Міністерство освіти та науки України
Національний технічний університет України
«Київський політехнічний інститут»
Навчально-науковий комплекс
«Інститут прикладного системного аналізу»

Формальні мови та автомати

Для студентів математичних спеціальностей університетів

Київ 2010

Зміст

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 $\alpha =$ фiIґфф – 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 $\alpha \cdot \beta = abbba$, $\beta \cdot \alpha = baabb$.

Наведений приклад 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, [1]).

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

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

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

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:

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

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:

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

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. $\{ab, c\}^* = \{(ab)^{n_1} c^{m_1} \cdots (ab)^{n_k} c^{m_k} : n_1, \ldots, n_k, m_1, \ldots, m_k \geqslant 0,\ k \geqslant 0\}$.

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

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 $(\alpha, \beta) \in (A^* \times A^*)$ і 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 1.8. 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 1.9. 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 призводить до видалення із слова γ одного із входжень підслова α.

Приклад 1.10. 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 1.11. 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.

Для запису кількох продукцій з однаковими лівими частинами, тобто для запису n продукцій вигляду $\alpha \to \beta_1$, $\alpha \to \beta_2$, …, $\alpha \to \beta_n$, часто використовують позначення $\alpha \to \beta_1 | \beta_2 | \dots | \beta_n$.

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^* складається із таких таких пар $(a_0, a_n) \in (X \times X)$, що

$$a_0 R a_1, a_1 R a_2, \dots, a_{n-1} R a_n$$

для деяких $a_1, \dots, a_{n-1} \in X$, $n \geqslant 0$. 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) із α скінченною кількістю застосувань продукцій із множини P. Конкретну послідовність слів $\alpha_1, \dots, \alpha_n$, 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 $\beta = \alpha$ виведено із α за 0 застосувань продукцій із P.

Детальніше про різні замикання бінарного відношення див. [1,2].

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 із джерела скінченною кількістю застосувань продукцій із множини P:

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

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

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

Приклад 1.12. 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 наведено схему виведення; поруч із символами виведення «$\Rightarrow$» вказано продукції, застосовані на конкретному кроці.

$$\begin{array}{cccccccc} S & \overset{p_1}{\Rightarrow} & abS & \overset{p_1}{\Rightarrow} & (ab)^2S & \overset{p_1}{\Rightarrow} \cdots \overset{p_1}{\Rightarrow} & (ab)^nS & \overset{p_1}{\Rightarrow} \cdots \\ \Downarrow p_2 & & \Downarrow p_2 & & \Downarrow p_2 & & \Downarrow p_2 & \\ \varepsilon & & ab & & (ab)^2 & & (ab)^n & \end{array}$$

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:

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

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

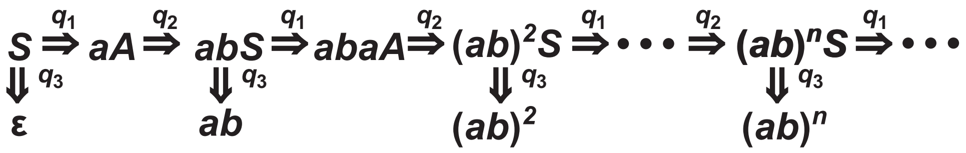

Рис. 1.2

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:

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

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

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 $L[G_3]$:

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

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

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

вважаючи, що обидві мови $L[G_1]$ та $L[G_3]$ задані над спільним алфавітом $\{a, b, c\}$ (див. заув. 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 $\{(ab)^n a^m c b^m : n, m \geqslant 0\}$. Спочатку зазначимо, що

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

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

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

Зазначимо, що $\{a^{n_1} c b^{n_1} \cdots a^{n_k} c b^{n_k} : n_1, \ldots, n_k \geqslant 0,\ k \geqslant 0\} = (L[G_3])^*$. 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. В 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 1.12.

1.4. Ієрархія Хомського

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 1.12). 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 питання, яка з еквівалентних граматик краща – все залежить від конкретної практичної задачі. У наш час найбільш поширеною класифікацією формальних граматик є так звана *ієрархія Хомського*[1], яка базується 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, [3–8]). 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 $\gamma_1 A \gamma_2 \to \gamma_1 \alpha \gamma_2$, де $A \in V$; $\alpha \in (V \cup T)^+$; $\gamma_1, \gamma_2 \in (V \cup T)^*$. 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 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, [8]).

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 про-

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

дукції 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. [3]). 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 $A \to a_1B_1$, $B_1 \to a_2B_2$, …, $B_{n-2} \to a_{n-1}B_{n-1}$, $B_{n-1} \to a_nB$ чи $B_{n-1} \to a_n$ (замість $B_{n-1} \to a_nB$), вводячи нові нетермінальні символи B_1, B_2, …, 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 1.13. 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 $\langle\{S, A\}, \{a, b\}, \{S \to aS|aA|\varepsilon, A \to bA|\varepsilon\}, S\rangle$ регулярна.

Приклад 1.14. 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 даною граматикою, є контекстно-вільною.

Вправа 1.2. Для граматики з прикладу 1.14 вказати мову, яку вона породжує, та еквівалентну контекстно-вільну граматику.

У прикладі 1.12 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 1.15. 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 1.12); однак ця мова, згідно з означенням 1.7, є регулярною, оскільки існує регулярна граматика, яка її породжує.

Приклад 1.16. Побудуємо регулярну формальну граматику, яка породжує мову $\{a^{2n}b^{3k+1} : n \geqslant 0, k \geqslant 0\}$. З 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 1.17. 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 1.18. Формальна мова $\{a^nb^nc^n : n \geqslant 1\}$ не тільки нерегулярна, а навіть не контекстно-вільна, що буде доведено у підрозд. **??**. Однак ця мова контекстно-залежна; це буде доведено у підрозд. 4.1 (приклад 4.1).

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

Вказівка. Використати прийом, застосований у зауваженні 1.7.

Розділ 2

Машини Тьюрінга і формальні граматики

2.1. Визначення машини Тьюрінга

2.1.1. Неформальний опис

Машина Тьюрінга[1] – абстрактний пристрій, який можна представити 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 символу, 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).

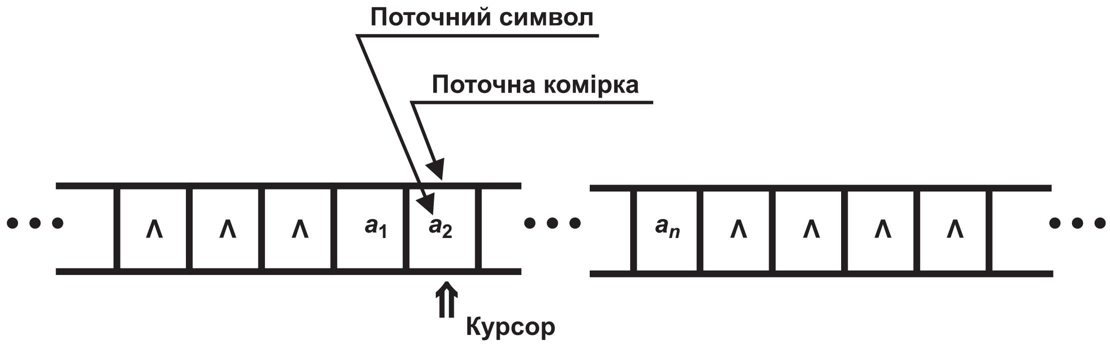

Рис. 2.1

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 знаходиться 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*. Кожен 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 $((\widetilde{q}_1, \widetilde{\alpha}_1), (\widetilde{q}_2, \widetilde{\alpha}_2, \widetilde{d}))$ відрізняються 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 відсутній перехід, у 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 символу 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 $T = \{a, b, c\}$ та множиною станів $Q = \{q_0, q_1\}$, серед яких один початковий стан q_0 і 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. заув. 2.1), вказує на перший зліва символ вхідного слова.

Якщо вхідне слово починається з 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 буде виконувати команду $((q_0, a), (q_0, a, r))$ доти, доки курсор не «проїде» всі символи a на початку слова. Далі можливі три різні випадки, які зараз 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 $((q_0, b), (q_1, c, s))$: в поточну комірку буде записано c, курсор залишиться на місці і машина зупиниться, перейшовши у заключний стан q_1. На рис. 2.2 показано роботу цієї машини на вхідному слові $aabac$ (тут і далі біля курсору 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 $aacab$.

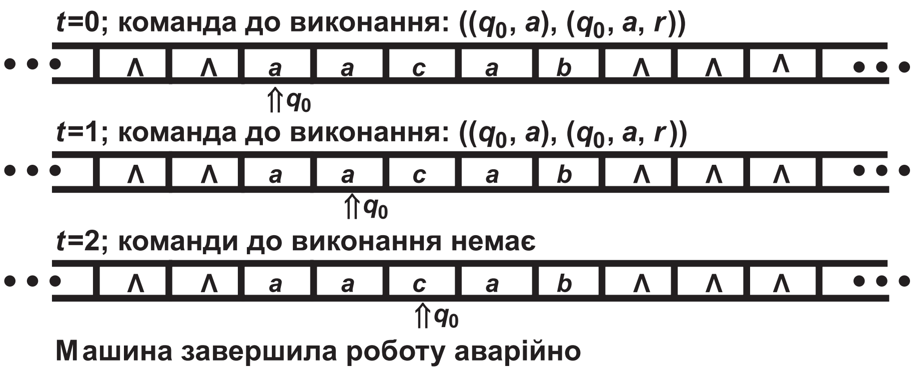

Рис. 2.3

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 призведе 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.

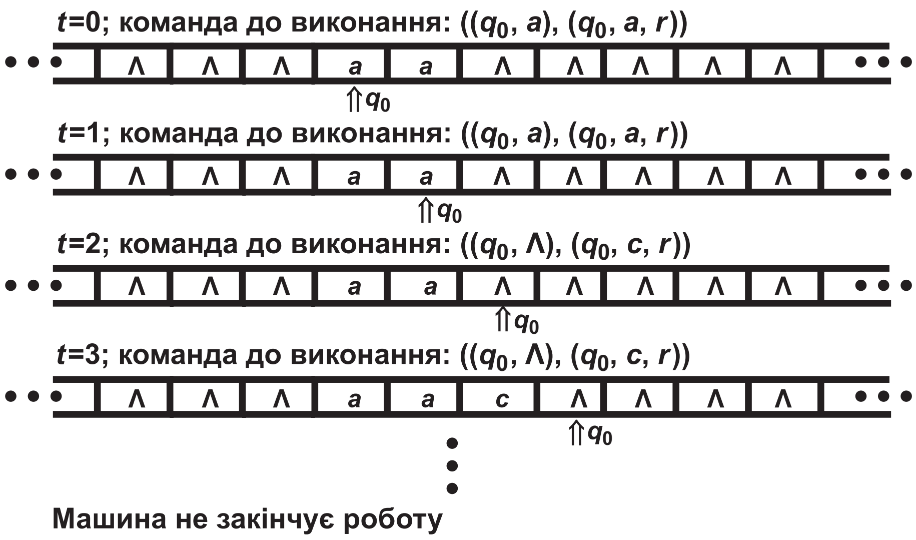

Рис. 2.4

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-якою 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 $(v, q_1, \alpha_1, \varepsilon) \vdash_M (v\alpha_2, q_2, \Lambda, \varepsilon)$ для $u \in X^*$, $v \in X^*$, $\beta \in X$.

Відношення «$\vdash^*_M$» – 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 $c \vdash^*_M c$ для будь-якої конфігурації c, оскільки відношення «$\vdash^*_M$» 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 символу:

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

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 $(\varepsilon, q_0, w) \vdash^*_M (\Lambda^k, q, \widetilde{w}\Lambda^m)$, то говорять, 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 «$\vdash^*_M$» писатимемо відповідно «$\vdash$» та «$\vdash^*$».

