Тема 38. Скінченні автомати

38.1. Скінченні автомати з виходом

Означення 38.1. **Скінченним автоматом** називають систему $M = (S, I, O, f, g, s_0)$, у якій S – скінченна **множина станів**, I – скінченна множина, яку називають **вхідним алфавітом**, O – скінченна множина, яку називають **вихідним алфавітом**, f: $S\times I \to S$ – **функція переходів**, g: $S\times I \to O$ – **функція виходів**, $s_0 \in S$ – виділений елемент, який називають **початковим станом**.

Елементи вихідного алфавіту називають **вхідними символами**, або **входами**, а вихідного – **вихідними символами**, або **виходами**. Рівність $f(s_i, x) = s_j$ означає, що в разі входу x автомат, який перебуває в стані s_i, переходить у стан s_j, а рівність $g(s_i, x) = y$ – що в цьому разі на виході з’являється y; тут $s_i, s_j \in S$, $x \in I$, $y \in O$.

Оскільки функції f і g означено на скінченних множинах, то їх можна задавати таблицями. Зазвичай дві таблиці зводять у **таблицю станів**. Вона містить значення функції переходів f і функції виходів g для всіх пар (s, x), де $s \in S$, $x \in I$.

Наприклад, у табл. 38.1 задано функції переходів і виходів для автомата з множиною станів $S = \{s_0, s_1, s_2, s_3\}$ та вхідним і вихідним алфавітами $I = \{0, 1\}$, $O = \{0, 1\}$.

Стан	f		g	
	Вхід		Вхід	
	0	1	0	1
s_0	s_1	s_0	1	0
s_1	s_3	s_0	1	1
s_2	s_1	s_2	0	1
s_3	s_2	s_1	0	0

Табл. 38.1.

Ще один поширений і наочний спосіб подання автомата – за допомогою орієнтованого мультиграфа, який називають **діаграмою станів**. Вершини графа відповідають станам; якщо $f(s_i, x_j) = s_k$ та $g(s_i, x_j) = y_r$, то з вершини s_i у вершину s_k веде дуга з позначкою x_j, y_r. Тут $x_j \in I$, $y_r \in O$. Діаграму станів для автомата, заданого табл. 38.1, показано на рис. 38.1.

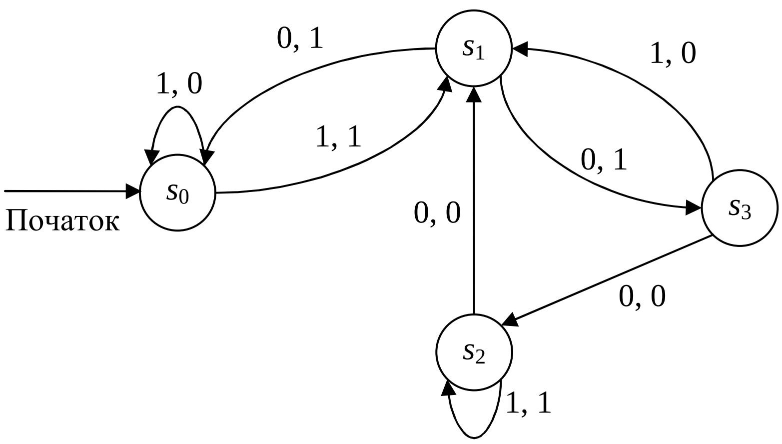

Рис. 38.1.

