_k^i та контекстність замін гарантує неможливість застосування наведених продукцій інакше, ніж для заміни слова $A_1A_2 \dots A_m$ на $\beta_1\beta_2 \dots \beta_n$. Також зазначимо, що продукція $\varepsilon \to \varepsilon$ невкорочувальної граматики не змінює мову, породжену граматикою, а продукцію $\varepsilon \to \beta$ $(|\beta| \geqslant 1)$ можна замінити згідно із зауваженням 1.5. Побудована граматика G_2 еквівалентна граматиці G_1, а тому й G. □

Зауваження 5.1. Якщо термінальний символ $a \in T$ міститься в лівій частині деяких продукцій, але всі такі продукції дозволеного вигляду, то зазвичай у практичних випадках можна не вводити відповідний нетермінальний символ T_a.

Зауваження 5.2. Іноді в літературі (див., наприклад, [4]) під КЗ-граматикою розуміють граматику, яка задовольняє означення 5.1.

Приклад 5.1. Розглянемо граматику

$$G = \langle\{S, B, C\}, \{a, b, c\}, \{S \to aSBC|aBC,\\ CB \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}, S\rangle,$$

яка породжує мову $L = \{a^n b^n c^n : n \geqslant 1\}$: легко перевірити, що G породжує ті й тільки ті слова, які належать L. Наприклад, слово $a^2b^2c^2 \in L$ можна отримати продукціями із G двома різними виведеннями:

$$S \Rightarrow aSBC \Rightarrow a^2(BC)^2 \Rightarrow a^2B^2C^2 \Rightarrow a^2bBC^2 \Rightarrow a^2b^2C^2 \Rightarrow a^2b^2cC \Rightarrow a^2b^2c^2,\\ S \Rightarrow aSBC \Rightarrow a^2(BC)^2 \Rightarrow a^2bCBC \Rightarrow a^2bBC^2 \Rightarrow a^2b^2C^2 \Rightarrow a^2b^2cC \Rightarrow a^2b^2c^2.$$

Граматика G невкорочувальна, отже, за теоремою 5.1 мова $L = L[G]$ контекстно-залежна. Між тим мова L не контекстно-вільна згідно з лемою про розростання для КВ-мов (див. приклад 4.24). Побудуємо еквівалентну КЗ-граматику за методом із доведення теореми 5.1. Згідно із зауваженням 5.1 позбуватись термінальних символів у лівих частинах продукцій не потрібно, оскільки всі такі продукції дозволеного вигляду. Єдину продукцію $CB \to BC$ недозволеного вигляду замінимо чотирма продукціями

$$CB \to X_1B, \quad X_1B \to X_1X_2, \quad X_1X_2 \to BX_2, \quad BX_2 \to BC,$$

отримуючи таким чином граматику

$$\langle\{S, B, C, X_1, X_2\}, \{a, b, c\}, \{S \to aSBC|aBC, CB \to X_1B, X_1B \to X_1X_2,\\ X_1X_2 \to BX_2, BX_2 \to BC, aB \to ab, bB \to bb, bC \to bc, cC \to cc\}, S\rangle.$$

Зауваження 5.3. Відомо (див. [9,10]), що клас КЗ-мов замкнений відносно операцій об'єднання, перетину, конкатенації, доповнення, обертання та замикання Кліні.

Вправа 5.1. Довести, що формальні мови із прикладів 4.25 і 4.26 контекстно-залежні, побудувавши відповідні невкорочувальні граматики.

Означення 5.2. Контекстно-залежну граматику $G = \langle V, T, P, S\rangle$ називають граматикою у нормальній формі Куроди[1], якщо P містить лише

[1]Курода Сігеюкі (1934–2009) – японський та американський вчений, спеціаліст у математичній лінгвістиці.

продукції вигляду $A \to a$, $A \to B$, $A \to BC$, $AB \to CD$, де $A \in V$, $B \in V$, $C \in V$, $D \in V$, $a \in T$.

