ЛЕКЦІЯ 11. АЛГОРИТМІЧНА СИСТЕМА ТЬЮРІНГА

Точний опис класу частково-рекурсивних функцій разом з тезою Черча дає один з можливих розв'язків задач про уточнення поняття алгоритму. Однак цей розв'язок не зовсім прямий, оскільки поняття обчислюваної функції є вторинним щодо інтуїтивного поняття алгоритму. Постає запитання: чи можна уточнити саме поняття алгоритму, а потім за його допомогою визначити клас обчислюваних функцій?

Відповідь на це питання дали 1936-1937 рр. Е. Пост і А. Тьюрінг та майже одночасно й незалежно А. Черч і С.-К. Кліні.

Головна думка праць Е. Поста і А. Тьюрінга полягає в тому, що алгоритмічні процеси - це процеси, які може виконувати відповідно побудована "машина". Відповідно до цього за допомогою точних математичних термінів вони описали досить вузькі класи машин, на яких можливо реалізувати або імітувати всі алгоритмічні процеси, які фактично будь-коли були описані математиками. Алгоритми, які можна реалізувати на машинах Тюрінга-Поста, запропоновано розглядати як математичний еквівалент алгоритмів у інтуїтивному понятті.

Отже, машина Тьюрінга - це математична модель пристрою, який породжує обчислювальні процеси. Її використовують для теоретичного уточнення поняття алгоритму та його дослідження.

Машина Тьюрінга

На змістовному рівні машина Тьюрінга (МТ) є деякою гіпотетичною (умовною) машиною, яка складається з трьох головних компонент: *інформаційної стрічки, головки для зчитування і запису та пристрою керування* (рис. 11.1).

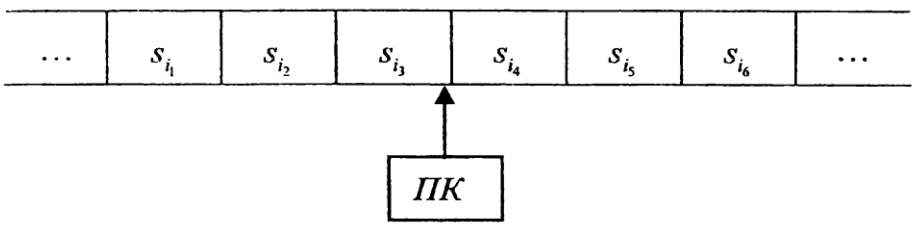

Рис. 11.1

Інформаційна стрічка призначена для записування вхідної, вихідної і проміжної інформації, яка виникає внаслідок обчислень. Стрічка потенційно безмежна в обидва боки і розбита на окремі не пронумеровані комірки (позиції, клітинки), в кожну з яких можна помістити один символ алфавіту $S = \{s_1, s_2, \ldots, s_k\}$, фіксованого для кожної МТ. Цей алфавіт називають ***зовнішнім алфавітом машини***.

Головка зчитування і запису - це спеціальний чуттєвий елемент, здатний читати і змінювати вміст комірок стрічки, зміщуючись уздовж неї вправо і вліво.

Рухається головка так, що в кожен момент часу *t* вона стоїть навпроти однієї певної комірки, і вважають, що в цей момент машина "сприймає" символ, записаний у цій позиції.

Пристрій керування (*функціональний механізм, логічний блок*) керує всією роботою машини відповідно до заданої програми обчислень. У кожний момент часу *t* функціональний механізм перебуває в одному зі станів, множина яких $Q = \{q_1, q_2, \ldots, q_n\}$ зафіксована для кожної МТ, її називають ***внутрішнім алфавітом машини***. В множині *Q* виділяють два особливі стани: *початковий* (q_0) і *заключний* (q_f), що відповідають за пуск і зупинку машини.