Для автомата M його функцію виходів g можна означити не тільки на множині I всіх вхідних символів (букв), а й на множині I^* всіх вхідних ланцюжків (слів). Розглянемо вхідний ланцюжок $\alpha = x_1x_2\ldots x_k$. Під час його обробки автомат спочатку переходить зі стану s_0 в стан s_1, де $s_1 = f(s_0, x_1)$, потім у стан s_2, де $s_2 = f(s_1, x_2)$, і цей процес триває до досягнення стану $s_k = f(s_{k-1}, x_k)$. Тут x_k – останній символ вхідного ланцюжка. Ця послідовність переходів у нові стани формує вихідний ланцюжок $\omega = y_1y_2\ldots y_k$, де $y_1 = g(s_0, x_1)$ – вихідний символ, який відповідає переходу з s_0 в s_1, $y_2 = g(s_1, x_2)$ – вихідний символ, що відповідає переходу з s_1 в s_2, і так до отримання вихідного символу $y_k = g(s_{k-1}, x_k)$. Загалом $y_j = g(s_{j-1}, x_j)$ для $j = 1, 2, ..., k$. Отже, ми можемо розширити означення функції виходів на вхідні ланцюжки так, що $g(a) = \omega$, де вихідний ланцюжок ω відповідає вхідному ланцюжку α.

Знайдемо вихідний ланцюжок, який видає скінченний автомат, зображений на рис. 38.1, якщо вхідний ланцюжок — 101001. Автомат видає на виході ланцюжок 011110. Послідовність його станів і вихідних символів наведено в табл. 38.2.

Вхід	1	0	1	0	0	1
Стан	s_0	s_0	s_1	s_0	s_1	s_3
Вихід	0	1	1	1	1	0
Новий стан	s_0	s_1	s_0	s_1	s_3	s_1

Табл. 38.2.

Означення 38.2. Відповідність, яка відображає вхідні ланцюжки α автомата M у вихідні ланцюжки ω описаним вище способом, називають **автоматним відображенням**, а також **автоматним оператором** $\boldsymbol{M}$.

Якщо результат застосування цього оператора до ланцюжка α – вихідний ланцюжок ω, то це позначають $M(\alpha) = \omega$. Кількість символів у ланцюжку α, як завжди, називають довжиною ланцюжка α та позначають $|\alpha|$ чи $l(\alpha)$.

Автоматне відображення має дві властивості.

1. Ланцюжки α та $\omega = M(\alpha)$ мають однакову довжину: $|\alpha| = |\omega|$ (збереження довжини).
2. Якщо $\alpha = \alpha_1\alpha_2$ й $M(\alpha_1\alpha_2) = \omega_1\omega_2$, де $|\alpha_1| = |\omega_1|$, то $M(\alpha_1) = \omega_1$, тобто образ відрізка довжиною l дорівнює відрізку образу з такою самою довжиною.

Властивість 2 означає, що автоматні оператори – це оператори без випередження, тобто такі, котрі, обробляючи ланцюжок зліва направо, "не підглядають уперед": i-та буква вихідного ланцюжка залежить тільки від перших i букв вхідного ланцюжка. Приклад оператора з випередженням – той, який ланцюжку $\alpha = x_1x_2\ldots x_k$ ставить у відповідність ланцюжок $x_k\ldots x_1x_2$, перша буква вихідного ланцюжка тут дорівнює останній букві вхідного ланцюжка. Зазначимо, що ці дві властивості – це не достатні умови автоматності відображення: існують відображення, які задовольняють умови 1 і 2, але не реалізовані в скінченному автоматі.

Розглянемо деякі корисні приклади скінченних автоматів, які свідчать, що стани скінченного автомата дають змогу застосовувати їх як скінченну пам'ять. Стани можна використовувати для запам'ятовування ситуацій або символів, які читає автомат. Проте через скінченну множину станів скінченні автомати не можна використовувати в деяких важливих застосуваннях.

Побудуємо скінченний автомат для додавання двох цілих додатних чисел у двійковій системі. Візьмемо числа $(x_n\ldots x_1x_0)_2$ та $(y_n\ldots y_1y_0)_2$. Додавши розряди x_0 та y_0, отримаємо розряд суми z_0 та біт перенесення c_0, який дорівнює 0 чи 1. Потім додамо розряди x_1 та y_1 і біт перенесення c_0. Одержимо розряд суми z_1 та біт перенесення c_1. Процедуру продовжуємо до стадії підсумування розрядів x_n та y_n і попереднього біта перенесення c_{n-1}; отримаємо розряд суми z_n і біт перенесення c_n, який дорівнює розряду суми z_{n+1}.