Очевидно, що кожна граматика у формі Куроди є невкорочувальною граматикою, а отже згідно з теоремою 5.1 породжує КЗ-мову. Справджується також і зворотний факт: кожна КЗ-мова, яка не містить ε, може бути породжена граматикою у нормальній формі Куроди (доведення див. [28]).

Приклад 5.2. Покажемо зведення граматики з прикладу 5.1 до нормальної форми Куроди, не вказуючи загальний алгоритм такого зведення. Введемо нові нетермінальні символи T_a, T_b і T_c, замінимо у правій частині кожної продукції всі входження a на T_a, b на T_b і c на T_c та додамо до множини P продукції $T_a \to a$, $T_b \to b$ і $T_c \to c$. Отримуємо еквівалентну граматику без термінальних символів у правих частинах продукцій, окрім $T_a \to a$, $T_b \to b$ і $T_c \to c$:

$$\langle \{S, B, C, T_a, T_b, T_c\}, \{a, b, c\}, \{S \to T_aSBC | T_aBC, CB \to BC, T_aB \to T_aT_b,$$
$$T_bB \to T_bT_b, T_bC \to T_bT_c, T_cC \to T_cT_c, T_a \to a, T_b \to b, T_c \to c\}, S\rangle.$$

Далі дві продукції $S \to T_aSBC$ і $S \to T_aBC$ недозволеного вигляду замінимо чотирма продукціями

$$S \to T_aD_1, \quad D_1 \to SD_2, \quad D_2 \to BC, \quad S \to T_aD_2,$$

отримуючи таким чином граматику

$$\langle \{S, B, C, T_a, T_b, T_c, D_1, D_2\}, \{a, b, c\}, \{S \to T_aD_1 | T_aD_2,$$
$$D_1 \to SD_2, D_2 \to BC, CB \to BC, T_aB \to T_aT_b, T_bB \to T_bT_b,$$
$$T_bC \to T_bT_c, T_cC \to T_cT_c, T_a \to a, T_b \to b, T_c \to c\}, S\rangle.$$

5.2. Лінійно обмежені машини Тьюрінга

Лінійно обмежені машини Тьюрінга (інший термін лінійно обмежені автомати) порівняно з машинами Тьюрінга (див. підрозд. 2.1) мають обмеження на довжину стрічки: фактично можна використовувати лише ту частину стрічки, яка зайнята вхідним словом. Нагадаємо: робочим алфавітом X називають множину тих символів, які можуть містити комірки стрічки, а вхідне слово складається із символів зовнішнього алфавіту $T \subset X$.

Означення 5.3. Машину Тьюрінга $M = \langle Q, X, T, \Lambda, \Delta, I, F\rangle$ (взагалі кажучи, недетерміновану) називають лінійно обмеженою, якщо система команд не містить жодної команди вигляду $((q_1, \Lambda), (q_2, \alpha, d))$, де $\alpha \neq \Lambda$, $d \in \{l, r, s\}$.

Зокрема, машина Тьюрінга з прикладу 2.3 не є лінійно обмеженою.

Зауваження 5.4. Іноді в літературі лінійну обмеженість машини Тьюрінга визначають за допомогою маркерів (див., наприклад, [10]). Вхідне слово обмежують зліва та справа маркером – символом «#», а курсор машини Тьюрінга $\langle Q, X, T, \Lambda, \#, \Delta, I, F\rangle$ у процесі роботи не виходить за межі, визначені маркерами. Також є варіант обмеження вхідного слова зліва маркером «$\#_l$», а справа – «$\#_r$», тоді машину Тьюрінга можна записати у вигляді $\langle Q, X, T, \Lambda, \#_l, \#_r, \Delta, I, F\rangle$ (див., наприклад, [29]).

