ДМ-2

Лекція 10

Тема 8. Теорія обчислень

Скінченні автомати

План лекції

- **Скінченні автомати з виходом**
- **Скінченні автомати без виходу**
- **Мови, які розпізнаються скінченними автоматами**
- **Лема про накачування для регулярних мов**

Скінченні автомати з виходом

Скінченним автоматом називають систему $M=(S, I, O, f, g, s_0)$, у якій S, I, O – скінченні множини, а f: $S\times I\to S$ та g: $S\times I\to O$ – функції, визначені на декартовому добутку $S\times I$. Множину S називають *множиною станів*, I – *вхідним алфавітом*, O – *вихідним алфавітом*, f – *функцією переходів*, g – *функцією виходів*, виділений елемент $s_0\in S$ – *початковим станом*.

Елементи вхідного 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 та g визначено на скінченних множинах, то їх можна задати таблицями. Зазвичай дві таблиці зводять в одну й називають *таблицею станів*, або *автоматною таблицею*. Вона містить значення функції переходів f і функції виходів g для всіх пар (s, x), де $s\in S$, $x\in I$. Таблиця станів задає скінченний автомат.

Приклад 1. Табл. 1 задає функції переходів і виходів для автомата з множиною станів $S=\{s_0, s_1, s_2, s_3\}$ та вхідним і вихідним алфавітами $I = \{0, 1\}$, $O = \{0, 1\}$.

Таблиця 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

Іще один поширений і наочний спосіб задати автомат – за допомогою орієнтованого мультиграфа, який називають *діаграмою станів*. Вершини графа відповідають станам; якщо $f(s_i, x) = s_j$ та $g(s_i, x) = y$ (тут $x\in I$, $y\in O$), то з вершини s_i у вершину s_j веде дуга з позначкою x, y.

Приклад 2. Діаграму станів для автомата, заданого табл. 1, наведено на рис. 1.

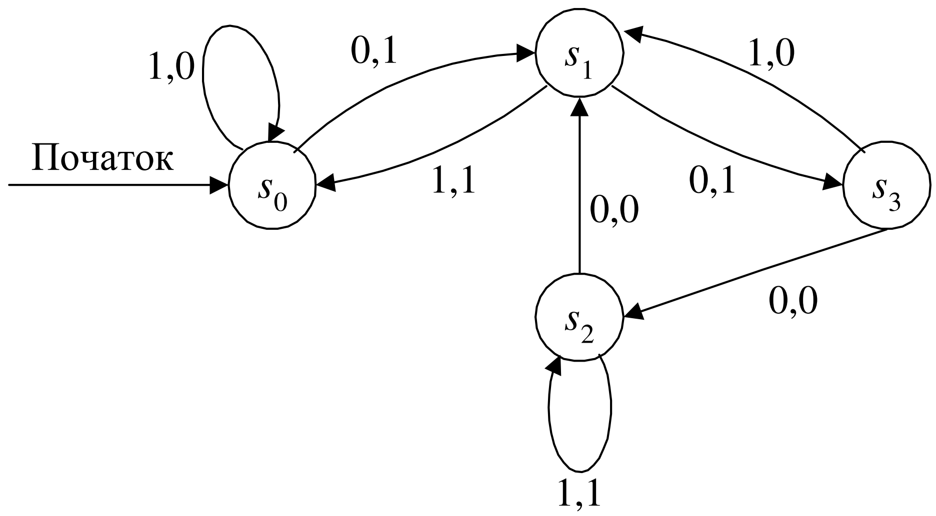

Рис. 1

Розглянемо скінченний автомат M. Кожному вхідному ланцюжку α поставимо у відповідність вихідний ланцюжок ω. Нехай вхідний ланцюжок $\alpha = x_1x_2\ldots x_k$. Тоді під час читання цього ланцюжка автомат спочатку переходить зі стану s_0 у 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 $s_k = f(s_{k-1}, x_k)$. Зазначимо, що тут x_k – останній символ вхідного ланцюжка. Ця послідовність переходів у нові стани формує вихідний ланцюжок $\omega = y_1y_2...y_k$, 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 $j = 1, 2, \ldots, k$.