Приклад 2.2. Опишемо роботу машини Тьюрінга з прикладу 2.1

для вхідних слів $aabac$, $aacab$ та aa через відношення «$\vdash$»:

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

На слові $aabac$ машина закінчує роботу нормально, перейшовши у заключний стан q_1; на слові $aacab$ машина закінчує роботу аварійно, за відсутності 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. Наведемо 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 (номера 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):

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

$$\overset{17}{\vdash}(aa, q_0, b, AA)\overset{2}{\vdash}(aa\widetilde{b}, q_b, A, A)\overset{9}{\vdash}(aa\widetilde{b}A, q_b, A, \varepsilon)\overset{9}{\vdash}(aa\widetilde{b}AA, q_b, \Lambda, \varepsilon)\overset{12}{\vdash}$$
$$\overset{12}{\vdash}(aa\widetilde{b}A, q_1, A, B)\overset{13}{\vdash}(aa\widetilde{b}, q_1, A, AB)\overset{13}{\vdash}(aa, q_1, \widetilde{b}, AAB)\overset{18}{\vdash}(aab, q_0, A, AB)\overset{19}{\vdash}$$
$$\overset{19}{\vdash}(aaba, q_2, A, B)\overset{21}{\vdash}(aabaa, q_2, B, \varepsilon)\overset{22}{\vdash}(aabaab, q_2, \Lambda, \varepsilon)\overset{23}{\vdash}(aabaa, q_3, b, \Lambda)\overset{25}{\vdash}$$
$$\overset{25}{\vdash}(aaba, q_3, a, b\Lambda)\overset{24}{\vdash}(aab, q_3, a, ab\Lambda)\overset{24}{\vdash}(aa, q_3, b, aab\Lambda)\overset{25}{\vdash}(a, q_3, a, baab\Lambda)\overset{24}{\vdash}$$
$$\overset{24}{\vdash}(\varepsilon, q_3, a, abaab\Lambda)\overset{24}{\vdash}(\varepsilon, q_3, \Lambda, aabaab\Lambda)\overset{26}{\vdash}(\Lambda, q_4, a, abaab\Lambda),$$

і машина завершує роботу переходом у заключний стан q_4.

Отже, $(\varepsilon, q_0, aab) \overset{*}{\models} (\Lambda, q_4, aabaab\Lambda)$. Легко зрозуміти, що для довільного $w \in \{a,b\}^+$ отримуємо: $(\varepsilon, q_0, w) \overset{*}{\models} (\Lambda, q_4, ww\Lambda)$, тобто машина копіює непорожнє вхідне слово w, перетворюючи його у слово ww. 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 $L \subset T^*$ із станом допуску q_a, якщо:

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

2) для кожного слова $w \notin L$ знайдуться такі $k \geqslant 0$, $m \geqslant 0$, що $(\varepsilon, q_0, w) \overset{*}{\models}_M (\Lambda^k, q_r, \Lambda^m)$. 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 $L \subset T^*$ із станом допуску q_a, то ця ж машина розв'язує мову $\overline{L}$ із станом допуску q_r.

Приклад 2.4. Наведемо 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 $X = \{a, b, c, *, \Lambda\}$, $T = \{a, b, c\}$, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_a, q_r\}$, $I = \{q_0\}$, $F = \{q_a, q_r\}$ із 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 $L=\{a^nb^nc^n : n \geqslant 0\}$. Дійсно, $(\varepsilon,q_0,a^nb^nc^n)\vdash^{*}(\Lambda^{n+1},q_a,\Lambda)$ для $n\in\mathbb{N}$, та $(\varepsilon,q_0,\Lambda)\vdash(\varepsilon,q_a,\Lambda)$; якщо $w\notin L$, то $(\varepsilon,q_0,w)\vdash^{*}(\Lambda^{n+1},q_r,\Lambda)$, де $n=|w|$.

Зазначимо, що ця ж машина розв'язує мову $\{a,b,c\}^*\setminus\{a^nb^nc^n : n\geqslant 0\}$ із 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 $(\varepsilon,q_0,w)\vdash^{*}_{M}(\Lambda^k,q,\Lambda^m)$ 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 переховною.

Зауваження 2.3. У термінах неформального опису, якщо машина не допускає задане вхідне слово, 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 сприймає будь-яке слово $w\in T^*$, тобто сприймає 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 $L\subset T^*$ із станом допуску q_a. Тоді машина Тьюрінга

$\langle Q, X, T, \Lambda, \Delta, \{q_0\}, \{q_a\}\rangle$ сприймає 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 сприймає 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 $\langle \widetilde{Q}, X, T, \Lambda, \widetilde{\Delta}, \{q_0\}, \{q_a\}\rangle$ сприймає 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 $a^2 b^2 c^3 \notin L$ та порівняти з результатом вправи 2.2.

Приклад 2.6. Наведемо машину Тьюрінга $\langle Q, X, \{a,b\}, \Lambda, \Delta, I, F\rangle$, яка сприймає 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 сприймає ті й тільки ті слова над алфавітом $\{a, b\}$, які можна зобразити у формі $\alpha_1\alpha_2\cdots\alpha_n$, 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 сприймає мову $\{a^2, ab\}^*$.

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

Схема доведення. Нехай мову $L \subset T^*$ сприймає деяка, можливо недетермінована, машина $M = \langle Q, X, T, \Lambda, \Delta, I, F\rangle$, $I = \{q_0^1, q_0^2, \ldots, q_0^n\}$. Без втрати 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 сприймає слово $w \in T^*$ тоді і тільки тоді, коли це слово сприймає машина M.

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

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

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 знаходиться 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 машини $\widetilde{M}$ 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 сприймає слово $w \in T^*$, машина $\widetilde{M}$ врешті-решт знайде конфігурацію $(\Lambda^k, q, \Lambda^n)$ з деякими $q \in F$, $u = \Lambda^k$, $v = \Lambda^m$, $k \geqslant 0$, $m \geqslant 0$, і завершить роботу з очищенням своєї стрічки. Якщо машина M не сприймає слово w, машина $\widetilde{M}$ завершить роботу аварійно. Таким чином, $\widetilde{M}$ сприймає слово $w \in T^*$ тоді і тільки тоді, коли слово w сприймає 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., напр., [4]).

2.3. Напіврозв'язні мови і формальні граматики

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

Нижченаведена проста лема є допоміжною, 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) $\forall \gamma_1, \gamma_2 \in A^*\ \exists k, m \geqslant 0\colon (\gamma_1, q_0, u\gamma_2) \models^{*} (\Lambda^k\gamma_1, q_0, v\gamma_2\Lambda^m)$;

2) $\forall \gamma_1 \in A^*\ \exists k, m \geqslant 0\colon (\gamma_1, q_0, w) \models^{*} (\Lambda^k\gamma_1, q_0, w\Lambda^m)$, 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 сприймає 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, 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 $S \underset{G}{\overset{*}{\Rightarrow}} w$ тоді і 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 сприймає ті й тільки 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 сприймається 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

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

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

Розділ 3

Скінченні автомати та регулярні граматики

3.1. Скінченні автомати: основні поняття

3.1.1. Неформальний опис

Скінченний автомат можна представити 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 символу, 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 автомату; ця множина є скінченною і 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 автомату в момент $t = 0$ вказує на перший зліва символ вхідного слова (рис. 3.1).

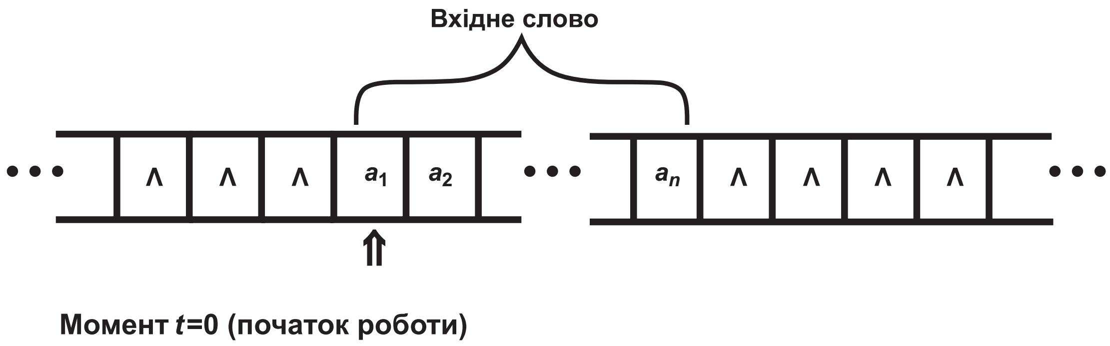

Рис. 3.1

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

Дії скінченного автомату 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 автомату 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 символу немає відповідного переходу.

Кажуть, що скінченний автомат допускає слово $w = a_1 a_2 \ldots a_n$ ($a_i \in T$, $1 \leqslant i \leqslant n$), якщо існує послідовність станів $q_0 \in I$, $q_1 \in Q$, $q_2 \in Q$, $\ldots$, $q_{n-1} \in Q$, $q_n \in F$, 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 з 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 символу b, буде виконаний перехід (q_0, b, q_2), скінченний автомат перейде в стан q_2 і далі, зчитуючи як a, так і b (переходи (q_2, a, q_2) та (q_2, b, q_2) відповідно), залишається в стані $q_2 \notin F$. Якщо вхідне слово починається з символу a, буде виконаний перехід (q_0, a, q_1), автомат перейде в 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 в 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 $aaab$.

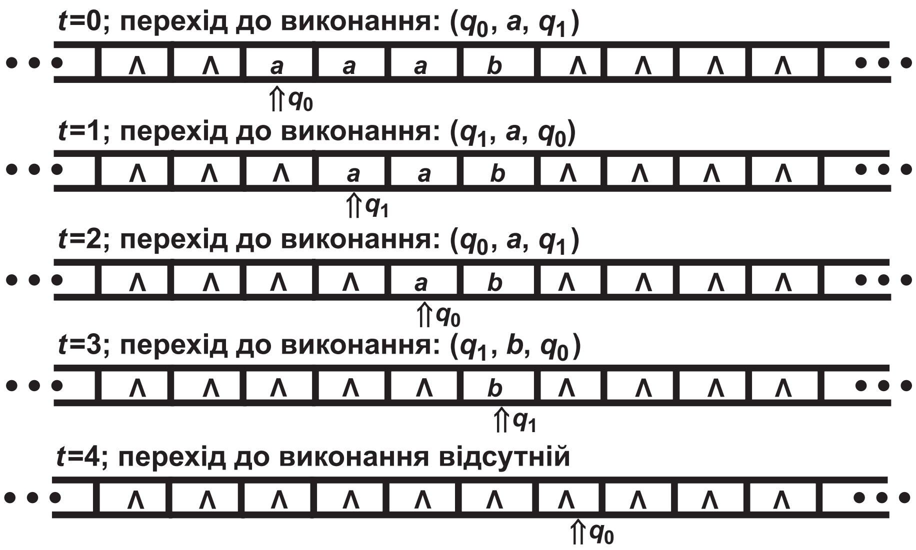

Рис. 3.2

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. [5, 9]).

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 автомату до детермінованого можна наступним 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 автомату $\langle Q, T, \Delta, I, F\rangle$ називають довільний набір $(q, w) \in (Q \times T^*)$.