Зауваження 5.5. Лінійна обмеженість машини M еквівалентна існуванню деякої константи $k = k(M) \in \mathbb{N}$, яка залежить лише від машини M та не залежить від вхідного слова w, такої, що машина може використовувати лише $k|w|$ комірок стрічки. Таку машину можна розглядати в контексті означення 5.3, розширюючи алфавіт: наприклад, у випадку $k = 2$ та робочому алфавіті $\{a, b, \Lambda\}$ можна перейти до робочого алфавіту $\{t_{\Lambda\Lambda}, t_{\Lambda a}, t_{\Lambda b}, t_{a\Lambda}, t_{aa}, t_{ab}, t_{b\Lambda}, t_{ba}, t_{bb}\}$ й модифікувати множину станів і команд для моделювання курсора.

Кажуть, що машина M допускає (сприймає) слово $w \in T^+$, якщо $(\varepsilon, q_0, w) \vdash_M^* (\Lambda^k, q, \Lambda^m)$ для деякого початкового стану $q_0 \in I$, деякого заключного стану $q \in F$, $k \geqslant 0$, $m \geqslant 0$ (див. також означення 2.5). Множину слів $w \in T^+$, яку допускає машина M, називають мовою, яку допускає (сприймає) M. Вважатимемо за визначенням що порожнє слово $w = \varepsilon$ не може сприйматися лінійно обмеженою машиною Тьюрінга.

Приклад 5.3. Наведемо детерміновану лінійно обмежену машину Тьюрінга $\langle Q, X, T, \Lambda, \Delta, \{q_0\}, \{q_a\}\rangle$, яка допускає мову $\{a^n b^n c^n : n \in \mathbb{N}\}$. Нехай $Q = \{q_0, q_1, q_2, q_3, q_4, q_a\}$, $X = \{a, b, c, \Lambda\}$, $T = \{a, b, c\}$, множина Δ містить такі переходи:

$$((q_0, a), (q_1, *, r)), ((q_1, a), (q_1, a, r)), ((q_1, b), (q_2, *, r)), ((q_1, *), (q_1, *, r)),$$
$$((q_2, b), (q_2, b, r)), ((q_2, c), (q_3, *, l)), ((q_2, *), (q_2, *, r)), ((q_3, b), (q_3, b, l)),$$
$$((q_3, *), (q_3, *, l)), ((q_3, a), (q_1, *, r)), ((q_3, \Lambda), (q_4, \Lambda, r)), ((q_4, *), (q_4, \Lambda, r)),$$
$$((q_4, \Lambda), (q_a, \Lambda, l)).$$

У термінах неформального визначення ця машина зупиняється в заключному стані q_a з очищенням стрічки для вхідних слів $w \in L$. Зауважимо, що машини Тьюрінга з прикладів 2.4 та 2.5 також лінійно обмежені, але вони допускають порожнє слово, на відміну від цієї машини.

Теорема 5.2. *Для довільної контекстно-залежної формальної мови L існує (взагалі кажучи, недетермінована) лінійно обмежена машина Тьюрінга M, яка допускає мову L.*

Доведення. Нехай мова $L = L[G]$ породжена контекстно-залежною, а отже, невкорочувальною граматикою $G = \langle V, T, P, S \rangle$. Нехай «$*$» – деякий символ, який не міститься у множинах T та V. Для кожної продукції $p_j = (\alpha_1\alpha_2 \ldots \alpha_m \to \beta_1\beta_2 \ldots \beta_n) \in P$ $(n > m \geqslant 1)$ введемо переходи, які замінюють одне із входжень слова $\beta_1\beta_2 \ldots \beta_n$ на $\alpha_1\alpha_2 \ldots \alpha_m *^{n-m}$:

$$((q_0, \beta_1), (q_j^2, \alpha_1, r)),$$
$$((q_j^k, \beta_k), (q_j^{k+1}, \alpha_k, r)) \quad (2 \leqslant k \leqslant m),$$
$$((q_j^k, \beta_k), (q_j^{k+1}, *, r)) \quad (m+1 \leqslant k \leqslant n-1),$$
$$((q_j^n, \beta_n), (q_0, *, s));$$