Приклад 3. Знайдемо вихідний ланцюжок, який видає скінченний автомат, діаграму якого зображено на рис. 1, якщо вхідний ланцюжок – 101001.

За допомогою діаграми станів виписуємо послідовності станів і вихідних символів, їх наведено в табл. 2. Отже, автомат видає на виході ланцюжок 011110.

Таблиця 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

Розглянуту вище відповідність, яка відображає вхідні ланцюжки у вихідні, називають *автоматним відображенням*, а також *автоматною* (або *обмежено детермінованою*) *функцією*, яка *реалізується автоматом M*. Якщо результат застосування цього відображення до ланцюжка α – вихідний ланцюжок ω, то це позначають $M(\alpha)=\omega$. Кількість символів у ланцюжку α, як завжди, називають довжиною α та позначають $|\alpha|$ чи $l(\alpha)$.

Автоматне відображення має дві властивості.

1. Ланцюжки α та $\omega = M(\alpha)$ мають однакову довжину: $|\alpha|=|\omega|$ (властивість збереження довжини).

2. Якщо $\alpha=\alpha_1\alpha_2$ і $M(\alpha_1\alpha_2)=\omega_1\omega_2$, 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 відображення – це відображення *без випередження*, тобто такі, котрі, переробляючи ланцюжок зліва направо, «не підглядають уперед»: *i*-та буква вихідного ланцюжка залежить тільки від перших *i* букв вхідного ланцюжка. Приклад відображення **з випередженням** – те, яке ланцюжку $\alpha = x_1x_2...x_k$ ставить у відповідність ланцюжок $\omega = x_k...x_2x_1$; 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 реалізуються в скінченному автоматі.

Розглянемо деякі корисні приклади скінченних автоматів. Ці приклади свідчать, що стани скінченного автомата дають змогу використовувати 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.

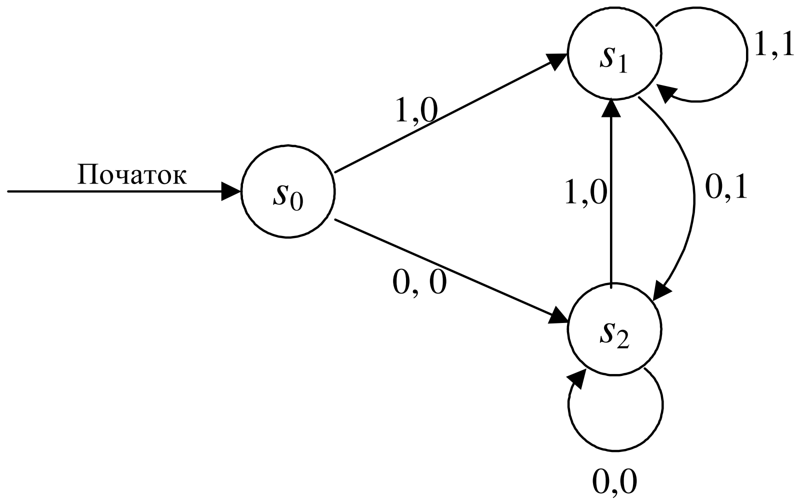

Рис. 2

Приклад 4. Важливий елемент багатьох пристроїв – автомат одиничної затримки. Він видає на виході вхідний ланцюжок, затриманий на одиницю часу. Отже, якщо на вході подано двійковий ланцюжок $x_1x_2...x_k$, то на виході буде ланцюжок $0x_1x_2...x_{k-1}$.

У такого автомата має бути два вхідні символи (нехай 0 і 1) і він має «пам'ятати», який із двох символів 0 або 1 був на вході в попередній момент. Отже, крім початкового стану s_0, потрібно ще два стани: нехай автомат перебуває в стані s_1, якщо попереднім вхідним символом була 1, і в стані s_2 – якщо 0. На виході в початковому стані завжди 0 незалежно від входу. Кожний перехід зі стану s_1 дає на виході 1, а зі стану s_2 – 0. Діаграму станів цього автомата зображено на рис. 2.

Приклад 5. Побудуємо скінченний автомат, який видає на виході 1 тоді й лише тоді, коли на вході останніми трьома символами були 1.