Вхідний алфавіт автомата складається з чотирьох символів: $I = \{00, 01, 10, 11\}$. Вони потрібні для подання можливих значень x_i й y_i i-го розряду обох доданків. Вихідний алфавіт $O = \{0, 1\}$, множина станів $S = \{s_0, s_1\}$. Стан s_0 відповідає ситуації, коли немає одиниці перенесення з попереднього розряду; цей стан початковий. Стан s_1 відповідає наявності одиниці перенесення з попереднього розряду. Розв'язок наведено у вигляді таблиці станів (табл. 38.3) та діаграми станів (рис. 38.2).

Стан	f				g			
	Вхід				Вхід			
	00	01	10	11	00	01	10	11
s_0	s_0	s_0	s_0	s_1	0	1	1	0
s_1	s_0	s_1	s_1	s_1	1	0	0	1

Табл. 38.3.

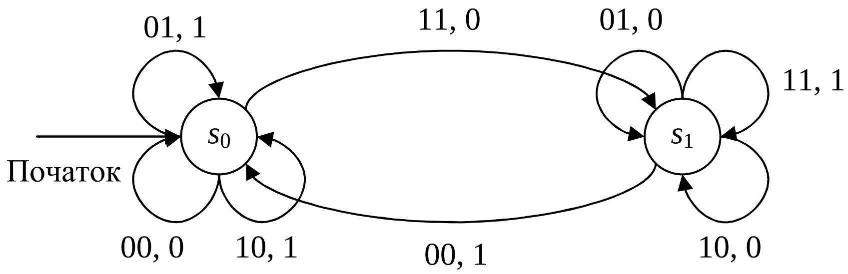

Рис. 38.2.

Розглянуті автомати називають автоматами Мілі. Уперше їх уведено 1955 р. Є також інший тип автоматів із виходом – так звані автомати Мура, запроваджені 1956 р. У цих автоматах вихід залежить лише від стану, а не від вхідного сигналу.

Автомати Мілі можна використовувати для розпізнавання мови. Проте для цього зазвичай використовують інший тип автоматів – скінченні автомати без виходу.

38.2. Скінченні автомати без виходу

Одне з найважливіших застосувань скінченних автоматів – розпізнавання (подання) мов, яке має фундаментальне значення в дослідженні й побудові компіляторів для мов програмування. Скінченні автомати без виходу мають множину заключних станів. Автомат допускає ланцюжок, якщо той переводить автомат із початкового стану в один із заключних.

Означення 38.3. **Скінченним автоматом без виходу** називають систему $M = (S, I, f, s_0, F)$, у якій S – скінченна **множина станів**, I – скінченний **вхідний алфавіт**, f: $S\times I \to S$ – **функція переходів**, означена на декартовому добутку $S\times I$, $s_0 \in S$ –**початковий стан**, $F \subset S$ – множина **заключних** (або **приймаючих**) **станів**.

Елементи вхідного алфавіту, як і раніше, називають вхідними символами чи входами.

Скінченні автомати без виходу можна задавати таблицями станів або діаграмами станів. Заключні стани на діаграмі зображають подвійними колами. Позаяк у автоматах без виходу є тільки входи (символи вхідного алфавіту I), то на дугах діаграми записують тільки їх.

Далі ми будемо розглядати лише скінченні автомати без виходу, тому називатимемо їх скінченними автоматами чи просто автоматами.

Функцію переходів f можна розширити й означити її для всіх пар станів і ланцюжків. У такому разі нехай $\alpha = x_1x_2\ldots x_k$ – ланцюжок із множини I^*. Тоді $f(s_1, \alpha)$ – стан, обчислений із використанням послідовних символів ланцюжка α зліва направо як вхідних символів, починаючи зі стану s_1. Процес відбувається так: $s_2 = f(s_1, x_1)$; $s_3 = f(s_2, x_2)$, … Вважатимемо, що $f(s_1, \alpha) = f(s_1, x_k)$.