Функціонування МТ відбувається дискретними кроками. Кожен крок охоплює три елементарні операції:

1. символ s_i, на який вказує головка в цей момент, заміняється іншим символом алфавіту *S*:

$$s_i \to s_j \ (i.j = \overline{1,k}) ;$$

2. головка зсувається на одну позицію вліво (L) чи вправо (R) або ж залишається на місці (нейтральний зсув H);
3. функціональний механізм змінює свій стан q_l, на новий стан з алфавіту *Q*:

$$q_l \to q_r \ (l, r = \overline{1,n}) .$$

Машину Тьюрінга можна використовувати для *розпізнавання мов*. Символи на стрічці такої машини містять алфавіт мови, яку розглядають (її

букви відіграють роль вхідних символів), порожній системи символ і, можливо, інші символи. Спочатку на стрічці записано слово з вхідних символів (по одному символу в комірці), починаючи з першої ліворуч комірки. Всі комірки праворуч від комірок, що містять вхідне слово, порожні.

Слово з вхідних символів ***допустиме*** тоді й тільки тоді, коли машина Тьюрінга, почавши роботу у виділеному початковому стані, зробить послідовність кроків, які приведуть її в допускаючий (заключний) стан.

Мовою, яку ***розпізнає*** ця машина Тьюрінга, називають множину всіх слів із вхідних символів, які допустимі цією МТ.

Роботу машини Тьюрінга формально можна описати за допомогою конфігурацій. Під ***конфігурацією*** МТ в момент часу t (позначимо K_t,) будемо розуміти таку сукупність умов:

- конкретне заповнення комірок стрічки в момент t буквами алфавіту S;
- конкретне положення головки на стрічці в момент часу t;
- стан функціонального механізму в цей момент.

Тоді процес роботи МТ полягатиме в послідовній зміні конфігурацій, починаючи від початкової конфігурації K_{t_0}, яка відповідає стану q_0 в початковий момент часу t_0. Якщо внаслідок роботи МТ породжує процес

$$MT\left(K_{t_0}\right) = K_{t_0}, K_{t_1}, K_{t_2}, \ldots, K_{t_m}, K_{t_{m+1}}, \ldots, K_{t_F},$$

який завершується заключною конфігурацією K_{t_F}, (яка відповідає заключному стану q_F), то така МТ ***застосовна*** до початкової конфігурації K_{t_0}. Результатом роботи машини в такому разі вважають слово, яке буде записане на стрічці в заключній конфігурації з внутрішнім станом q_F.

Якщо ж заключна конфігурації K_{t_F} не досяжна, то МТ працює безмежно довго без зупинки (зациклюється). Вважають, що така МТ ***незастосовна*** до початкової конфігурації K_{t_0}.

Перехід МТ від конфігурації K_{t_m} до наступної конфігурації $K_{t_{m+1}}$ виконується за один крок, ініціалізований відповідною командою. ***Командою*** МТ називають таку п'ятірку:

$$\langle s_i, q_l \rangle \to \langle s_j, q_r, Z \rangle,$$

де Z - одна з дій (зсув уліво, вправо чи на місці - $\{\text{L,R,H}\}$. Команда означає заміну s_i, на s_j, q_l на q_r і зсув головки згідно з Z.

Жодні дві команди *не можуть мати однакових лівих частин*.

Стан q_F не може траплятися в лівій частині команди.

Множину команд, що визначає процес обчислень на МТ, називають ***програмою МТ***. Оскільки для кожної пари $\langle s_i, q_l \rangle \left(i = \overline{1,k}; l = \overline{1,n}\right)$ програма може містити відповідну п'ятірку, то розмір програми не перевищуватиме $k \times n$ п'ятірок.

Програму можна задавати у вигляді таблиці, яку називають ***функціональною схемою***, рядки і стовпці цієї таблиці позначені, відповідно, символами внутрішнього і зовнішнього алфавіту, а на перетині q_l рядка та s_i стовпця є трійка s_j, q_r, Z. Якщо один або декілька елементів залишаються незмінними, то їх не зазначають.

Зазначимо, що, крім інтерпретації МТ як пристрою для розпізнавання мов, її можна розглядати і як пристрій, який обчислює деяку функцію *f*. Аргументи цієї функції кодовані на вхідній стрічці у вигляді слова *x* зі спеціальним маркером '*', який відділяє їх один від одного. Якщо МТ зупиняється, а на стрічці є ціле число *y* (значення функції), то приймають $f(x) = y$. Отже, процес обчислення мало відрізняється від процесу розпізнавання мови.