у випадку $n = m \geqslant 2$ введемо переходи:

$$((q_0, \beta_1), (q_j^2, \alpha_1, r)),$$
$$((q_j^k, \beta_k), (q_j^{k+1}, \alpha_k, r)) \quad (2 \leqslant k \leqslant n-1),$$
$$((q_j^n, \beta_n), (q_0, \alpha_n, s));$$

у випадку $m = n = 1$ наведені переходи замінюються на один перехід $((q_0, \beta_1), (q_0, \alpha_1, s))$. Також введемо команди пересування курсора, які забезпечують «ігнорування» машиною символів «$*$»: $((q_j^k, *), (q_j^k, *, r))$ $(2 \leqslant k \leqslant n-1)$. Як і при доведенні теореми 5.1 зазначимо, що продукція $\varepsilon \to \varepsilon$ невкорочувальної граматики G не змінює мову, породжену G, а продукцію $\varepsilon \to \beta$ $(|\beta| \geqslant 1)$ можна замінити згідно із зауваженням 1.5.

Далі введемо команди пересування курсора, які забезпечують пошук деякого слова $\beta_1\beta_2 \ldots \beta_n$ для наступної його заміни на $\alpha_1\alpha_2 \ldots \alpha_m *^{n-m}$: $((q_0, \xi), (q_0, \xi, l))$, $((q_0, \xi), (q_0, \xi, r))$ $(\xi \in T \cup V \cup \{*\})$.

Нарешті, введемо команди очищення стрічки та переходу машини Тьюрінга у стан допуску q_a, якщо на стрічці наявні лише символи «$*$»

та один символ S (джерело граматики):

$$((q_0, S), (p_1, *, r)), \quad ((p_1, *), (p_1, *, r)), \quad ((p_1, \Lambda), (p_2, \Lambda, l)),$$
$$((p_2, *), (p_2, \Lambda, l)), \quad ((p_2, \Lambda), (q_a, \Lambda, r)).$$

Легко зрозуміти, що машина Тьюрінга з наведеною системою команд сприймає ті й тільки ті слова, які породжуються продукціями граматики G, оскільки для сприйняття слова має існувати хоча б одна послідовність команд, яка переводить машину Тьюрінга у стан допуску q_a та очищує стрічку. Ця машина лінійно обмежена та недетермінована. □

Теорема 5.3. *Нехай формальну мову L, що не містить порожнього слова, допускає деяка лінійно обмежена машина Тьюрінга M. Тоді існує контекстно-залежна формальна граматика, яка породжує мову L.*

Доведення. Нехай $M = \langle Q, X, T, \Lambda, \Delta, I, F\rangle$ має лише один заключний стан: $F = \{q_a\}$. В інших випадках вводимо новий стан $\tilde{q}_a \notin Q$ та розглядаємо машину

$$\langle Q \cup \{\tilde{q}_a\}, X, T, \Lambda, \Delta \cup \{((q, \alpha), (\tilde{q}_a, \alpha, s)) : q \in F, \alpha \in X\}, I, \{\tilde{q}_a\}\rangle.$$

Побудуємо невкорочувальну граматику $G = \langle V, T, P, S\rangle$. Візьмемо множину нетермінальних символів $V = \{A_{\alpha,q} : \alpha \in X, q \in Q\} \cup (X \setminus T)$, де символ $A_{\alpha,q}$ відповідає поточному символу α та поточному стану q машини. Нехай джерело $S = A_{\Lambda,q_a}$. Множина продукцій P містить:

1) продукції $A_{\Lambda,q_a} \to A_{\Lambda,q_a}\Lambda | \Lambda A_{\Lambda,q_a}$;

2) продукцію $\alpha_2 A_{\beta,q} \to A_{\alpha_1,p}\beta$ для кожного символа $\beta \in X$ та кожного переходу $((p, \alpha_1), (q, \alpha_2, r)) \in \Delta$;

3) продукцію $A_{\beta,q}\alpha_2 \to \beta A_{\alpha_1,p}$ для кожного символа $\beta \in X$ та кожного переходу $((p, \alpha_1), (q, \alpha_2, l)) \in \Delta$;

4) продукцію $A_{\alpha_2,q} \to A_{\alpha_1,p}$ для кожного переходу $((p, \alpha_1), (q, \alpha_2, s)) \in \Delta$;

5) продукції $A_{\alpha,q} \to \alpha$ для кожного символа $\alpha \in T$ та кожного стану $q \in I$.