Означення 38.4. Говорять, що скінченний автомат $M = (S, I, f, s_0, F)$ **допускає** (**приймає**) ланцюжок α, якщо він переводить початковий стан s_0 в заключний стан; це означає, що стан $f(s_0, \alpha)$ – елемент множини F.

Означення 38.5. Мова $L(M)$, яку **розпізнає автомат** M, – це множина всіх ланцюжків, які допускає автомат M. Два автомати називають **еквівалентними**, якщо вони розпізнають одну й ту саму мову.

Наприклад, знайдемо мову, яку розпізнає скінченний автомат M з діаграмою станів, зображених на рис. 38.3. Відмітимо, що дуги, які відмічені відразу двома символами (0, 1), відповідають відразу двом дугам, тобто замість двох дуг зі стану s_2 у s_2 спрямована одна, яка їх об’єднує.

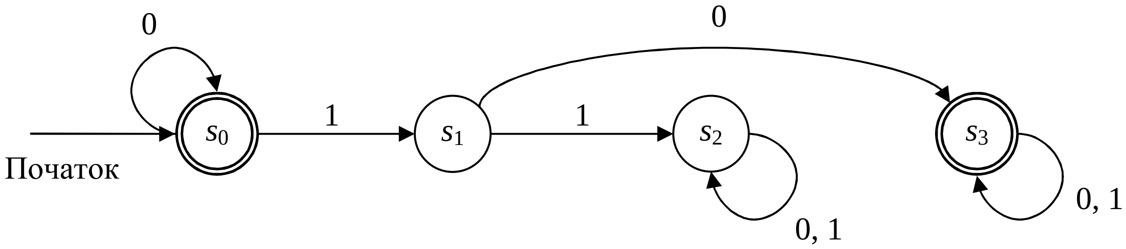

Рис. 38.3.

Заключні стани автомата M – s_0 та s_3. Стан s_0 переводить у самого себе порожній ланцюжок ε, а також ланцюжки з довільної кількість нулів 0, 00, 000, … Ланцюжки, які переводять стан s_0 у s_3, складаються з якоїсь кількості нулів, після яких є 10 і довільний ланцюжок β з нулів і одиниць. Отже, $L(M) = \{0^n, 0^n10\beta \mid n = 0, 1, 2,\ldots;\ \beta$ – довільний ланцюжок$\}$.

Розглянуті скінченні автомати без виходу називають детермінованими, бо для кожної пари “стан – вхід” існує єдиний наступний стан, заданий функцією переходів. Є й інший тип автоматів без виходу – недетерміновані. У них може бути декілька можливих наступних станів для кожної пари “стан – вхідний символ”.

<u>Означення 38.6.</u> **Недетермінованим скінченним автоматом без виходу** називають систему $M = (S, I, f, s_0, F)$, у якій S — скінченна множина станів, I – скінченний вхідний алфавіт, f –функція переходів, яка кожній парі “стан – вхід” ставить у відповідність множину станів, $s_0 \in S$ –початковий стан, $F \subset S$ –множина заключних (або приймаючих) станів.

Єдина відмінність між недетермінованим і детермінованим автоматами – тип значень функції переходів f. Для недетермінованого автомата це множина станів (вона може бути й порожньою), а для детермінованого – один стан. Недетермінований скінченний автомат задають таблицею чи діаграмою станів. У таблиці для кожної пари “стан – вхід” записують множину всіх можливих наступних станів (якщо вона порожня, то ставлять прочерк). У діаграмі переходів проводять дуги з кожного стану до всіх можливих наступних станів, на цих дугах записують входи, які спричинюють переходи одного стану в інший.

На рис. 38.4 і в табл. 38.4 подано відповідно діаграму та таблицю станів якогось недетереміннованого автомату.