Зауваження 3.4. Порожній символ Λ у неформальному описі скінченного автомату використовується лише для опису закінчення роботи, коли вхідне слово прочитане повністю, і курсор вказує саме на Λ. Для формального опису скінченного автомату, що не передбачає введення стрічки та курсору, порожній символ не потрібен.

На множині конфігурацій скінченного автомату $M = \langle Q, T, \Delta, I, F \rangle$ 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* із 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 автомату 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 і тільки тоді, коли $\forall\, \tilde{q} \in Q : (q, a, \tilde{q}) \notin \Delta$. 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

$$c_0 \vdash_M c_1 \vdash_M c_2 \ldots \vdash_M c_n$$

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 $(q, \varepsilon) \vdash^*_M (q, \varepsilon)$. Із 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 із 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 автомату 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 $(q_0, aaab) \mathrel{\vdash^*_M} (q_0, \varepsilon)$.

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 $\{a, b\}$, з 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, $(q_0, aaab) \vdash^* (q_2, aab)$, $(q_0, aaab) \vdash^* (q_1, b)$ та $(q_0, aaab) \vdash^* (q_0, \varepsilon)$. Отже, автомат допускає слово $aaab$, оскільки $(q_0, aaab) \vdash^* (q_0, \varepsilon)$, 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 дійсно еквівалентний автомату з прикладу 3.1.

3.1.3. Задання скінченного автомату за допомогою графу

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

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

2) кожне ребро графу 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 і тільки тоді, коли $(q_1, a, q_2) \in \Delta$;

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

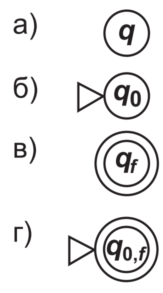

Рис. 3.3

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

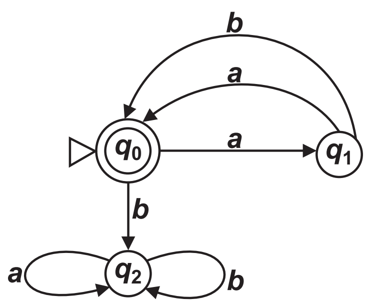

Рис. 3.4

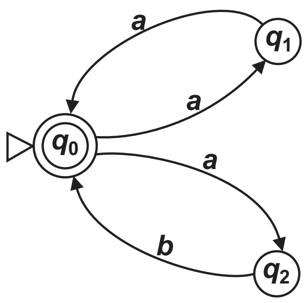

Рис. 3.5

Приклад 3.5. На рис. 3.6 наведено граф скінченного автомату, що сприймає таку формальну мову:

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

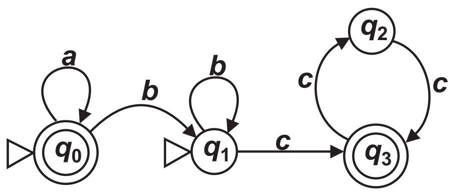

Рис. 3.6

Наведений 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 наведений граф зображує скінченний автомат $\langle\{a, b, c\}, Q, \Delta, I, F\rangle$ 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 $M = \langle T, Q, \Delta, I, F\rangle$ – 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, дозволяють однозначно визначити функцію переходів δ через відношення переходів Δ, і навпаки – відношення Δ через функцію δ. Також зазначимо, що для недетермінованих автоматів розглядають [5–7](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 автомату з 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 табл. 3.1 із значеннями функції переходів детермінованого автомату M називають *таблицею переходів* автомату 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* автомату 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 розпізнає детермінований скінченний автомат $M = \langle T, Q, \Delta, \{q_0\}, F \rangle$, 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 $q_2 = \delta^*(q_1, w)$ автомату M після зчитування слова $w \in T^*$, якщо на початку зчитування слова w автомат знаходився у стані q_1.

Приклад 3.7. Опишемо розширену функцію переходів для автомату з 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 і 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 автомату достатньо трьох станів q_0, q_1 та q_2; $(q_0, w) \overset{*}{\vdash} (q_i, \varepsilon)$ тоді і тільки тоді, коли w містить $3k+i$ входжень a $(i = 0, 1, 2)$. Відповідний автомат наведено на 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 – $\delta^*(q_i, w) = q_{(i+|w|_a) \bmod 3}$.

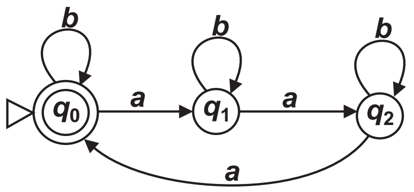

Рис. 3.7

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 автомату достатньо шести станів q_{ij} $(i = 0, 1, 2,\ j = 0, 1)$: $(q_{00}, w) \overset{*}{\vdash} (q_{ij}, \varepsilon)$ тоді і тільки тоді, коли w містить $3k+i$ входжень a та $2m+j$ входжень b. Відповідний автомат наведено на рис 3.8. Розширена функція переходів $\delta^*(q_{ij}, w) = q_{((i+|w|_a) \bmod 3)\ ((j+|w|_b) \bmod 2)}$.

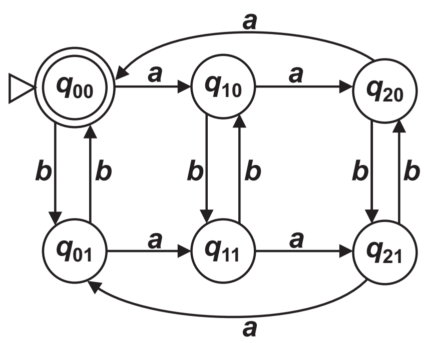

Рис. 3.8

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 автомату 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 сприймає $w \in T^*$ тоді й тільки тоді, коли w сприймає вихідний автомат M:

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

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 автомату $M_0 = \langle 2^Q, T, \Delta_0, I_0, F_0\rangle$, еквівалентного заданому довільному (можливо недетермінованому) скінченному автомату $M = \langle Q, T, \Delta, I, F\rangle$.

Приклад 3.10. Для недетермінованого скінченного автомату

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

(див. приклад 3.3) побудуємо відповідний детермінований автомат за методом із доведення теореми 3.1 (див. таблицю переходів 3.2). Дійсно, 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 жоден із 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 жоден із станів недосяжний: $\delta_0(\{q_1\}, a) = \{q_0\}$, $\delta_0(\{q_1\}, b) = \varnothing$. Із стану q_2 символом a жоден із 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 таблиці 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 із початкового стану I_0: стани $\{q_1, q_2\}$, $\varnothing$ і, власне, I_0 (див. таблицю 3.2). Вилучивши із автомату M_0 ті 5 станів, які недосяжні з початкового, отримуємо детермінований скінченний автомат, граф якого зображений на рис. 3.9. Зазначимо, що побудований автомат збігається з автоматом із прикладу 3.1 з точністю до перейменування станів.

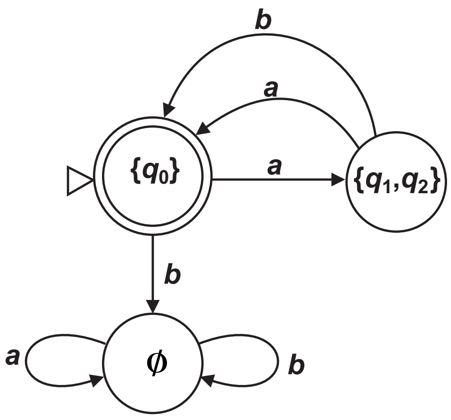

Рис. 3.9

Зазначимо, що з переходом до детермінованого скінченного автомату 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 із станів автомату 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 мову $L[M_0] = L[M]$ (детальніше див. підрозд. 3.6.2).

Отже, для побудови таблиці переходів автомату M_0 доцільно обмежуватись станами, досяжними з початкового, використовуючи, наприклад, механізм черги: на початку роботи в чергу записується 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 автомату, 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, із яких 5 досяжні з початкового стану $\{q_0, q_1\}$ (див. таблицю 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	$\varnothing$	$\{q_1\}$	$\varnothing$	$\varnothing$	$\varnothing$

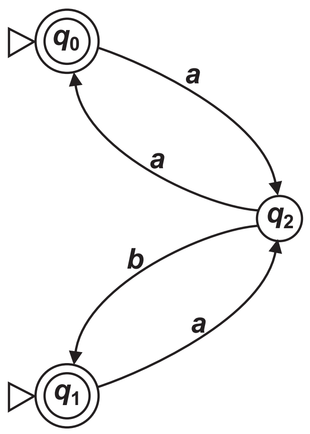

Рис. 3.10

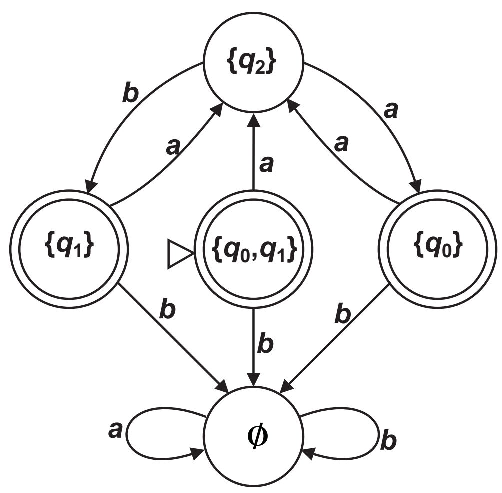

Рис. 3.11

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

Вправа 3.5. Для недетермінованого скінченного автомату, 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

У цьому підрозділі доведемо, що класи регулярних і автоматних мов збігаються.

Теорема 3.2. *Нехай G – регулярна граматика. Тоді існує скінченний автомат M, який розпізнає 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 $S \underset{G}{\overset{*}{\Rightarrow}} w$, 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) $F = \{q_f, S\}$.

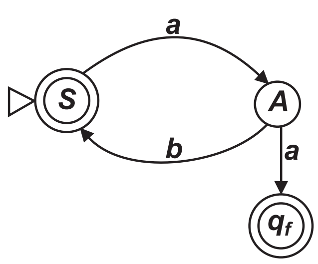

Рис. 3.12

Граф побудованого скінченного автомату 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 автомату 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) $F = \{S\}$.

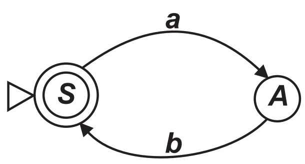

Рис. 3.13

Граф побудованого скінченного автомату 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 розпізнає мову $L[G]$ для граматики G з прикладу 1.16, 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 і тільки тоді, коли $S \underset{G}{\overset{*}{\Rightarrow}} w$, тобто $L[G] = L[M]$.

Теорему доведено. □

Приклад 3.14. Для скінченного автомату $M = \langle Q, T, \Delta, I, F\rangle$, 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 автомату M, граф якого наведено 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 автомату, 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 автомату з ε-переходами.

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

$$c_0 \vdash_M c_1 \vdash_M c_2 \ldots \vdash_M c_n$$

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 графу 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 наведено граф скінченного автомату з ε-переходами, що сприймає формальну мову $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}$.

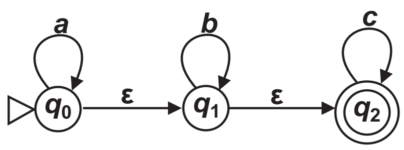

Рис. 3.14

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 наведено граф скінченного автомату над алфавітом $T = \{0, 1, +, -\}$, що сприймає формальну мову L, яка складається із двійкових записів цілих чисел:

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

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).