Легко перевірити, що із джерела A_{Λ,q_a}, яке відповідає заключному стану q_a машини M, застосуванням наведених продукцій можна отримати слово $w \in T^+$ (вхідне слово на початку роботи машини) тоді й тільки тоді, коли машина допускає слово $w \in T^+$.

Нарешті, для граматики G за теоремою 5.1 існує еквівалентна КЗ-граматика. Зауважимо, що в загальному випадку граматику G можна спростити, прибравши ті символи $A_{\alpha,q}$ та продукції, які не використовуються. □

Приклад 5.4. Для лінійно обмеженої машини Тьюрінга

$$\langle \{q_0, q_1, q_a\}, \{a, b, \Lambda\}, \{a, b\}, \Lambda, \{((q_0, a), (q_1, \Lambda, r)),$$
$$((q_1, a), (q_1, \Lambda, r)), ((q_1, \Lambda), (q_a, \Lambda, l))\}, \{q_0\}, \{q_a\}\rangle,$$

яка допускає регулярну мову $L = \{a^n : n \in \mathbb{N}\}$, побудуємо невкорочувальну граматику, яка породжує мову L, використовуючи метод із доведення теореми 5.3. Множина нетермінальних символів

$$V = \{A_{\Lambda,q_0}, A_{\Lambda,q_1}, A_{\Lambda,q_a}, A_{a,q_0}, A_{a,q_1}, A_{a,q_a}, A_{b,q_0}, A_{b,q_1}, A_{b,q_a}, \Lambda\},$$

джерело $S = A_{\Lambda,q_a}$. Множина продукцій P містить:

1) продукції $A_{\Lambda,q_a} \to A_{\Lambda,q_a}\Lambda | \Lambda A_{\Lambda,q_a}$;

2) продукції $\Lambda A_{\Lambda,q_1} \to A_{a,q_0}\Lambda$, $\Lambda A_{a,q_1} \to A_{a,q_0}a$, $\Lambda A_{b,q_1} \to A_{a,q_0}b$ для переходу $((q_0, a), (q_1, \Lambda, r))$ та продукції $\Lambda A_{\Lambda,q_1} \to A_{a,q_1}\Lambda$, $\Lambda A_{a,q_1} \to A_{a,q_1}a$, $\Lambda A_{b,q_1} \to A_{a,q_1}b$ для переходу $((q_1, a), (q_1, \Lambda, r))$;

3) продукції $A_{\Lambda,q_a}\Lambda \to \Lambda A_{\Lambda,q_1}$, $A_{a,q_a}\Lambda \to aA_{\Lambda,q_1}$, $A_{b,q_a}\Lambda \to bA_{\Lambda,q_1}$ для переходу $((q_1, \Lambda), (q_a, \Lambda, l))$;

4) відповідних переходів немає;

5) продукції $A_{a,q_0} \to a$, $A_{b,q_0} \to b$.

Вправа 5.2. Переконатись, що граматика з прикладу 5.4 справді породжує мову L, побудувавши для цього схему виведення (див. приклад 1.13). Спростити наведену граматику, прибравши нетермінальні символи та продукції, які не використовуються.

Вправа 5.3. Побудувати лінійно обмежені машини Тьюрінга для формальних мов із прикладів 4.25, 4.26 (див. також вправу 5.1).

Вправа 5.4. Довести, що формальна мова $L = \{a^q : q$ – просте число$\}$ контекстно-залежна, побудувавши відповідну лінійно обмежену машину Тьюрінга M (див. також приклад 3.49 та зауваження 4.17).