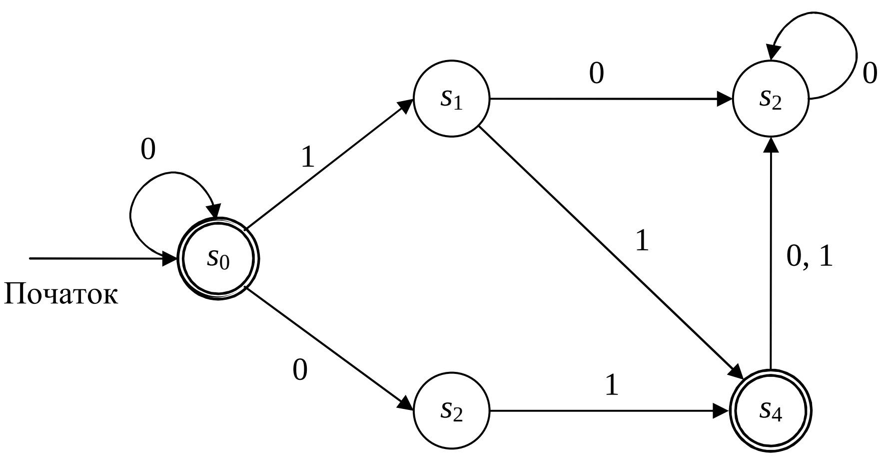

Рис. 38.4.

Стан	f	
	Вхід	
	0	1
s_0	s_0, s_2	s_1
s_1	s_3	s_4
s_2	-	s_4
s_3	s_3	-
s_4	s_3	s_3

Табл. 38.4.

Тепер визначимо, як недетермінований скінченний автомат допускає (приймає) ланцюжок $\alpha = x_1x_2\ldots x_k$. Перший вхідний символ x_1 переводить стан s_0 в множину S_1, яка може містити більше одного стану. Наступний вхідний символ x_2 переводить кожний зі станів множини S_1 у якусь множину станів, і нехай S_2 – об'єднання цих множин. Цей процес продовжують, вибираючи на кожній стадії всі стани, отримані з використанням поточного вхідного символу й усіх станів, одержаних на попередній стадії.

Означення 38.7. Недетермінований скінченний автомат **допускає** (**приймає**) ланцюжок α, якщо в множині станів, отриманій з початкового стану s_0 під дією ланцюжка α, є заключний. Мова, яку **розпізнає недетермінований скінченний автомат**, – це множина всіх ланцюжків, які він допускає.

Знайдемо мову, яку розпізнає недетермінований скінченний автомат, поданий на рис. 38.4. Оскільки s_0 – заключний стан і вхід 0 переводить його в себе, то автомат допускає ланцюжки ε, 0, 00, 000, 0000, … Стан s_4 також заключний, і нехай s_4 є в множині станів, що досягаються зі стану s_0 з ланцюжком α на вході. Тоді ланцюжок α допускається. Такими ланцюжками є 0^n01 і 0^n11, де $n = 0, 1, 2, \ldots$ Інших заключних станів немає, тому цей недетермінований автомат розпізнає таку мову: $\{0^n, 0^n01, 0^n11 \mid n = 0, 1, 2, \ldots\}$.

Теорема 38.1. Якщо мову L розпізнає недетермінований скінченний автомат M_0, то її розпізнає також детермінований скінченний автомат M_1.

Із цієї теореми випливає, що недетерміновані скінченні автомати розпізнають ті самі мови (множини ланцюжків), що й детерміновані. Проте є причини розглядати недетерміновані автомати, бо вони часто компактніші та їх легше побудувати, ніж детерміновані. Крім того, хоча недетермінований автомат завжди можна перетворити на детермінований, останній може мати експоненіціально більше станів. На щастя, таке трапляється дуже рідко.

38.3. Подання мов

Спочатку розглянемо деякі фундаментальні питання, що стосуються множин ланцюжків.

Означення 38.8. Нехай A та B – підмножини множини V^*, де V – алфавіт. **Конкатенацією множин** A та B (позначають AB) називають множину всіх ланцюжків $\alpha\beta$, де α – ланцюжок із множини A, а β – ланцюжком із множини B.