Приклад. Побудуємо МТ для додавання двох натуральних чисел, записаних в унарному вигляді. На стрічці МТ задають два числа як послідовності відповідної кількості символів '1', розділених символом '*'. Програму для машини Тьюрінга задамо у вигляді такої таблиці:

T	1	Λ	*
q_0	Λq_2R	R	Λq_F
q_1	L	q_0R	L

q_2	R	$1q_1$L	R

Множина станів МТ $Q=\{q_0,q_1,q_2,q_F\}$, зовнішній алфавіт $S=\{1,\Lambda,*\}$, де Λ - порожній символ.

Ця програма почергово заміняє символи '1' першого доданка, починаючи з лівого краю, на порожні символи і дописує відповідну кількість '1' після другого доданка. Такі дії відбуваються аж до роздільника '*', який також заміняється символом Λ.

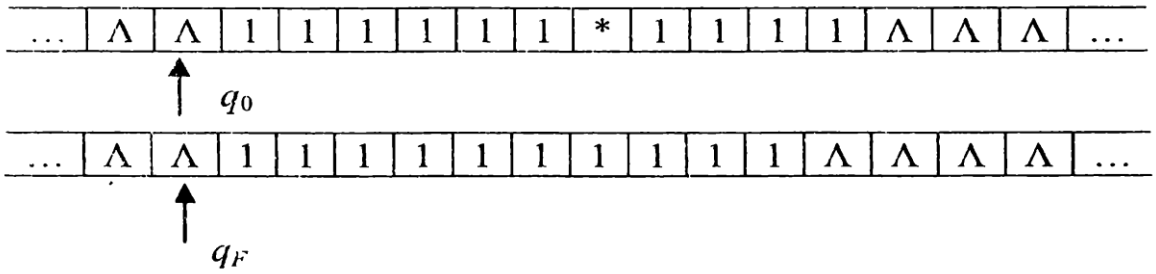

Рис. 11.2

На рис. 11.2 показано процес переробки початкової конфігурації в заключну у разі реалізації 6 + 4.

Можна побудувати й іншу програму МТ для розв'язування цієї задачі:

T	1	Λ	*
q_0	Λq_1R	R	-
q_1	R	-	$1q_2$
q_2	L	q_F	-

Ця програма перший символ '1' першого доданка заміняє на порожній символ Λ, а роздільник '*' – на '1'.

Нагадаємо, що головними параметрами, які оцінюють складність алгоритму, є часова та ємнісна складність.

Часова складність $T(n)$ машини Тьюрінга дорівнює найбільшій кількості кроків, зроблених нею під час опрацювання входу довжиною n (для всіх входів довжиною n). Якщо на деякому вході довжиною n МТ не зупиняється, то часова складність на цьому вході не визначсна.

Ємнісна складність $S(n)$ машини Тьюрінга дорівнює максимальній відстані, яку потрібно пройти головці під час опрацювання входу довжини *n*. Якщо головка для деякого входу довжини *n* невизначено довго рухається, то $S(n)$ на цьому вході не визначена.

Формальне визначення МТ. Теза Тьюрінга

З математичного погляду МТ - це певний алгоритм для переробки вхідних слів. Оскільки спосіб переробки машинних слів відомий, якщо відома програма машини, то МТ вважають заданою, якщо задані її зовнішній і внутрішній алфавіти, програма та визначено, які з символів відповідають порожній комірці й заключному стану.

Зазначимо, що кожна МТ реалізує таку функцію:

$$\delta : S \times \{Q \setminus q_F\} \to S \times Q \times Z,$$

або інакше

$$\delta = \{\langle s_i, q_l \rangle \to \langle s_j, q_r, Z \rangle\}, q_l \neq q_F.$$

Тоді формально машиною Тьюрінга називають сімку

$$(S, Q, q_0, q_F, I, \Lambda, \delta),$$

де $I \subseteq S$ - множина вхідних символів; Λ - порожній символ.

Якщо визначити строго формально МТ як математичний об'єкт, то можна на формальному рівні визначити поняття конфігурації, процесу обчислення і всі інші поняття, які використовують у доведеннях різних тверджень про процеси переробки інформації на МТ. Проте в теорії алгоритмів застосовують змістовний опис МТ і доведення виконують на змістовному рівні, вважаючи, що за ними легко відновити формальний еквівалент МТ.