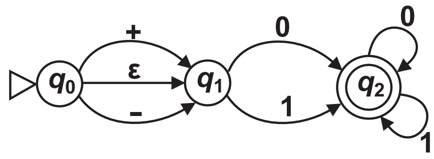

Рис. 3.15

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 автомату $M' = \langle Q', T, \Delta', I', F' \rangle$ 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 автомату визначимо як «найширшу» з 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 автомату 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 автомату 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 сприймає автомат $\langle Q', T, \Delta_n, I', F' \rangle$. □

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-переходів):

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

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 автомату без ε-переходів. Побудований автомат (див. рис. 3.16) еквівалентний вихідному, тобто обидва сприймають ту ж саму мову $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}$.

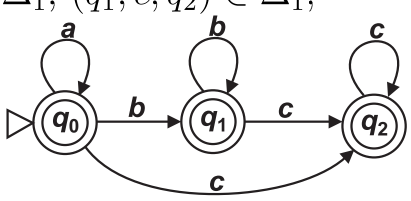

Рис. 3.16

Приклад 3.18. Позбудемося ε-переходів в автоматі з 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 сприймає ту ж саму формальну мову

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

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

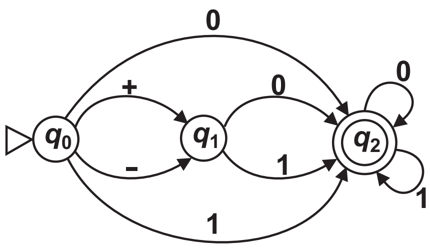

Рис. 3.17

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 $m = \min\{i \geqslant 0 \colon I_i = I_{i+1}\}$. 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 заув. 3.7. Видалення 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 (див. рис. 3.18) еквівалентний вихідному, тобто сприймає ту ж саму формальну мову $\{sw \colon w \in \{0,1\}^+, s \in \{+, -, \varepsilon\}\}$.

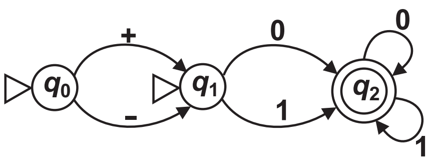

Рис. 3.18

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

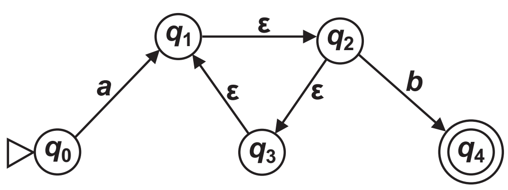

Рис. 3.19

3.5. Мінімізація детермінованих скінченних автоматів: теорема Майхілла–Нерода

У цьому підрозділі розглянемо проблему існування мінімального за кількістю станів детермінованого скінченного автомату, який сприймає 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) $C_L^{(r)}(a^n) = \{a^i b^{i+n} \colon i \geqslant 0\}$ 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 даному 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:

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

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

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

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:

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

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 автомату

Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат, який розпізнає регулярну мову L. Слова $w_1, w_2 \in T^*$ називають *еквівалентними відносно автомату* M, якщо $\delta^*(q_0, w_1) = \delta^*(q_0, w_2)$, де $\delta^* : (Q \times T^*) \to Q$ – розширена функція переходів автомату M (див. підрозд. 3.2.1). Еквівалентність слів $w_1, w_2 \in T^*$ відносно скінченного автомату 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, розпізнає мову $L = \{a^n b^m : n \geqslant 0,\ m \geqslant 0\}$ над алфавітом $T = \{a, b\}$. Для розширеної функції переходів отримуємо:

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

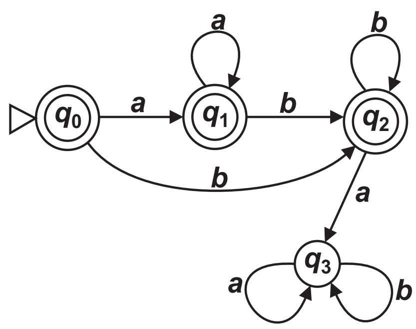

Рис. 3.20

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

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

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 автомату 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 розпізнає автомат M, зображений на рис. 3.20 (див. приклади 3.20, 3.25), отримуємо відповідність

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

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

Нехай L – регулярна мова над алфавітом T, що розпізнається 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 розпізнається 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 $\bigcup\limits_{u \underset{L}{\sim} w} [u]_M \subset [w]_L$. З іншого боку, 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 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 $[v_1]_L \cap [v_2]_L = \varnothing$ при $v_1 \underset{L}{\not\sim} v_2$ (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 розпізнає 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 розпізнає 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.

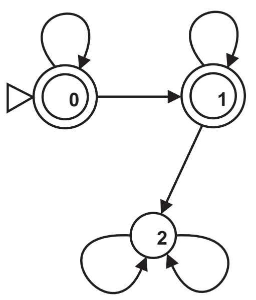

Рис. 3.21

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 розпізнає 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. заув. 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 автомату 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 розпізнає 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 розпізнає 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

[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.

$([w]_L, a, [u]_L) \in \Delta_L$: 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 $w_1 \underset{L}{\sim} w$, 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 автомату має вигляд $\delta_L([w]_L, a) = [wa]_L$, де $w \in T^*$, $a \in T$.

Розширена функція переходів $\delta_L^* : (Q_L \times T^*) \to Q_L$ для побудованого автомату має вигляд $\delta_L^*([w]_L, u) = [wu]_L$, де $w \in T^*$, $u \in T^*$. Дійсно, 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 дійсно розпізнає мову L. За еквівалентністю (3.4), автомат M_L сприймає слово $u \in T^*$ тоді і 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 сприймає u тоді і 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 розпізнає задану мову L. Теорему доведено. □

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

Теорема 3.5 не тільки стверджує, що будь-яку регулярну мову L розпізнає деякий мінімальний детермінований скінченний автомат M, але й надає конкретний спосіб побудови автомату 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 $T = \{a, b\}$ (див. приклад 3.24) отримуємо такий детермінований автомат, мінімальний за кількістю станів (див. рис. 3.22):

$$Q_L = \{a,b\}^*/_{\underset{L}{\sim}} = \{[\varepsilon]_L, [b]_L, [ba]_L\};$$

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

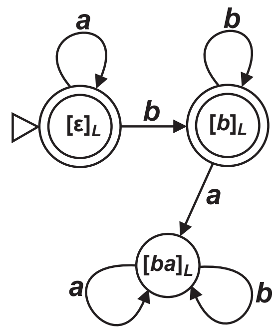

Рис. 3.22

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 автомату (3.6) (мінімального детермінованого автомату, що розпізнає L) необхідною є лише умова $i_L < \infty$.

Таким чином, будь-яку формальну мову L із скінченним індексом i_L розпізнає скінченний (навіть мінімальний) детермінований автомат, а отже мова L із 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. В літературі саме теорему 3.6 іноді називають теоремою Майхілла – Нерода (див., наприклад, [7]).

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 автомату

3.6.1. Еквівалентність станів детермінованого автомату

Нехай $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ – детермінований скінченний автомат, $\delta^* : (Q \times T^*) \to Q$ – розширена функція переходів автомату 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 $\delta^*(q_2, w)$ рівно один є допускаючим. Стани $q_1, q_2 \in Q$ називають *еквівалентними*, якщо ці стани не розрізняє жодне слово $w \in T^*$:

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

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 $M=\langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$ з 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 автомату $M = \langle Q, \{a, b, c\}, \Delta, \{q_0\}, F\rangle$, зображеного на рис. 3.23, розбиває $Q = \{q_0, q_1, q_2\}$ на два класи:

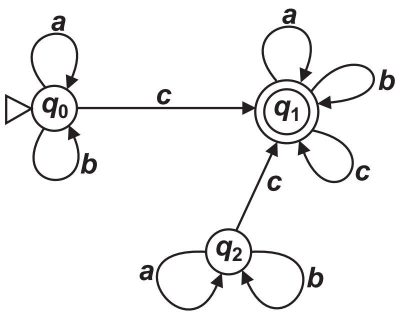

Рис. 3.23

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

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 автомату $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$ 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 і 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 розпізнає 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 $\delta\colon (Q \times T) \to Q$. Рекурсією за індексом k введемо множини X_k ($k \geqslant 0$):

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

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 більше 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 із станів q_1 або q_2 недосяжний з q_0, тобто перейти до множини переходів

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

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

Нарешті, деякі із 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 $\widetilde{M}$ має лише один початковий стан $q_0 \in \widetilde{Q}$, детермінованість автомату $\widetilde{M}$ повністю доведено. $\square$

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

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

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

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 $\widetilde{M} = \langle \widetilde{Q}, \{a, b, c\}, \widetilde{\Delta}, \{q_0\}, \widetilde{F} \rangle$ (див. рис. 3.24), який, за побудовою, еквівалентний вихідному автомату M та містить лише стани, досяжні з початкового стану q_0.

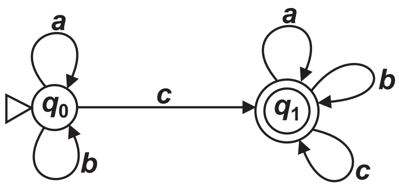

Рис. 3.24

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

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

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 з множини переходів вихідного автомату 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 з множини F, видаляючи стани q_5 та q_6, які є допускаючими у вихідному автоматі:

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

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

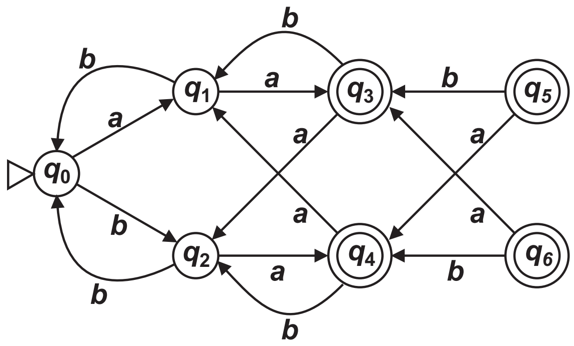

Рис. 3.25

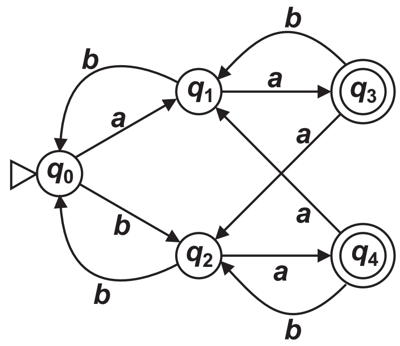

Рис. 3.26

Зауважимо, що процедуру видалення станів, недосяжних з жодного початкового, легко узагальнити на довільні скінченні автомати (див., наприклад, [5]).

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 розпізнає автомат M; $\delta\colon (Q \times T) \to Q$ – функція переходів автомату 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 розпізнає мову L.*

Доведення. **Детермінованість.** Автомат M_{merge} має один початковий стан $[q_0]$. Отже, для доведення детермінованості автомату M_{merge} необхідно показати, що для будь-якого стану $[q_1] \in Q_{\text{merge}}$ ($q_1 \in Q$) і символу $a \in T$ існує єдиний стан $\alpha \in Q_{\text{merge}}$, такий, що $([q_1], a, \alpha) \in \Delta_{\text{merge}}$. Враховуючи детермінованість автомату 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 автомату 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 $w \in T^*$ сприймається автоматом M_{merge} тоді і тільки тоді, коли воно сприймається 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 $Q_{\mathrm{merge}} = Q^*/_{\sim}$ та $T^*/_{\underset{L}{\sim}}$ існує взаємно однозначна відповідність (див. заув. 3.11), отримуємо рівність $|Q_{\mathrm{merge}}| = \left|T^*/_{\underset{L}{\sim}}\right| = i_L$. Таким чином, автомат M_{merge} має мінімально можливу кількість станів серед детермінованих автоматів, які розпізнають мову L. □

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

Приклад 3.34. Об'єднуючи еквівалентні стани скінченного автомату $M = \langle Q, \{a, b\}, \Delta, \{q_0\}, F\rangle$ з прикладів 3.25, 3.30 (див. рис. 3.20), отримуємо автомат, зображений на рис. 3.27.