У цього автомата має бути три стани. Початковий стан s_0, його також використовують для запам'ятовування ситуації, коли попередній вхідний символ – 0. Стан s_1 відповідає ситуації, коли попередній символ на вході – 1, а символ перед ним – 0. Стан s_2 запам'ятовує ситуацію двох поспіль вхідних 1. Отже, якщо автомат перейшов у стан s_2, то на вхід подано дві 1 поспіль. Вхід 1 у стані s_2 означає, що це була третя поспіль 1, і на виході з'являється 1. У всіх інших ситуаціях на виході з'являється 0. Діаграму станів цього автомата наведено на рис. 3.

Автомат із прикладу 5 подає мову, оскільки він видає на виході 1 тоді й тільки тоді, коли вхідний ланцюжок (слово) має спеціальні властивості (у цьому прикладі мова складається з ланцюжків нулів і одиниць, які закінчуються трьома 1 поспіль). Подання мов – одне із найважливіших застосувань скінченних автоматів.

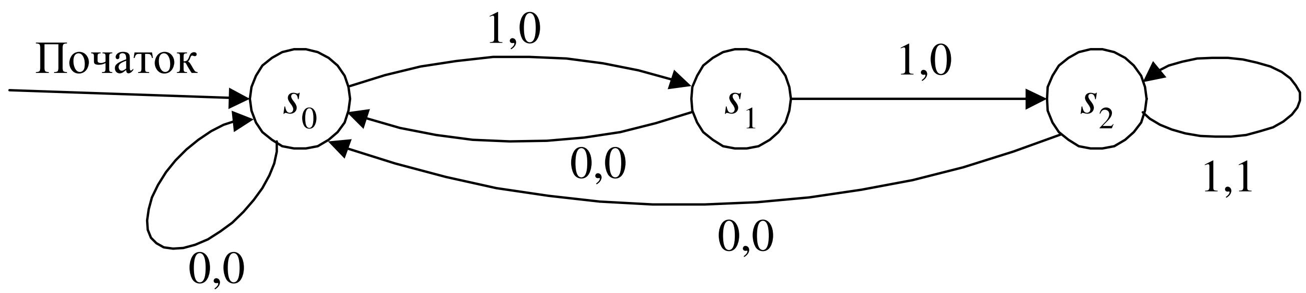

Рис. 3

Автомати, які ми розглянули, називають автоматами Мілі (G. Mealy), уперше їх уведено в 1955 р. Є також інший тип автоматів із виходом – так звані автомати Мура (E. Moore), запроваджені в 1956 р. У цих автоматах вихід визначається лише станом, тобто не залежить від вхідного сигналу.

У прикладі 5 описано, як автомат Мілі можна використати для розпізнавання мови. Проте для цього зазвичай застосовують інший тип автоматів – скінченні автомати без виходу. Такі автомати мають множину заключних (або приймаючих) станів і „приймають” ланцюжок тоді й лише тоді, коли цей ланцюжок переводить автомат без виходу з початкового стану в заключний.

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. У прикладі 4 описано, як скінченний автомат із виходом може розпізнати мову: він видає на виході 1, якщо вхідний ланцюжок належить мові, і 0 – у протилежному випадку. Проте є інший тип скінченних автоматів, спеціально призначений для розпізнавання мов. Замість виходу ці автомати мають множину заключних станів. Ланцюжок допускається автоматом, якщо він переводить автомат із початкового стану в один із заключних станів.

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*, визначена 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*.

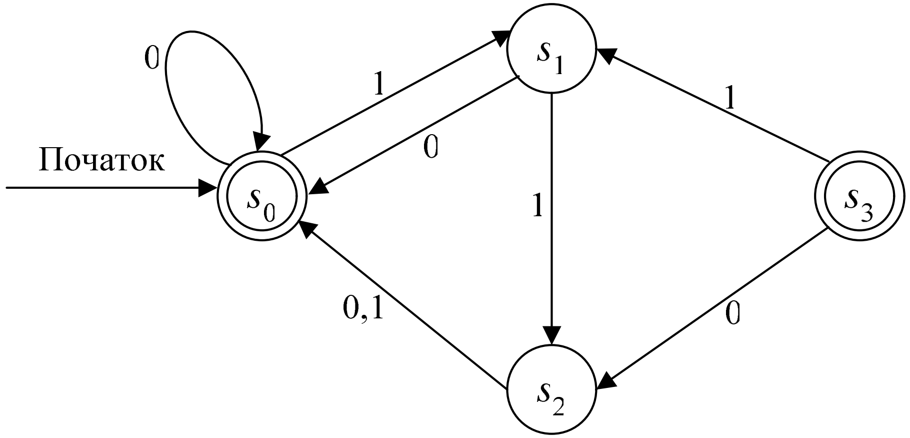

Рис. 4

Таблиця 3.