Вказівка. Робочий алфавіт $\{|, \Lambda\}$ машини M відповідно до зауваження 5.5 розширити до робочого алфавіту $\{|, \tilde{|}, \hat{|}, \Lambda\}$. Машина M повинна виконувати такі дії:

1) покласти $k = 2$ та замінити перші k символів $|$ на символ $\hat{|}$;

2) перевірити, чи ділиться q на k, замінюючи наступні символи $|$ на символи $\tilde{|}$;

3) у разі, якщо q не ділиться на k, замінити $k+1$-й символ на символ $\hat{|}$, збільшити k на 1 та перейти до п. 2.

Вправа 5.5. Довести, що формальна мова $L = \{a^{m^2} : m \geqslant 1\}$ контекстно-залежна, побудувавши відповідну лінійно обмежену машину Тьюрінга (див. також приклад 3.50 та зауваження 4.17).

Запитання та завдання для самоконтролю

1. Навести означення контекстно-залежної формальної граматики та контекстно-залежної формальної мови.

2. Чи є довільна КВ-граматика контекстно-залежною?

3. Навести означення невкорочувальної граматики.

4. Навести схему побудови КЗ-граматики, яка еквівалентна невкорочувальній граматиці.

5. Навести означення нормальної форми Куроди.

6. Сформулювати властивості замкненості класу КЗ-мов, навести методи побудови відповідних граматик.

7. Чи утворює множина контекстно-залежних формальних мов алгебру множин?

8. Навести означення лінійно обмеженої машини Тьюрінга.

9. Навести схему побудови лінійно обмеженої машини Тьюрінга, яка допускає КЗ-мову.

10. Навести схему побудови КЗ-граматики, що породжує мову, яку допускає лінійно обмежена машина Тьюрінга.

Список літератури

1. Chomsky N. Three models for the description of language / N. Chomsky // IEEE Transactions on Information Theory. – 1956. – Vol. 2:3. – P. 113–124.
2. Chomsky N. Syntactic structures / N. Chomsky. – Berlin, New York : Motion and Co., The Hague, 1957. – 117 p.
3. Chomsky N. On Certain Formal Properties of Grammars / N. Chomsky // Information And Control. – 1959. – Vol. 2:2. – P. 137–167.
4. Ахо А. Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ / А. Ахо, Дж. Ульман. – М. : Мир, 1978. – 612 с.
5. Пентус А. Е. Контекстно-свободные языки. Сб. задач / А. Е. Пентус, М. Р. Пентус. – М. : РГГУ, 2016. – 75 с. – ISBN 978-5-7281-2012-4.
6. Пентус А. Е. Конечные автоматы и регулярные выражения. Сб. задач / А. Е. Пентус, М. Р. Пентус. – М. : Изд-во попечительского совета механ.-мат. ф-та МГУ, 2015. – 72 с. – 100 экз.
7. Кук Д. Компьютерная математика / Д. Кук, Г. Бейз. – М. : Наука, 1990. – 384 с. – 23 000 экз. – ISBN 5-02-014216-6.
8. Новиков Ф. Дискретная математика для программистов / Ф. Новиков. – СПб. : Питер, 2000. – 304 с. – 5000 экз. – ISBN 5-272-0183-4.
9. Пентус А. Е. Теория формальных языков / А. Е. Пентус, М. Р. Пентус. – М. : Изд-во Центра прикладн. исслед. при механ.-мат. ф-те МГУ, 2004. – 80 с. – 100 экз.
10. Гросс М. Теория формальных грамматик / М. Гросс, А. Лантен. – М. : Мир, 1971. – 294 с.
11. Рейуорд-Смит В. Дж. Теория формальных языков. Вводный курс / В. Дж. Рейуорд-Смит. – М. : Радио и связь, 1988. – 128 с. – 30 000 экз.