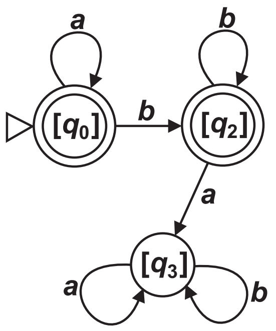

Рис. 3.27

Класи еквівалентності за відношенням «∼», як було показано у прикладі 3.30, мають вигляд:

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

Отриманий автомат дійсно еквівалентний вихідному автомату M, оскільки сприймає 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) $f\colon Q_1 \to Q_2$, 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 ізо-

морфний 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 $\delta_1 : (Q_1 \times T) \to Q_1$ 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 на рис. 3.27, однак жоден із цих автоматів не ізоморфний автомату на рис. 3.20. Зазначимо, що всі чотири автомати еквівалентні, оскільки розпізнають ту саму 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 розпізнає мову $L = L[M]$, отриманий за процедурою (3.6):

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

Оскільки всі стани вихідного автомату 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. завув. 3.11), де $w \in T^*$, $q = \delta^*(q_0, w)$, $\delta^* \colon (Q \times T^*) \to Q$ – розширена функція переходів автомату M.

Лема 3.6. *Відображення f є ізоморфізмом автомату 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:

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

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

Отже, визначена бієкція $f : Q/_{\sim} \to T^*/_{\underset{L}{\sim}}$ дійсно 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 автомату M, зображеного на рис. 3.20 (див. приклад 3.34). Автомат M_L, зображений на рис. 3.22 – мінімальний автомат, який розпізнає 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 $\alpha \in Q/_{\sim}$.

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

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 розпізнає 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 розпізнають мову L. Із мінімальності автомату M випливає:

1) кожний стан $q \in Q$ автомату 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 $q \in Q$ автомату M еквівалентний лише собі, тобто $[q] = \{q\}$, і $Q/_\sim = \{\{q\} \colon q \in Q\}$ (різні еквівалентні стани можна об'єднати, отримавши еквівалентний автомат з меншою кількістю станів). Таким чином, можна встановити очевидну бієкцію $\rho \colon Q \to Q/_\sim$, $\rho(q) = \{q\}$ 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. $\square$

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

Приклад 3.37. На рис. 3.21, 3.22 та 3.27 зображено автомати, які розпізнають 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 розпізнають мову L. Очевидно, що всі три автомати дійсно між собою ізоморфні.

3.7. Алгоритми мінімізації детермінованих скінченних автоматів

У цьому підрозділі розглянемо конкретні алгоритми побудови мінімального за кількістю станів детермінованого скінченного автомату, 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 автомату $M = \langle Q, T, \Delta, \{q_0\}, F \rangle$.

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 $\ll\overset{k}{\sim}\gg$ ($k \geqslant 0$) негайно отримуємо ланцюжок вкладень

$$\ll\overset{0}{\sim}\gg \supset \ll\overset{1}{\sim}\gg \supset \cdots \supset \ll\overset{k}{\sim}\gg \supset \cdots \supset \ll\sim\gg. \tag{3.12}$$

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

Доведення. Позначимо через X множину пар нееквівалентних станів:

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

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:

$$\ll\overset{0}{\sim}\gg \supset \ll\overset{1}{\sim}\gg \supset \cdots \supset \ll\overset{n}{\sim}\gg = \ll\sim\gg \tag{3.13}$$

для деякого $n \geqslant 0$.

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

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

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 протирічить 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* $\ll \overset{k}{\sim} \gg = \ll \overset{k+1}{\sim} \gg$ *для деякого* $k \geqslant 0$. *Тоді* $\ll \overset{k+1}{\sim} \gg = \ll \overset{k+2}{\sim} \gg$.

Доведення. Нехай $k \geqslant 0$ та $\ll \overset{k}{\sim} \gg = \ll \overset{k+1}{\sim} \gg$. Необхідно довести рівність $\ll \overset{k+1}{\sim} \gg = \ll \overset{k+2}{\sim} \gg$, а з урахуванням ланцюжка (3.13) – лише вкладення $\ll \overset{k+1}{\sim} \gg \subset \ll \overset{k+2}{\sim} \gg$. 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 $q_1 \overset{k+2}{\nsim} q_2$); з іншого боку, $|w| > k + 1$ (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 $\ll \overset{k}{\sim} \gg = \ll \overset{k+1}{\sim} \gg$, 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, $p_1 \overset{k}{\sim} p_2$. Отримане протиріччя доводить лему. □

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

$$\ll \overset{0}{\sim} \gg \supsetneq \ll \overset{1}{\sim} \gg \supsetneq \cdots \supsetneq \ll \overset{n}{\sim} \gg = \ll \sim \gg \tag{3.14}$$

для деякого $n \geqslant 0$.

Нехай i_k $(k \geqslant 0)$ – індекс відношення $\ll\overset{k}{\sim}\gg$, i_Q – індекс відношення еквівалентності станів $\ll\sim\gg$, 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 $\ll\overset{k}{\sim}\gg \supsetneq \ll\overset{k+1}{\sim}\gg$, то $i_k < i_{k+1}$.

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

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

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

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 $\ll\overset{0}{\sim}\gg \supsetneq \ll\overset{1}{\sim}\gg = \ll\sim\gg$, де

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

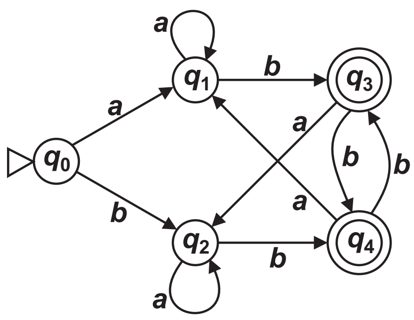

Рис. 3.28

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 дійсно не розрізняє жодне слово; що стани q_3 та q_4 дійсно не розрізняє жодне слово.

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

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

Запишемо розширену функцію переходів цього автомату:

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

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 $\ll \overset{k}{\sim} \gg$ $(k \geqslant 0)$:

$$\begin{aligned} \ll \overset{0}{\sim} \gg &= \{\{q_0\}, \{q_1, q_2, \ldots, q_{m-1}\}\}, \\ \ll \overset{1}{\sim} \gg &= \{\{q_0\}, \{q_{m-1}\}, \{q_1, q_2, \ldots, q_{m-2}\}\}, \\ &\vdots \\ \ll \overset{m-2}{\sim} \gg &= \{\{q_0\}, \{q_{m-1}\}, \{q_{m-2}\}, \ldots, \{q_1\}\}. \end{aligned}$$

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 автомату $M = \langle Q, T, \Delta, \{q_0\}, F\rangle$, який містить принаймні 2 стани, існує не більше ніж $|Q|-1$ різних відношень $\ll \overset{k}{\sim} \gg$ $(k \geqslant 0)$.*

Доведення. Лемма стверджує, що довжина ланцюжків (3.15) та (3.14) (кількість різних відношень $\ll \overset{k}{\sim} \gg$, $k \geqslant 0$) не перевищує $|Q|-1$. Одразу зазначимо, що $i_Q = \left|Q/_{\sim}\right| \leqslant |Q|$; з іншого боку, $i_0 \geqslant 2$ 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 $[2; |Q|]$ знаходиться 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 $\ll \overset{0}{\sim} \gg$. 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 $\ll \overset{0}{\sim} \gg$. 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 $\ll \overset{|Q|-2}{\sim} \gg \supsetneq \ll \overset{m}{\sim} \gg$ для деякого $m \geqslant |Q| - 1$ і, за лемою 3.9, відношення $\ll \overset{k}{\sim} \gg$ $(0 \leqslant k \leqslant |Q| - 1)$ є попарно різними. Таким чином, ланцюжок (3.14) містить не менш ніж $|Q|$ різних відношень, що протирічить 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.

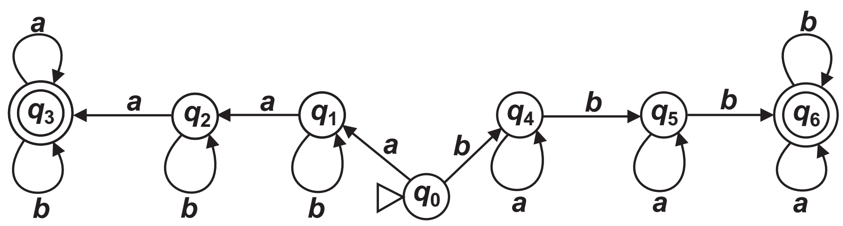

Рис. 3.29

Неважко зрозуміти, що ланцюжок вкладень (3.14) набуває вигляду $\ll \overset{0}{\sim} \gg \supsetneq \ll \overset{1}{\sim} \gg \supsetneq \ll \overset{2}{\sim} \gg = \ll \sim \gg$, 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 автомату можна розрізнити деяким словом довжиною не більш ніж $|Q|-2=7-2=5$. Насправді, в даному 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 автомату $M=\langle Q,\{a\},\Delta,\{q_0\},\{q_0\}\rangle$ з 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 автомату, 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 автомату 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

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

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

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

5. Якщо $\ll \overset{k+1}{\sim} \gg \, = \, \ll \overset{k}{\sim} \gg$, закінчити роботу.
6. Збільшити k на 1 і перейти до п. 3.

Зауваження 3.15. Зазначимо, що алгоритм визначає факт $q_i \overset{k}{\sim} q_j$ чи $q_i \overset{k}{\not\sim} q_j$ $(k \geqslant 0)$ лише для таких пар станів $q_i, q_j \in Q$, коли $i < j$. З урахуванням симетричності відношень еквівалентності, це визначає $q_j \overset{k}{\sim} q_i$ чи $q_j \overset{k}{\not\sim} q_i$ $(k \geqslant 0,\ i < j)$. Нарешті, за рефлексивністю відношень еквівалентності, $q \overset{k}{\sim} q$ для всіх $q \in Q$, $k \geqslant 0$.

Зауваження 3.16. Згідно з лемами 3.9 та 3.10, останнє обчислене наведеним алгоритмом відношення $\ll \overset{k+1}{\sim} \gg$ збігається з відношенням еквівалентності станів $\ll\sim\gg$ автомату M. Відношення $\ll\sim\gg$ 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. Якщо існує $a \in T$, такий, що $\delta(q_i, a) \overset{k+1}{\not\sim} \delta(q_j, a)$, клас еквівалентності A розділяється за символом a. Якщо $\delta(q_i, a) \overset{k+1}{\sim} \delta(q_j, a)$ 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 автомату M визначимо, які із станів автомату 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 автомату 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

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

Аналогічно, з табл. 3.5 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 «$\overset{1}{\sim}$» (див. 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 $q_0 \overset{1}{\sim} q_3$, $q_1 \overset{2}{\sim} q_5$, $q_2 \overset{1}{\sim} q_4$, 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 автомату M за процедурою (3.9), отримуємо зображений на рис. 3.31 мінімальний детермінований скінченний автомат M_{merge}, еквівалентний автомату M.