Наприклад, нехай $A = \{0, 11\}$ і $B = \{1, 10, 110\}$. Знайдемо множини AB та BA. Множина AB містить усі конкатенації ланцюжка з A та ланцюжка з B, тому

$$AB = \{01, 010, 0110, 111, 1110, 11110\}.$$

Множина BA містить усі конкатенації ланцюжка з B та ланцюжка з A:

$$BA = \{10, 111, 100, 1011, 1100, 11011\}.$$

Загалом $AB \neq BA$.

Використавши означення конкатенації двох множин ланцюжків, можна означити степінь множини ланцюжків A^n для $n = 0, 1, 2, \ldots$. Це роблять за допомогою рекурсії:

$$A^0 = \{\varepsilon\};$$

$$A^{n+1} = A^nA \text{ для } n = 0, 1, 2, \ldots$$

Нехай $A = \{1, 00\}$. Знайдемо A^n для $n = 1, 2, 3$. $A^0 = \{\varepsilon\}$, $A^1 = A^0A = \{1, 00\}$. Щоб знайти A^2, візьмемо конкатенацію всіх пар елементів з A: $A^2 = \{11, 100, 001, 000\}$. Для відшукання A^3 потрібно взяти конкатенації елементів з A^2 й A:

$$A^3 = \{111, 1100, 1001, 10000, 0011, 00100, 00001, 000000\}.$$

Означення 38.9. Нехай A – підмножина множини V*. **Замиканням Кліні** множини A (позначають A*) називають множину всіх ланцюжків, які можна утворити конкатенацією довільної кількості ланцюжків з A. Отже,

$$A^* = \bigcup_{k=0}^{\infty} A^k .$$

Скінченні автомати можна використовувати для подання (розпізнавання) мов (множин ланцюжків). Які саме множини розпізнаються ними? Цю проблему вперше розв'язав американський математик Кліні в 1956 р. Він довів, що існує скінченний автомат, який розпізнає множину ланцюжків тоді й лише тоді, коли її можна побудувати з порожньої множини (що не містить жодного ланцюжка), множини, що містить тільки порожній ланцюжок, і множин, що містять один односимвольний ланцюжок, за допомогою операцій конкатенації, об'єднання та замикання Кліні в довільному порядку. Множини ланцюжків, які можна побудувати таким способом, називають **регулярними**. Регулярний вираз над множиною I рекурсивно означають так:

- символ $\varnothing$ – регулярний вираз;
- символ ε – регулярний вираз;
- символ x – регулярний вираз, якщо $x \in I$;
- вирази (AB), $(A \cup B)$ й A^* регулярні, якщо вирази A та B регулярні.

Кожний регулярний вираз задає множину ланцюжків, визначену за такими правилами:

- $\varnothing$ – порожню множину, тобто таку, що не містить жодного ланцюжка;
- ε – множину $\{\varepsilon\}$, що містить тільки порожній ланцюжок;
- x – множину $\{x\}$, яка має один ланцюжок, що складається з одного символу x;
- (AB) – конкатенацію множин, поданих виразами A та B;
- $(A \cup B)$ – об'єднання множин, поданих виразами A та B;
- A^* – замикання Кліні множини, поданої виразом A.

Множину, задану регулярним виразом, називають **регулярною**. Відтепер регулярні вирази будемо використовувати для опису регулярних множин. Це означає таке: посилаючись на регулярну множину A, ми будемо мати на увазі регулярну множину, задану регулярним виразом A.

Нижче наведено приклад того, як регулярні вирази використовують для того, щоб задавати регулярні множин.

Покажемо, з яких ланцюжків утворено регулярні множини, задані регулярними виразами 10*, (10)*, $0 \cup 01$, $0(0 \cup 1)^*$ та $(0^*1)^*$:

Вираз	Ланцюжки, з яких складається відповідна регулярна множина
10*	1, а потім довільна кількість 0 (або без нулів)
(10)*	Довільна кількість повторень 10 (включно з порожнім ланцюжком)
$0 \cup 01$	Ланцюжок 0 і ланцюжок 01
$0(0 \cup 1)^*$	Довільний ланцюжок, який починається з 0
$(0^*1)^*$	Довільний ланцюжок, який закінчується 1

Теорема 38.2 (Кліні). Для того щоб множина була регулярною, необхідно й достатньо, щоб її розпізнав скінчений автомат.