12. Ющенко К. Л. Алгоритмічні алгебри / К. Л. Ющенко, С. В. Суржко, Г. О. Цейтлін, А. І. Шевченко . – Київ : ІЗМН, 1997. – 480 с. – 500 екз. – ISBN 5-7763-9094-X.
13. Hopcroft John E. Formal languages and their relation to automata / John E. Hopcroft, Jeffrey D. Ullman. – Reading, Massachusetts : Addison-Wesley Publishing Company, 1969. – 242 p.
14. Хопкрофт Джон Э. Введение в теорию автоматов, языков и вычислений / Джон Э. Хопкрофт, Раджив Мотвани, Джеффри Д. Ульман. – М. : Изд. дом «Вильямс», 2002. – 528 с. – 3500 экз. – ISBN 5-8459-0261-4.
15. Лихтарников Л. М. Математическая логика. Курс лекций. Задачник-практикум и решения / Л. М. Лихтарников, Т. Г. Сукачева. – СПб. : Изд-во «Лань», 1999. – 288 с. – 3000 экз. – ISBN 5-8114-0082-9.
16. Нікітченко М. С. Математична логіка та теорія алгоритмів / М. С. Нікітченко, С. С. Шкільняк. – Київ : ВПЦ Київ. ун-т, 2008. – 528 с. – ISBN 966-439-007-0.
17. Шкільняк С. С. Теорія алгоритмів. Приклади й задачі : навч. посіб. / С. С. Шкільняк. – Київ : ВПЦ Київ. ун-т, 2012. – 151 с.
18. Нікітченко М. С. Теорія алгоритмів / М. С. Нікітченко, О. С. Шкільняк, С. С. Шкільняк. – Київ : ВПЦ Київ. ун-т, 2015. – 239 с.
19. Теорія цифрових автоматів та формальних мов. Вступний курс : навч. посіб. / С. Ю. Гавриленко, А. М. Клименко, Н. Ю. Любченко та ін. – Харків : НТУ «ХПІ», 2011. – 176 с.
20. Brzozowski J. A. Canonical regular expressions and minimal state graphs for definite events / J.A. Brzozowski // Mathematical theory of automata, MRI Symposia Series, Polytechnic Institute of Brooklyn. – 1962. – Vol. 12. – P. 529–561.
21. Шишков Д. Б. Минимизация конечных автоматов / Д. Б. Шишков // Kybernetika. – 1972. – Т. 8. – Вып. 4. – С. 297–316.
22. Яглом И. М. Булева структура и ее модели / И. М. Яглом. – М. : Сов. радио, 1980. – 192 с. – 25 000 экз.
23. Гинзбург С. Математическая теория контекстно-свободных языков / С. Гинзбург. – М. : Мир, 1970. – 326 с.
24. Сопронюк Т. М. Системне програмування. У 2-х ч. Ч. І. Елементи теорії формальних мов : навч. посіб. / Т. М. Сопронюк. – Чернівці : ЧНУ, 2008. – 84 с.

25. Пентус А. Е. Математическая теория формальных языков : учеб. пособие / А. Е. Пентус, М. Р. Пентус. – М. : Интернет-ун-т информацион. технологий; БИНОМ. Лаборатор. знаний, 2006. – 247 с. – ISBN 5-9556-0062-0.
26. Саломаа А. Жемчужины теории формальных языков / А. Саломаа. – М. : Мир, 1986. – 160 с.
27. Захарія Л. М. Формальні мови, граматики та автомати / Л. М. Захарія, М. М. Заяць. – Львів : Вид-во Львівськ. політехн., 2016. – 196 с. – 200 екз.
28. Revesz G. E. Introduction to formal languages / G. E. Revesz. – Dover Publications, Inc. : Mineola, New York, 2015. – 208 p.
29. Мартыненко Б. К. Языки и трансляции : учеб. пособие / Б. К. Мартыненко. – СПб. : Изд-во С.-Петербург. ун-та, 2004. – 231 с. – ISBN 5-288-02870-2.

Покажчик термінів