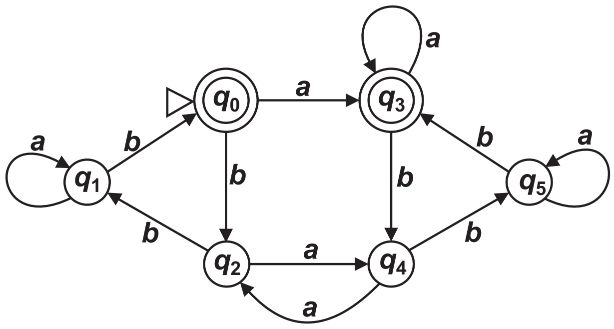

Рис. 3.30

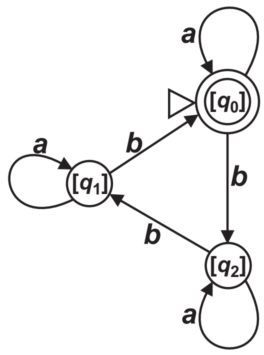

Рис. 3.31

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

Детальніше про алгоритм побудови відношень k-еквівалентності, а також про його модифікації, див., наприклад, [5].

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 $k + 1$.

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). Вважатимемо, що $|Q| \geqslant 2$ (якщо автомат містить лише один стан, мінімізація, очевидно, неможлива). Також вважатимемо, що всі стани автомату 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. 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 більше 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 автомату 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. [10]).

Приклад 3.43. Застосуємо алгоритм заповнення таблиці до автомату M із прикладу 3.42 (рис. 3.30). Зазначимо, що всі стани автомату M

досяжні з початкового стану q_0 (це було перевірено у прикладі 3.42). У таблицю 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_1	×				
q_2	×				
q_3		×	×		
q_4	×			×	
q_5	×			×	
	q_0	q_1	q_2	q_3	q_4

Поклавши $k = 0$, обчислимо множину A_1, яка, згідно з визначенням, містить всі 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 символу $x \in \{a, b\}$ визначимо набір пар $\{p, q\}$, таких, що $\{\delta(p, x), \delta(q, x)\} = \{\widetilde{p}, \widetilde{q}\}$ (див. 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 $\{q_1, q_2\} \in A_1$ дійсно 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 таблицю 3.11. Бачимо, що множина A_1 містить 4 пари станів, які не входять в A_0 (підкреслені в таблиці 3.10), тобто розрізняються словом довжини 1, але не розрізняються словом ε.

Таблиця 3.11

q_1	×				
q_2	×	×			
q_3		×	×		
q_4	×	×		×	
q_5	×		×	×	×
	q_0	q_1	q_2	q_3	q_4

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 символу $x \in \{a, b\}$ визначимо набір пар $\{p, q\}$, таких, що $\{\delta(p, x), \delta(q, x)\} = \{\widetilde{p}, \widetilde{q}\}$ (див. 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 (див. 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 таблиці 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 наведено на рис. 3.31.

Детальніше про алгоритм заповнення таблиці нееквівалентних станів див. [10].

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 автомату, еквівалентного заданому (навіть якщо заданий автомат є детермінованим); див. контрприклад в [10]. Мінімізація недетермінованих скінченних автоматів досліджувалась, наприклад, Я. Бржозовським [].

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 замкненості 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 $f(x_1, x_2, \dots, x_n) \in Y$ для будь-яких $x_1, x_2, \dots, x_n \in Y$ (детально про алгебричні структури див., наприклад, [1]).

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

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

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

□

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

Доведення. **Об'єднання.** Нехай мови L_1 та L_2 розпізнаються 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 $L_1 \cup L_2$ розпізнає 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 розпізнаються 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 $L_1 \cdot L_2$ розпізнає 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 розпізнає детермінований скінченний автомат $\langle Q_1, T, \Delta_1, \{p_1\}, F\rangle$ (за теоремою 3.1, такий автомат існує). Тоді мову $\overline{L_1}$ розпізнає автомат $\langle Q_1, T, \Delta_1, \{p_1\}, Q \setminus F\rangle$.

Перетин. За законом де Моргана (див., наприклад, [1, 2]) 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 розпізнається скінченним автоматом $\langle Q_1, T, \Delta_1, \{p_1\}, \{q_1\}\rangle$ (за лемою 3.12, такий автомат існує). Тоді мову L_1^* розпізнає автомат

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

де $q_* \notin Q_1$. □

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

Детально про булеві алгебри та алгебри множин див., наприклад, [11].

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 $\{a, b, c\}$. Тоді скінченний автомат, який розпізнає мову $L_1 \cup L_2 \cup L_3$, можна отримати з автоматів, які розпізнають мови L_1, L_2 та L_3 (див. рис. 3.32).

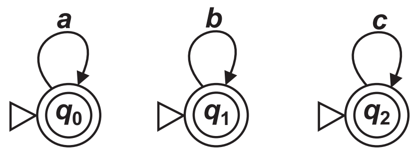

Рис. 3.32. Автомат, що розпізнає мову $\{a^n : n \geqslant 0\} \cup \{b^n : n \geqslant 0\} \cup \{c^n : n \geqslant 0\}$

Скінченний автомат із прикладу 3.15 (рис. 3.14), що сприймає 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 розпізнають мови L_1, L_2 та L_3.

Зображений на рис. 3.32 скінченний автомат, що розпізнає мову $L_1 \cdot L_2 \cdot L_3$, містить рівно один початковий та рівно один допускаючий стан, що дозволяє побудувати скінченний автомат, який розпізнає формальну мову $(L_1 \cdot L_2 \cdot L_3)^*$ (рис. 3.33).

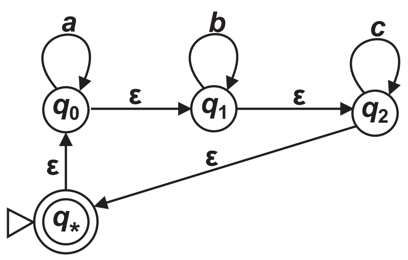

Рис. 3.33. Автомат, що розпізнає мову $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}^*$

Скінченний автомат, зображений на рис. 3.32, не є детермінованим (більше того, містить ε-переходи), а отже не може бути використаний для побудови доповнення до мови $L_1 \cdot L_2 \cdot L_3$. На рис. 3.44

зображено детермінований скінченний автомат, який розпізнає мову $L_1 \cdot L_2 \cdot L_3 = \{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}$, а також автомат, який розпізнає мову $\overline{\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}}$.

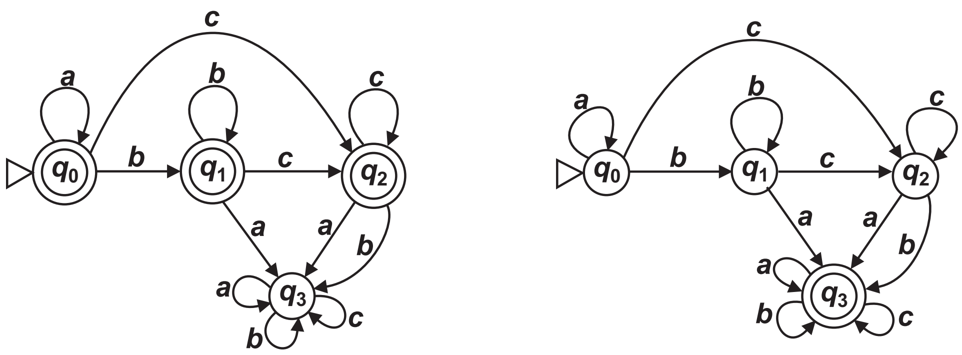

Рис. 3.34. Детерміновані автомати, що розпізнають мови $\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}$ та $\overline{\{a^n b^m c^k : n \geqslant 0, m \geqslant 0, k \geqslant 0\}}$

Зауваження 3.19. Замкненість класу регулярних мов відносно перетину можна довести, безпосередньо побудувавши відповідний скінченний автомат: якщо мови L_1 та L_2 розпізнаються 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 $L_1 \cap L_2$ розпізнає автомат

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

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

3.8.2. Видалення ε-продукцій в 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 додаткові ε-продукції (якщо $(S_0 \to \varepsilon) \in P_0$, то $(S \to \varepsilon) \in P$).

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 умовам теореми 3.11. Тоді мова $L \setminus \{\varepsilon\}$ породжується граматикою $\langle V, T, P \setminus \{S \to \varepsilon\}, S\rangle$. Зазначимо, що множина продукцій $P \setminus \{S \to \varepsilon\}$ дійсно 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 дійсно 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 дійсно 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 1.15 та 1.16, 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 виконується умова: $xu^i y \in L$ для всіх $i \geqslant 0$.*

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 $w = a_1 a_2 \ldots a_N \in L$ ($a_k \in T$, $1 \leqslant k \leqslant N$) довжиною $|w| = N \geqslant n$ є непорожнім і може бути виведене із джерела S за N кроків:

$$A_0 \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow \cdots \Rightarrow a_1 a_2 \ldots a_{N-1} A_{N-1} \Rightarrow a_1 a_2 \ldots a_N = w, \tag{3.18}$$

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

$$A_0 \overset{*}{\Rightarrow} a_1 a_2 \ldots a_i A \overset{*}{\Rightarrow} a_1 a_2 \ldots a_i a_{i+1} \ldots a_j A \overset{*}{\Rightarrow} w.$$

Увівши позначення $x = a_1 a_2 \ldots a_i$, $u = a_{i+1} a_{i+2} \ldots a_j$, $y = a_{j+1} a_{j+2} \ldots a_N$, отримуємо виведення

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xuy = w. \tag{3.19}$$

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

$$A_0 \overset{*}{\Rightarrow} xA \overset{*}{\Rightarrow} xuA \overset{*}{\Rightarrow} xu^2 A \overset{*}{\Rightarrow} \cdots \overset{*}{\Rightarrow} xu^i A \overset{*}{\Rightarrow} xu^i y$$

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 автомату (див., наприклад, [3, 5, 8]).

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. Дійсно, фіксуючи $n = \max\{|w| : w \in L\} + 1$, 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 $L = \{a^m b^m : m \geqslant 0\}$ нерегулярна. Припустимо, що L – регулярна мова, і n – додатна константа, існування якої постулюється лемою про розростання. Тоді для слова $a^n b^n \in L$ мають існувати такі $x, u, y \in \{a, b\}^*$, $u \neq \varepsilon$, що $w = xuy$, $xu^i y \in L$ для всіх

$i \geqslant 0$, та $|xu| \leqslant n$. Але тоді u містить лише символи a, тобто $|u|_b = 0$. З іншого боку, $|xuy|_a = |xuy|_b$ 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 $u = \varepsilon$. Отримане протиріччя доводить, що мова L нерегулярна.

Підкреслимо, 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

$$L = \{a^j b^m c^m d^k : j, m, k \geqslant 1\} \cup \{b^{m_1} c^{m_2} : m_1, m_2 \geqslant 1\}$$

можна вибрати константу $n = 1$, і довільне $w \in L$ довжиною $|w| \geqslant 1$ має вигляд $aa^j b^m c^m d^k$ ($j \geqslant 0$, $m \geqslant 1$, $k \geqslant 1$) або $bb^{m_1} c^{m_2} d^k$ ($m_1 \geqslant 0$, $m_2 \geqslant 1$, $k \geqslant 1$). В обох випадках можна використати розклад $w = xuy$ із $x = \varepsilon$, $|u| = 1$, тобто за u взяти перший символ слова w. Легко перевірити, що цей розклад в обох випадках відповідає твердженню теореми (3.12):

1) $w = aa^j b^m c^m d^k = \varepsilon \cdot a \cdot a^j b^m c^m d^k$, $\varepsilon \cdot a^i \cdot a^j b^m c^m d^k \in L$ для всіх $i \geqslant 0$;

2) $w = bb^{m_1} c^{m_2} d^k = \varepsilon \cdot b \cdot b^{m_1} c^{m_2} d^k$, $\varepsilon \cdot b^i \cdot b^{m_1} c^{m_2} d^k \in L$ для всіх $i \geqslant 0$.