Є тісний зв’язок між регулярними множинами та регулярними граматиками.

Теорема 38.3. Регулярна граматика породжує регулярну множину й лише її.

Доведення. Спочатку доведемо, що множина породжена регулярною граматикою, регулярна. Нехай $G = (V, T, S, P)$ – регулярна граматика, яка породжує множину $L(G)$. Доведемо, що $L(G)$ – регулярна множина. Для цього побудуємо недетермінований скінченний автомат $M = (S, I, f, s_0, F)$, який розпізнає множину $L(G)$. Множина станів S автомата M

містить стан s_A для кожного нетермінального символу A граматики G та заключний стан s_F. Початковий стан s_0 відповідає початковому символу S граматики G.

Переходи в автоматі M утворимо за допомогою продукцій граматики G таким способом. Якщо в граматиці є продукція $A \rightarrow a$, то в автоматі M має бути перехід зі стану s_A до заключного стану s_F для вхідного символу a. За наявності продукції $A \rightarrow aB$ утворюють перехід зі стану s_A до стану s_B для вхідного символу a. Множина F заключних станів автомата M містить стан s_F і, якщо $S \rightarrow \varepsilon$ – продукція граматики G, то ще й стан s_0. Тепер неважко переконатись, що мова $L(M)$, яку розпізнає побудований автомат M, збігається з мовою $L(G)$, породженою граматикою G. Отже, $L(M) = L(G)$. Це можна зробити, перевіривши ланцюжки, які переводять автомат у заключний стан. Отже, ми побудували недетермінований скінченний автомат, який розпізнає множину, породжену регулярною граматикою G. Тепер регулярність цієї множини випливає з теореми 38.2.

Доведемо, що для регулярної множини існує регулярна граматика, яка її породжує. Нехай M – недетермінований скінченний автомат, який розпізнає цю множину (він існує за теоремою 38.2). Можна вважати, що в ньому немає переходу в початковий стан s_0. Побудуємо регулярну граматику $G = (V, T, S, P)$. Кожний символ алфавіту V в граматиці G вводимо відповідно до символу стану чи вхідного символу автомата M. Множина T термінальних символів граматики G якраз і відповідає символам вхідного алфавіту I автомата M, а початковий символ S граматики G – символу початкового стану s_0 автомата M.

Множину P продукцій граматики G сформуємо на основі переходів у автоматі M. Якщо стан s у разі вхідного символу a переходить у кінцевий стан, то до множини P додають продукцію $A_s \rightarrow a$, де A_s – нетермінальний символ граматики G, що відповідає стану s. Якщо стан s переходить у стан t в разі вхідного символу a, то до множини P додають продукцію $A_s \rightarrow aA_t$. Продукцію $S \rightarrow \varepsilon$ додають до P, якщо й тільки якщо $\varepsilon \in L(M)$. Отже, продукції в граматиці G відповідають переходам в автоматі M. Тепер легко переконатись, що $L(G)=L(M)$. ►

Побудуємо недетермінований скінченний автомат для розпізнавання мови, породженої регулярною граматикою $G = (V, T, S, P)$, де $V = \{0, 1, A, S\}$, $T = \{0, 1\}$, а множина P складається з таких продукції: $S \rightarrow 0S$, $S \rightarrow 1A$, $S \rightarrow 1$, $A \rightarrow 1A$, $A \rightarrow 1$, $S \rightarrow \varepsilon$. Діаграму переходів для автомата, який розпізнає мову $L(G)$, зображено на рис. 38.5. Тут s_0 – стан, який відповідає початковому символу S граматики G, s_1 – стан, який відповідає нетерміналу A, s_2 – заключний стан.