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

Скінченні автомати без виходу можна задавати таблицями станів або діаграмами станів. Заключні стани на діаграмі зображають подвійними кружечками. Зазначимо, що в автоматах без виходу є тільки входи (символи вхідного алфавіту I), тому на дугах діаграми записують тільки їх.

Оскільки далі ми будемо розглядати лише скінченні автомати без виходу, то називатимемо їх *скінченними автоматами*, чи просто *автоматами*.

Приклад 6. На рис. 4 наведено діаграму станів для скінченного автомата $M = (S, I, f, s_0, F)$, де $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0, 1\}$, $F = \{s_0, s_3\}$, а функцію переходів задано табл. 3. Оскільки обидва входи 0 і 1 переводять автомат зі стану s_2 в стан s_0, то замість двох дуг від s_2 до s_0 використано лише одну дугу, на якій написано два входи: 0 і 1.

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: $s_2 = f(s_1, x_1)$; $s_3 = f(s_2, x_2)$, … Нарешті вважаємо, що $f(s_1, \alpha)= f(s_k, x_k)$.

Говорять, що ланцюжок α *допускається* (*приймається*) скінченним автоматом $M = (S, I, O, f, s_0, F)$, якщо він переводить початковий стан s_0 у заключний стан; це означає, що стан $f(s_0, \alpha)$ – елемент множини F.

Мова, що *розпізнається* автоматом M, позначають $L(M)$, – це множина всіх ланцюжків, які допускаються автоматом M. Два автомати називають *еквівалентними*, якщо вони розпізнають одну й ту саму мову.

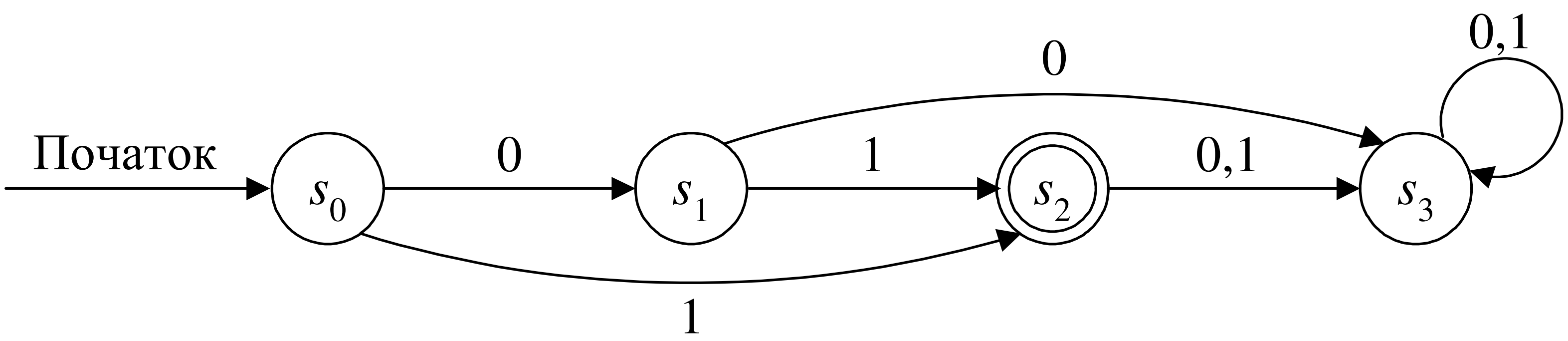

Рис. 5

Приклад 7. Знайдемо мову, яка розпізнається скінченним автоматом M_1 із діаграмою станів, зображеною на рис. 5. Автомат M_1 має тільки один заключний стан s_2. Тільки два ланцюжки переводять s_0 в s_2: 1 і 01. Отже, $L(M_1)=\{1,01\}$.