Зазначимо, що при заміні 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 твердженню леми про розростання, однак ця мова нерегулярна. Дійсно, якщо б L була регулярною мовою, то, оскільки мова $\{ab^{m_1} c^{m_2} d : m_1, m_2 \geqslant 1\}$ регулярна, мова $L \cap \{ab^{m_1} c^{m_2} d : m_1, m_2 \geqslant 1\} = \{ab^m c^m d : m \geqslant 1\}$, згідно з теоремою 3.10, також була б регулярною. Однак нерегулярність мови $\{ab^m c^m d : m \geqslant 1\}$ неважко встановити лемою про розростання, аналогічно доведенню нерегулярності мови $\{a^m b^m : m \geqslant 0\}$ (див. приклад 3.47).

Приклад 3.49. Доведемо, що мова $L = \{a^q : q$ – просте число$\}$ нерегулярна.Припустимо, що L – регулярна мова і n – додатня 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 $i \geq 0$, та $|xu| \leqslant n$. Розглянемо два випадки.

Перший випадок $n = 1$. Оскільки $u \neq \varepsilon$, то $u = a$, звідки $x = \varepsilon$, $y = \varepsilon$. Покладаємо $i = 4$: $xu^iy = a^4 \in L$, 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

$$xu^iy = a^{|x|}(a^{|u|})^{n+1}a^{n-|x|-|u|} = a^{|x|+|u|(n+1)+n-|x|-|u|} = a^{n(|u|+1)} \in L.$$

Але число $n(|u| + 1)$ складене, оскільки $n \geqslant 2$ та $|u| + 1 \geqslant 2$. Отримана суперечність доводить нерегулярність мови L. Зауважимо без доведення, що мова L не контекстно-вільна, але контекстно-залежна.

Використовуючи метод доведення леми про розростання, нескладно довести таке твердження.

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 для нескінченності мови L необхідно і достатньо існування такого слова $w \in L$, що $|V| + 1 \leqslant |w| \leqslant 2|V|$.*

Приклад 3.50. 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 $3 + 1 \leqslant |w| \leqslant 2 \cdot 3$. Дійсно, 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 $\{ac, abc\}$; 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 [3, 13]).

Лема 3.14. *Для будь-якої регулярної мови L існує скінченний набір невід'ємних цілих чисел a_1, $a_2, \ldots$, a_n, d_1, $d_2, \ldots$, 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 3.51. Для регулярної мови $\{a^{4n}b^{6m} : n \geqslant 0, m \geqslant 0\}$ отримуємо множину довжин

$$\{4n + 6m \colon n \geqslant 0, m \geqslant 0\} = \{0\} \cup \{4 + 2k \colon k \geqslant 0\}.$$

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 3.52. 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 «·», якщо це не викликає непорозумінь, взагалі опускатимемо, тобто замість $r + s \cdot t$ писатимемо $r + st$.

З 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 3.53. 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 $(s + t) \cdot r = s \cdot r + t \cdot r$ доводиться 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 $L[r^+] = (L[r])^+$.

Детальніше про властивості регулярних виразів див., наприклад, [3, 5, 10]. Про методи доведення тотожностей в алгебрі регулярних виразів див., зокрема, [10].

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 автомату M назвемо довільний набір $(q, w) \in (Q \times T^*)$. На множині конфігурацій узагальненого автомату 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 автомату M можна об'єднати в один перехід $(q, r_1 + r_2, p)$, не змінюючи мову $L[M]$; надалі такі переходи називатимемо *паралельними*. З іншого боку, 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 автомату 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 зображувати у вигляді графів.

Приклад 3.54. Зображені на рис. 3.35 та 3.36 узагальнені скінченні автомати над алфавітом $\{a, b, c\}$ сприймають 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)).

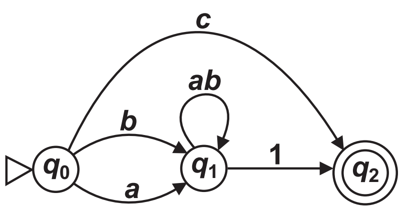

Рис. 3.35

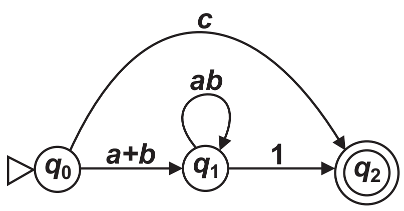

Рис. 3.36

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 розпізнається 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). Для знаходження регулярного виразу, якому відповідає мова L, спростимо автомат M, видаливши із множини Q всі стани, окрім 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 $(q, r_{qt}r_{tt}^*r_{tp})$ 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 автомату із стану q в 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 сприймає мову, яка відповідає регулярному виразу

$$r = (r^*_{q_0,q_0} r_{q_0,q_f} r^*_{q_f,q_f} r_{q_f,q_0})^* r^*_{q_0,q_0} r_{q_0,q_f} r^*_{q_f,q_f}, \tag{3.20}$$

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 $\{L[r] \colon r \in R[T]\}$.*

Доведення. Твердження теореми негайно випливає з лем 3.15 та 3.16. □

Наслідок. *Клас регулярних мов над алфавітом T є найменшою за відношенням вкладення («⊂») 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 (3.20).

Приклад 3.55. Користуючись методом видалення станів, визначимо регулярний вираз, якому відповідає регулярна мова

$$L = \{w \in \{a, b\}^* \colon |w|_a - |w|_b = 4k + 1,\ k \in \mathbb{Z}\};$$

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 $4k$ та не допустити слова з різницями $4k+1$, $4k+2$ та $4k+3$. Отже, для побудови автомату достатньо чотирьох станів q_0 (початковий), q_1, q_2 та q_3; $(q_0, w) \vdash^* (q_i, \varepsilon)$

тоді і тільки тоді, коли $|w|_a - |w|_b = 4k + i$ $(i = 0, 1, 2, 3)$. Такий автомат наведено на рис. 3.37; його можна розглядати як узагальнений, порожні переходи з метою спрощення на рисунку не наведені.

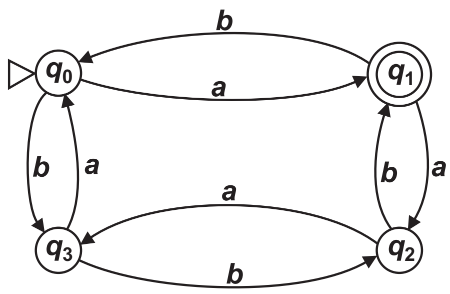

Рис. 3.37

Для видалення стану q_2 необхідно додати чотири переходи: (q_1, aa, q_3), (q_3, bb, q_1), (q_1, ab, q_1), (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). Отриманий автомат наведено на рис. 3.38.

Аналогічно видаляємо стан q_3. Отриманий автомат (після об'єднання паралельних переходів) наведено на рис. 3.39.

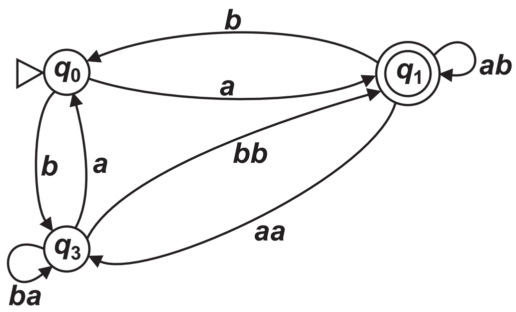

Рис. 3.38

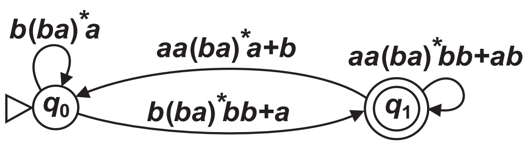

Рис. 3.39

Користуючись формулою (3.20), можемо знайти регулярний вираз, якому відповідає мова L:

$$r_1 = \big((b(ba)^*a)^*(b(ba)^*bb + a)(aa(ba)^*bb + ab)^*(aa(ba)^*a + b)\big)^* \cdot \\ \cdot (b(ba)^*a)^*(b(ba)^*bb + a)(aa(ba)^*bb + ab)^*. \qquad (3.21)$$

Зауваження 3.24. Зауважимо, що в автоматі на рис. 3.38 можна замінити переходи (q_3, a, q_0) та (q_1, b, q_0) на чотири переходи (q_3, ab, q_3), (q_1, ba, q_1), (q_3, aa, q_1) та (q_1, bb, q_3) та об'єднати їх з існуючими паралельними (див. рис. 3.40), а лише після цього видалити стан q_3 (див. рис. 3.41).

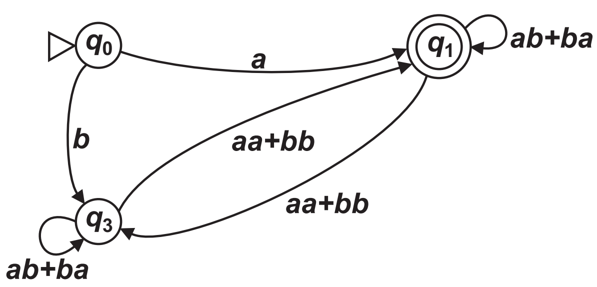

Рис. 3.40

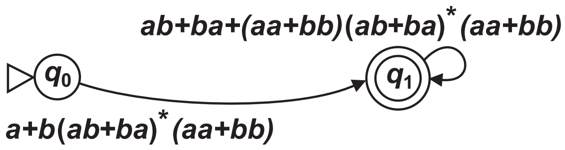

Рис. 3.41

В такому випадку отримуємо простіший регулярний вираз, якому відповідає мова L:

$$r_2 = \big(a+b(ab+ba)^*(aa+bb)\big)\big(ab+ba+(aa+bb)(ab+ba)^*(aa+bb)\big)^*. \quad (3.22)$$

Вправа 3.25. Визначити регулярні вирази, яким відповідають регулярні мови із прикладів 3.8 та 3.9.

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. [14, 15]). У прикладі 3.55 висота регулярного виразу (3.21) $\operatorname{sh} r_1 = 3$, висота регулярного виразу (3.22) $\operatorname{sh} r_2 = 2$, а висота регулярної мови $\operatorname{sh} L = 2$, оскільки не існує регулярного виразу висотою 1, якому відповідає мова L (див. [15]). 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 автомату.

Розділ 4

Контекстно-залежні граматики та лінійно-обмежені машини Тьюрінга

4.1. Невкорочуючі 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 4.1. Граматику $G = \langle V, T, P, S\rangle$ називають «невкорочуючою», якщо для кожної продукції вигляду $p = (\alpha \to \beta) \in P$ має місце нерівність $|\alpha| \leqslant |\beta|$.

Зауважимо, що граматики з прикладу 1.12, п. 3 та прикладу 1.14 є невкорочуючими. Продукції контекстно-залежних граматик не зменшують довжину поточного слова, отже, кожна контекстно-залежна граматика є невкорочуючою. Виявляється, що клас мов, породжених невкорочуючими граматиками, повністю вичерпується контекстно-залежними мовами.