Говорять, що слово *P* задане в ***стандартному положені***, якщо головка фіксує перший зліва символ вхідного слова. Якщо за початкову конфігурацію K_{t_0}, взяти стандартне положення слова *P*, то процес $MT(K_{t_0})$ є *процесом переробки вхідного слова P*, і його позначають *MT*(*P*).

Машина Тьюрінга ***обчислює*** *деяку словникову функцію* $\varphi(P)$, *яка відображає слова з деякої множини слів у значущі (не порожні) слова, якщо для довільного слова P процес MT(P) досягає заключної стандартної конфігурації, асоційованої зі словом* $\varphi(P)$ *- результатом цієї функції.*

Якщо для часткової словникової функції $\varphi(P)$ існує МТ, яка її обчислює, то функцію $\varphi(P)$ називають ***обчислюваною за Тьюрінгом***.

Якщо функція f є m-місною, то вважають, що обчислюється функція від одного аргументу, який є кодом слів $P_1, P_2, \ldots, P_m$. Наприклад, $f(P_1, P_2, \ldots, P_m) = f(P_1 * P_2 * \ldots * P_m)$, де символ $* \notin S$.

Теза Тьюрінга. *Для довільного алгоритму* $A = \langle \varphi, P \rangle$ *у довільному скінченному алфавіті X існує функція* $\varphi(P): \{P\}_X \to \{Q\}_X$, *обчислювана за Тьюрінгом.*

Теза Тюрінга визначає відповідність між інтуїтивним поняттям алгоритму і точним математичним поняттям функції, обчислюваної на МТ: *будь-який алгоритм, заданий у довільній формі, можна замінити еквівалентною йому МТ.*

Згідно з цією тезою, питання про можливість алгоритмізації того чи іншого процесу рівносильне питанню про можливість реалізації цього процесу на МТ.

Як і принцип нормалізації, теза Тьюрінга стверджує неможливість побудови в майбутньому алгоритму, який не можна було б реалізувати на МТ.

Тезу Тьюрінга, як і тези Маркова і Черча, довести неможливо, оскільки в її формулюванні використане інтуїтивне поняття алгоритму.

Тюрінг довів такі важливі теореми.

Теорема 1. *Всі часткові словникові функції, обчислювані за Тьюрінгом, є частково-рекурсивними.*

Теорема 2. *Для кожної частково-рекурсивної словникової функції, визначеної в деякому алфавіті* $A = \{a_1, \ldots, a_m\}$, *існує МТ з відповідним*

внутрішнім алфавітом, зовнішній алфавіт якої збігається з A (S = A), і яка обчислює задану функцію.

Теореми Тьюрінга і засвідчують еквівалентність трьох алгоритмічних систем - нормальних алгоритмів, частково-рекурсивних функцій і машин Тьюрінга.

Універсальна машина Тьюрінга

Дотепер ми розглядали різні алгоритми, виконувані на різних МТ, що відрізняються програмами, внутрішніми і зовнішніми алфавітами. За аналогією з універсальним нормальним алгоритмом можна побудувати універсальну машину Тьюрінга (УМТ), яка здатна виконувати роботу будь-якої конкретної МТ $T^{(k)}$ за умови, що УМТ отримує інформацію про програму П машини $T^{(k)}$ і вхідне слово *P*, подане на обробку в $T^{(k)}$.

В УМТ, як і в будь-якій МТ, інформація задана на стрічці. У цьому разі УМТ має фіксований скінченний. зовнішній алфавіт S_U, який називають *стандартним*. Проте УМТ повинна бути пристосована до сприйняття програм і конфігурацій довільних $T^{(k)}$ і довільних вхідних слів, у яких можуть траплятися букви з різноманітних алфавітів і з як завгодно великою кількістю різних букв.

Це досягають шляхом кодування в стандартному алфавіті S_U $T^{(k)} = \left\langle S^{(k)}, Q^{(k)}, q_0, q_F, \Pi^{(k)} \right\rangle$. Кодування можна виконати, як і у випадку УНА, при $S_U = \{0,1\}$: всі букви нумерують натуральними числами, після чого *i*-й букві ставлять у відповідність двійковий код $0\underbrace{11\ldots1}_{i}0$.