Приклад 8. Знайдемо мову, яка розпізнається скінченним автоматом M_2 із діаграмою станів, зображеною на рис. 6. Заключні стани автомата M_2 – s_0 та s_3. Стан s_0 переводять у самого себе порожній ланцюжок λ, а також ланцюжки з довільної кількості нулів: 0, 00, 000, ... Ланцюжки, які переводять стан s_0 в s_3, складаються з якоїсь кількості нулів, після яких є 10 і довільний ланцюжок β з нулів та одиниць. Отже,

$$L(M_2) = \{0^n, 0^n10\beta \mid n=0,1,2,\ldots;\ \beta \text{ – довільний ланцюжок}\}.$$

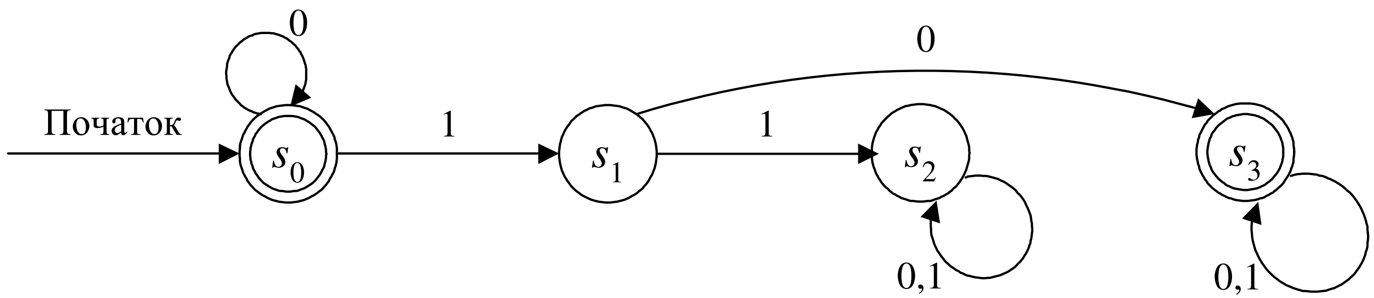

Рис. 6

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 – це недетерміновані автомати. У них може бути декілька можливих наступних станів для кожної пари «стан – вхід».

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*.

Зазначимо, що 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 або 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.

Приклад 9. На рис. 7 і в табл. 4 подано відповідно діаграму та таблицю станів недетермінованого автомата.

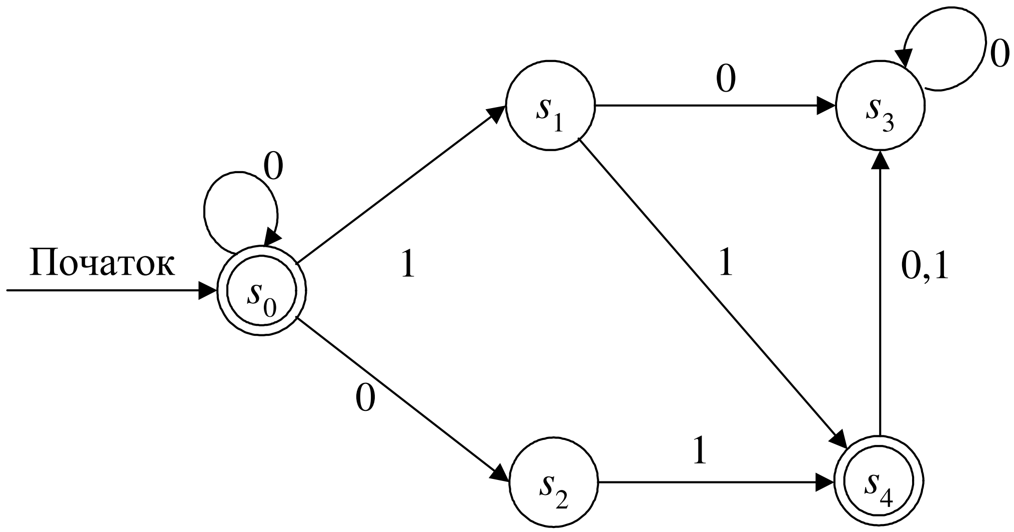

Рис. 7

Таблиця 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

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 кожному кроці всі стани, отримані з використанням поточного вхідного символу й усіх станів, одержаних на попередньому кроці. Ланцюжок α *допускається* (*приймається*) недетермінованим скінченним автоматом, якщо в множині станів, отриманій з початкового стану s_0 під дією ланцюжка α, є заключний стан.