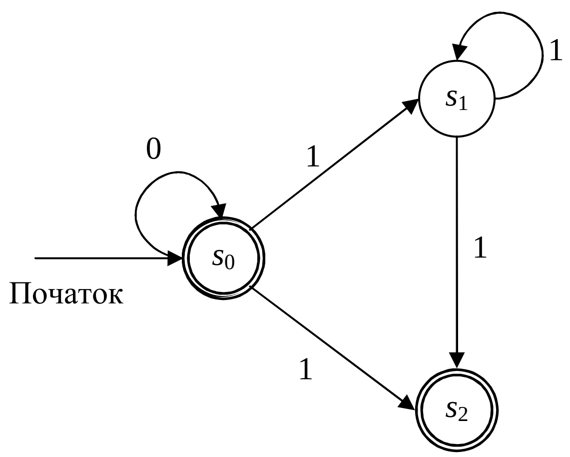

Рис. 38.5.

Розглянемо зворотній приклад. На рис. 38.6 зображено діаграму переходів скінченого автомата. Знайдемо регулярну граматику, яка породжує розпізнавану ним регулярну множину.

Граматика $G = (V, T, S, P)$ має алфавіт $V = \{S, A, B, 0, 1\}$, $T = \{0, 1\}$; нетеремінальні символ S, A та B відповідають станам s_0, s_1 та s_2 автомата; її початковий символ – S, а продукції: $S \to 0A$, $S \to 1B$, $S \to 1$, $S \to \varepsilon$, $A \to 0A$, $A \to 1B$, $A \to 1$, $B \to 0A$, $B \to 1B$ та $B \to 1$. Усі ці продукції отримано так, як це описано в другій частині доведення теореми 38.3.

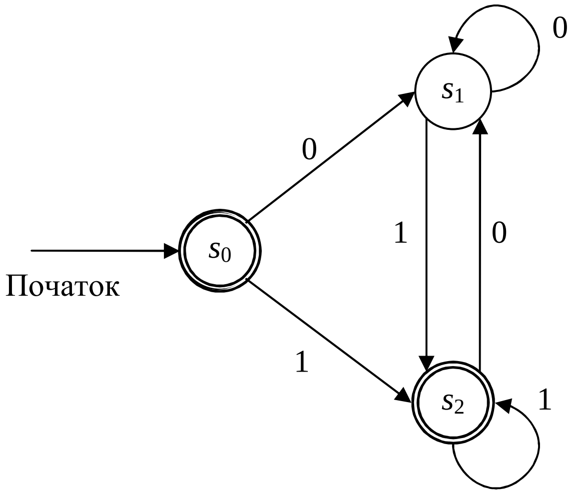

Рис. 38.6.

Із теорем 38.2 та 38.3 безпосередньо випливає таке твердження.

Теорема 38.4. Для того щоб мова була регулярною, необхідно й достатньо, щоб її розпізнав скінченний автомат.

Типова мова нескінченна, і тому немає сенсу наводити її ланцюжки та перевіряти нескінченну множину ланцюжків. Набагато розумніше використовувати один зі способів скінченного подання мови, а саме – детерміновані та недетерміновані скінченні автомати, регулярні вирази. Очевидно, що подані одним із цих способів мови регулярні.

Скінченний автомат для мов типу 3 (регулярних) – адекватна модель, а для складніших мов адекватні інші автоматні моделі, які відрізняються від скінченних автоматів нескінченністю пам'яті. Проте на цю нескінченність накладають різні обмеження залежно від типу моделі та пов'язаної з нею мови. Пам'ять може бути магазинною, тобто доступною лише з одного кінця (стек); такий автомат – адекватне подання мов типу 2. Вона може бути також лінійно обмеженою, тобто такою, що лінійно залежить від довжини розпізнаваного слова. Зокрема, такі автомати розпізнають мови типу 1. Усі названі обмеження на нескінченність пам'яті обмежують можливості цих моделей порівняно з машинами Тюрінга. Машини Тюрінга можна вважати автоматною моделлю мов типу 0, однак їх використовують переважно для уточнення поняття алгоритму.