Теорема 4.1. *Нехай формальна мова L породжена невкорочуючою граматикою $G = \langle V, T, P, S \rangle$. Тоді L є контекстно-залежною мовою.*

Доведення. Для кожного термінального символу $a \in T$, який міститься в лівій частині хоча б однієї продукції, введемо новий нетермінальний символ $T_a \notin V$, замінимо усі входження символу 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 $p_i = (A_1 A_2 \ldots A_m \to \beta_1 \beta_2 \ldots \beta_n) \in P$ $(n \geqslant m \geqslant 2)$ не дозволеного для контекстно-залежних 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:

$$
\begin{aligned}
A_1 A_2 A_3 \ldots A_{m-1} A_m &\to X_1^i A_2 A_3 \ldots A_{m-1} A_m,\\
X_1^i A_2 A_3 \ldots A_{m-1} A_m &\to X_1^i X_2^i A_3 \ldots A_{m-1} A_m,\\
&\vdots\\
X_1^i X_2^i X_3^i \ldots X_{m-1}^i A_m &\to X_1^i X_2^i X_3^i \ldots X_{m-1}^i X_m^i,\\
X_1^i X_2^i X_3^i \ldots X_{m-1}^i X_m^i &\to \beta_1 X_2^i X_3^i \ldots X_{m-1}^i X_m^i,\\
\beta_1 X_2^i X_3^i \ldots X_{m-1}^i X_m^i &\to \beta_1 \beta_2 X_3^i \ldots X_{m-1}^i X_m^i,\\
&\vdots\\
\beta_1 \beta_2 \ldots \beta_{m-2} X_{m-1}^i X_m^i &\to \beta_1 \beta_2 \ldots \beta_{m-1} X_m^i,\\
\beta_1 \beta_2 \beta_3 \ldots \beta_{m-1} X_m^i &\to \beta_1 \beta_2 \beta_3 \ldots \beta_{m-1} \beta_m \beta_{m+1} \ldots \beta_n.
\end{aligned}
$$

Зазначимо, що символи A_j можуть зустрічатись у різних продукціях, однак унікальність символів X_k^i та контекстність замін гарантує неможливість застосування наведених продукцій інакше, ніж для заміни слова $A_1 A_2 \ldots A_m$ на $\beta_1 \beta_2 \ldots \beta_n$. Також зазначимо, що продукція $\varepsilon \to \varepsilon$ невкорочуючої граматики не змінює мову, породжену граматикою, а продукцію $\varepsilon \to \beta$ можна замінити згідно з зауваженням 1.5. Побудована граматика G_2 еквівалентна граматиці G_1, а тому й G. □

Зауваження 4.1. 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 4.2. Іноді в літературі (див., наприклад, [5]) під контекстно-залежною граматикою розуміють граматику, яка задовольняє означенню 4.1.

Приклад 4.1. 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:

$$S \Rightarrow aSBC \Rightarrow a^2(BC)^2 \Rightarrow a^2B^2C^2 \Rightarrow a^2bBC^2 \Rightarrow a^2b^2C^2 \Rightarrow a^2b^2cC \Rightarrow a^2b^2c^2,$$
$$S \Rightarrow aSBC \Rightarrow a^2(BC)^2 \Rightarrow a^2bCBC \Rightarrow a^2bBC^2 \Rightarrow a^2b^2C^2 \Rightarrow a^2b^2cC \Rightarrow a^2b^2c^2.$$

Дана граматика невкорочуюча, отже, за теоремою 4.1, мова $L = L[G]$ контекстно-залежна (зазначимо, що згідно з підрозд. **??** мова L не контекстно-вільна). Побудуємо еквівалентну контекстно-залежну граматику за методом із доведення теореми 4.1. Згідно з зауваженням 4.1 позбуватись термінальних символів у лівих частинах продукцій не потрібно, оскільки всі такі продукції дозволеного вигляду. Єдину продукцію $CB \to BC$ не дозволеного 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 4.3. Клас контекстно-залежних мов замкнений відносно операцій об'єднання, перетину, конкатенації, доповнення, обертання та замикання Кліні [3, 6].

4.2. Лінійно-обмежені машини Тьюрінга

Лінійно-обмежені машини Тьюрінга (інший термін лінійно-обмежені автомати), у порівнянні із 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 4.2. 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 4.4. Іноді в літературі лінійно-обмеженість машини Тьюрінга визначають за допомогою маркерів (див., наприклад, [6]). Вхідне слово обмежують зліва та зправа маркером – символом «#», а курсор машини Тьюрінга $\langle Q, X, T, \Lambda, \#, \Delta, I, F\rangle$ в процесі роботи не виходить за межі, визначені маркерами. Також є варіант обмеження вхідного слова зліва маркером «$\#_l$», а зправа – «$\#_r$», тоді машину Тьюрінга можна записати у вигляді $\langle Q, X, T, \Lambda, \#_l, \#_r, \Delta, I, F\rangle$ (див., наприклад, [16]).

Зауваження 4.5. Лінійно-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 4.2 шляхом розширення алфавіту: наприклад, при $k = 2$ та робочому алфавіті $\{a, b, \Lambda\}$ можна перейти до робочого алфавіту $\{t_{\Lambda\Lambda}, t_{\Lambda a}, t_{\Lambda b}, t_{a\Lambda}, t_{aa}, t_{ab}, t_{b\Lambda}, t_{ba}, t_{bb}\}$ та 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 4.2. 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 4.2. *Для довільної контекстно-залежної формальної мови L існує (взагалі кажучи, недетермінована) лінійно-обмежена машина Тьюрінга M, яка сприймає мову L.*

Доведення. Нехай мова $L = L[G]$ породжена контекстно-залежною, а, отже, невкорочуючою граматикою $G = \langle V, T, P, S\rangle$. Нехай «$*$» – деякий символ, який не міститься у множинах T та V. Для кожної продукції $p_j = (\alpha_1\alpha_2\dots\alpha_m \to \beta_1\beta_2\dots\beta_n) \in P$ $(n > m \geqslant 1)$ введемо переходи, які замінюють одне з входжень слова $\beta_1\beta_2\dots\beta_n$ на $\alpha_1\alpha_2\dots\alpha_m *^{n-m}$:

$$((q_0, \beta_1), (q_j^2, \alpha_1, r)),$$
$$((q_j^k, \beta_k), (q_j^{k+1}, \alpha_k, r)) \quad (2 \leqslant k \leqslant m),$$
$$((q_j^k, \beta_k), (q_j^{k+1}, *, r)) \quad (m+1 \leqslant k \leqslant n-1),$$
$$((q_j^n, \beta_n), (q_0, *, s));$$

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 $((q_0, \beta_1), (q_0, \alpha_1, s))$. Як і при доведенні теореми 4.1 зазначимо, що продукція $\varepsilon \to \varepsilon$ невкорочуючої граматики G не змінює мову, породжену G, а продукцію $\varepsilon \to \beta$ можна замінити згідно з зауваженням 1.5.

Далі введемо команди пересування курсора, які забезпечують пошук деякого слова $\beta_1\beta_2\dots\beta_n$ для наступної його заміни на $\alpha_1\alpha_2\dots\alpha_m *^{n-m}$: $((q_0, \xi), (q_0, \xi, l))$, $((q_0, \xi), (q_0, \xi, r))$ $(\xi \in T \cup V \cup \{*\})$.

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 4.3. *Нехай формальну мову L, що не містить порожнього слова, сприймає 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

$$\langle Q \cup \{\tilde{q}_a\}, X, T, \Lambda, \Delta \cup \{((q, \alpha), (\tilde{q}_a, \alpha, s)) \colon q \in F, \alpha \in X\}, I, \{\tilde{q}_a\}\rangle.$$

Побудуємо невкорочуючу 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 символу $\beta \in X$ 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 символу $\beta \in X$ 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 символу $\alpha \in T$ та кожного стану $q \in I$.

Легко перевірити, що з 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 4.1 існує еквівалентна контекстно-залежна граматика. Зауважимо, що в загальному випадку граматику G можна спростити, прибравши ті символи $A_{\alpha,q}$ та продукції, які не використовуються. $\square$

Приклад 4.3. 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 сприймає регулярну мову $L = \{a^n : n \in \mathbb{N}\}$, побудуємо невкорочуючу граматику, яка породжує мову L, використовуючи метод із доведення теореми 4.3. 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) відповідні переходи відсутні;

5) продукції $A_{a,q_0} \to a$, $A_{b,q_0} \to b$.

Вправа 4.1. Переконатись, що граматика з прикладу 4.3 дійсно породжує мову L, побудувавши для цього схему виведення (див. приклад 1.12). Спростити наведену граматику, прибравши нетермінальні символи та продукції, які не використовуються.

Список літератури

1. Кук Д. Компьютерная математика [Текст] / Д. Кук, Г. Бейз. – М.: Наука, 1990. – 384 с. – 23000 экз. – ISBN 5-02-014216-6.
2. Новиков Ф. Дискретная математика для программистов [Текст] / Ф. Новиков. – Санкт-Петербург: Питер, 2000. – 304 с. – 5000 экз. – ISBN 5-272-0183-4.
3. Пентус А. Е. Теория формальных языков [Текст] / А. Е. Пентус, М. Р. Пентус. – М.: Издательство Центра прикладных исслледований при механико-математическом факультете МГУ, 2004. – 80 с. – 100 экз.
4. Лихтарников Л. М. Математическая логика. Курс лекций. Задачник-практикум и решения [Текст] / Л. М. Лихтарников, Т. Г. Сукачева. – Санкт-Петербург: Издательство «Лань», 1999. – 288 с. – 3000 экз. – ISBN 5-8114-0082-9.
5. Ахо А. Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ [Текст] / А. Ахо, Дж. Ульман. – М.: Мир, 1978. – 612 с. – ??? экз.
6. Гросс М. Теория формальных грамматик [Текст] / М. Гросс, А. Лантен. – М.: Мир, 1971. – 294 с. – ??? экз.
7. Рейуорд-Смит В. Дж. Теория формальных языков. Вводный курс [Текст] / В. Дж. Рейуорд-Смит. – М.: Радио и связь, 1988. – 128 с. – 30000 экз.
8. Hopcroft John E. Formal languages and their relation to automata [Текст] / John E. Hopcroft, Jeffrey D. Ullman. – Reading, Massachusetts: Addison-Wesley Publishing Company, 1969. – 242 с.
9. Ющенко К. Л. Алгоритмічні алгебри [Текст] / К. Л. Ющенко, С. В. Суржко, Г. О. Цейтлін Г., А. І. Шевченко . – К.: ІЗМН, 1997. – 480 с. – 500 екз. – ISBN 5-7763-9094-X.

10. Хопкрофт Джон Э. Введение в теорию автоматов, языков и вычислений [Текст] / Джон Э. Хопкрофт, Раджив Мотвани, Джеффри Д. Ульман. – Москва: Издательский дом "Вильямс 2002. – 528 с. – 3500 экз. – ISBN 5-8459-0261-4.
11. Яглом И. М. Булева структура и ее модели [Текст] / И. М. Яглом. – М.: Сов. радио, 1980. – 192 с. – 25000 экз.
12. Brzozowski J.A. Canonical regular expressions and minimal state graphs for definite events [Текст] / J.A. Brzozowski. – New York: Mathematical theory of automata, 1962. – с.
13. Гинзбург С. Математическая теория контекстно-свободных языков [Текст] / С. Гинзбург. – М.: Мир, 1970. – 326 с. – ??? экз.
14. Пентус А. Е. Математическая теория формальных языков: Учеб. пособие [Текст] / А. Е. Пентус, М. Р. Пентус. – М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 247 с. – ??? экз. – ISBN 5-9556-0062-0.
15. Саломаа А. Жемчужины теории формальных языков [Текст] / А. Саломаа. – М.: Мир, 1986. – 160 с. – ??? экз.
16. Мартыненко Б. К. Языки и трансляции: Учеб. пособие [Текст] / Б. К. Мартыненко. – СПб.: Издательство С.-Петербургского университета, 2004. – 231 с. – ??? экз. – ISBN 5-288-02870-2.

Покажчик термінів