Мова, що *розпізнається* недетермінованим скінченним автоматом – це множина всіх ланцюжків, які допускаються автоматом M.

Приклад 10. Знайдемо мову, яка розпізнається недетермінованим скінченим автоматом із рис. 7.

Оскільки s_0 – заключний стан і вхід 0 переводить його в себе, то ланцюжки λ, 0, 00, 000, 0000, ... допускаються цим автоматом. Стан s_4 також заключний, і нехай s_4 є в множині станів, що досягаються зі стану s_0 із ланцюжком α на вході. Тоді ланцюжок α допускається. Такими ланцюжками є 0^n01 та 0^n11, де n=0,1,2,… . Інших заключних станів немає, тому мова, яка розпізнається цим недетермінованим автоматом, така: $\{0^n, 0^n01, 0^n11 | n=0, 1, 2, \ldots\}$.

Зазначимо, що коли мова розпізнається недетермінованим автоматом, то вона також розпізнається детермінованим автоматом.

Теорема 1. Якщо мова L розпізнається недетермінованим скінченним автоматом M_0, то L розпізнається також детермінованим скінченним автоматом M_1.

Мови, які розпізнаються скінченними автоматами

Сформулюємо без доведення важливий результат.

Теорема. Для того щоб мова була регулярною, необхідно й достатньо, щоб вона розпізнавалась скінченним автоматом.

З'ясуємо, що означає задати питання щодо деякої мови. Зазначимо, що типова мова нескінченна, і тому немає сенсу наводити комусь ланцюжки цієї мови й задавати питання, яке потребує перевірки нескінченної множини ланцюжків. Набагато розумніше використовувати один зі способів скінченного подання мови, а саме – детерміновані скінченні автомати, недетерміновані скінченні автомати.. Очевидно, що подані одним із цих способів мови регулярні.

Отже, для мов типу 3 (регулярних) скінченний автомат – адекватна модель. Для складніших мов адекватні інші автоматні моделі, які відрізняються від скінченних автоматів **нескінченністю** пам'яті. Проте на цю нескінченність накладаються 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. Пам'ять може бути *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.

Лема про накачування для регулярних мов

Розглянемо інструмент, для доведення **нерегулярності** деяких мов – *лему про накачування* (pumping lemma). Інша назва – *лема про розростання*.

Лема про накачування для регулярних мов. Нехай L – регулярна мова. Існує константа n (залежна від L) така, що кожний ланцюжок $\alpha\in L$, який задовольняє нерівність $|\alpha|\geq n$, можна розбити на три ланцюжки $\alpha = \beta\gamma\omega$ так, що виконуються умови:

1) $\gamma \neq \lambda$;

2) $|\beta\gamma| \leq n$;

3) для довільного $k \geq 0$ ланцюжок $\beta\gamma^k\omega\in L$.

Це означає, що завжди можна знайти такий ланцюжок γ недалеко від початку ланцюжка α, котрий можна «накачати». Отже, якщо ланцюжок γ повторити довільну кількість разів або вилучити (k=0), то одержаний ланцюжок буде належати мові L.

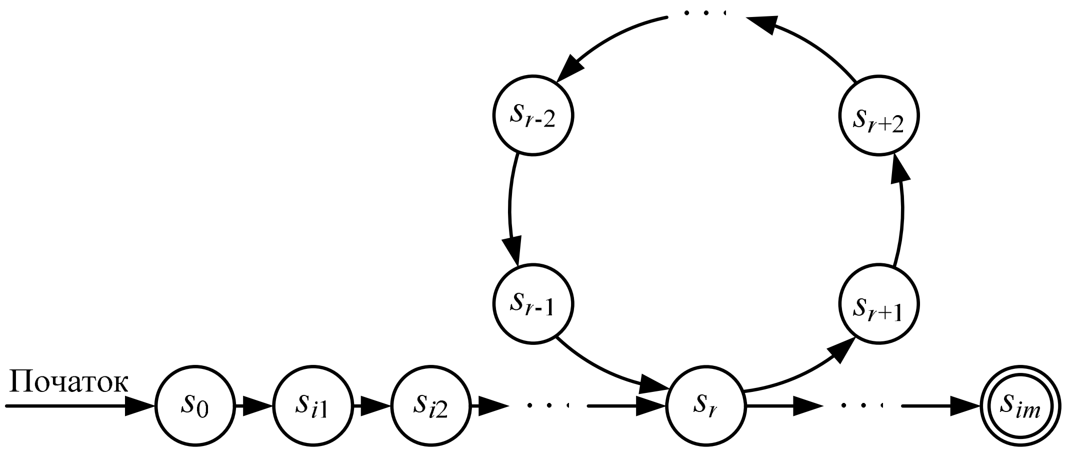

Рис. 8

Доведення. Нехай L – регулярна мова. Тоді існує детермінований скінченний автомат M, для якого $L = L(M)$, тобто що розпізнає цю мову. Нехай кількість станів автомата M дорівнює n (константа, яка фігурує у формулюванні леми). Припустимо, що послідовні стани, у які автомат M переходить під дією вхідного ланцюжка α, такі:

$$s_0,\ s_{i_1},\ s_{i_2},\ \ldots,\ s_{i_m},$$

де $m = |\alpha|$ – довжина ланцюжка α. За умовою леми, $m \geq n$, тому в списку $s_0,\ s_{i_1},\ s_{i_2},\ \ldots,\ s_{i_m}$, який складається принаймні з n+1 стану, обов'язково є повторення (це випливає з принципу коробок Діріхле). Нехай s_r – перший повторюваний стан. Позначимо як γ ту частину ланцюжка α, що переводить автомат зі стану s_r, коли він зустрічається вперше, до стану s_r, коли він зустрічається вдруге. Позначимо як β частину ланцюжка α перед γ, а як ω – частину ланцюжка α після γ. Очевидно, що $|\gamma| \geq 1$ (отже, $\gamma \neq \lambda$) та $|\beta\gamma| \leq n$ (оскільки всі стани до другої появи s_r, різні). Більше того, ланцюжок $\beta\gamma^k\omega$ для будь-якого цілого невід'ємного числа k має переводити автомат у той самий заключний стан, що й ланцюжок $\beta\gamma\omega$. Справді, частина γ^k ланцюжка $\beta\gamma^k\omega$ просто переводить стани автомата вздовж петлі, яка починається та закінчується в стані s_r. Ця петля проходиться k разів (рис. 8). Звідси випливає, що ланцюжки $\beta\gamma^k\omega$ допускаються автоматом тому, що ланцюжок $\beta\gamma\omega$ допускається; отже, усі вони належать мові $L(M)$. Доведення леми завершено.

Приклад 11. Доведемо, що мова $L = \left\{ 0^m 1^m \middle| m = 0,1,2,\ldots \right\}$ нерегулярна.

Від протилежного. Припустимо, що мова L регулярна. Нехай ланцюжок $\alpha = 0^m 1^m$ і $|\alpha| = 2m \geq n$. Тоді за лемою про накачування ланцюжки $\alpha = 0^m 1^m = \beta\gamma\omega$ та $\beta\gamma^k\omega$ належать мові L, причому $\gamma \neq \lambda$. Тепер розглянемо такі міркування. Ланцюжок γ не може містити водночас нулі й одиниці, бо ланцюжок γ^2 тоді містив би 10, і ланцюжок $\beta\gamma^2\omega$ не належав би мові L. Отже, ланцюжок γ складається або тільки з нулів, або тільки з одиниць. Але тоді ланцюжок $\beta\gamma^2\omega$ містить або забагато нулів, або забагато одиниць. Звідси випливає, що ланцюжок $\beta\gamma^2\omega$ не належить L. Ця суперечність доводить, що мова $L = \left\{ 0^m 1^m \middle| m = 0,1,2,\ldots \right\}$ нерегулярна.