Для кожної машини $T^{(k)}$ повинні бути закодовані всі символи на $S^{(k)} \cup Q^{(k)} \cup \{\mathrm{L},\mathrm{R},\mathrm{H}\} \cup \{\Delta,*\}$, де символ Δ використовують для відділення один від одного команд програми $\left\langle s_i, q_l \right\rangle \to \left\langle s_j, q_r, Z \right\rangle$, а символ '*' - для відділення лівої частини команди від правої. Якщо $S^{(k)}$ має *n* символів, а $Q^{(k)}$ - *m* символів, то всього повинно бути закодовано $m \times n + 5$ символів.

Якщо записати Π^{T_k} одним словом і кодувати його букви, то отримаємо слово в алфавіті S_U, яке називають ***кодом*** Π^{T_k}; (позначають $\mathrm{cod}\Pi^{T_k}$).

А. Тьюрінг довів теорему про існування універсальної МТ.

***Теорема**. Існує така МТ, її називають універсальною, яка для довільної машини Тьюрінга $T^{(k)}$ і довільного вхідного слова P перетворює вхідне слово* $\mathrm{cod}\Pi^{T_k}\mathrm{cod}P$*, отримане конкатенацією кодів* Π^{T_k} *і слова P, у код слова* $T^{(k)}(P)$*:*

$$\mathrm{cod}\Pi^{T_k}\mathrm{cod}P \xrightarrow{УМТ} codT^{(k)}(P).$$

Якщо $T^{(k)}$ *до слова P не застосовна, то УМТ теж є незастосовною до слова* $\mathrm{cod}\Pi^{T_k}\mathrm{cod}P$.

З огляду на існування УМТ будь-яку програму Π^{T_k} можна розглядати двояко:

- як програму деякої конкретної МТ,
- як вхідну інформацію для УМТ, що виконує роботу T_k.

Із теореми Тьюрінга випливає принципова можливість побудови машини, на якій можна реалізувати довільний алгоритм $A=\langle\varphi,P\rangle$. Водночас програмування для машини, яка моделює УМТ, є нере- альним через його громіздкість. Тому на практиці універсальні машини будують на підставі інших алгоритмічних систем.

Різновиди машин Тьюрінга

Ми розглянули однострічкому одноголовкову МТ з одним функціональним пристроєм, яку називають ***класичною машиною Тьюрінга***. Далі опишемо декілька варіантів МТ, які відрізняються один від одного кількістю зчитувальних головок, видом і кількістю стрічок, а також розглянемо операції композиції, застосовні до них.

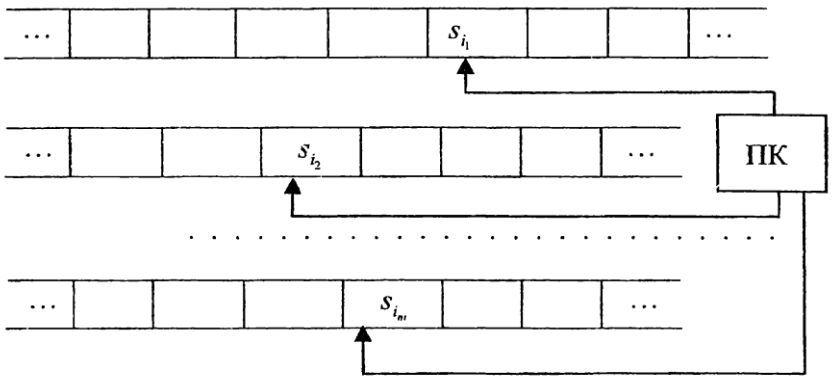

Рис. 11.3

Багатострічкова машина - це МТ зі скінченною кількістю стрічок (рис. 11.3). Її визначення можна розглядати на змістовному і формальному рівнях. На змістовному рівні багатострічкова МТ складається з декількох, наприклад *m*, стрічок, кожна з яких має свою головку, що працює незалежно від інших. У кожен момент часу *t* машина спостерігає *m* комірок, по одній на кожній стрічці. Крок машини охоплює три елементарні операції:

1. заміна деяких або і всіх *m* символів комірок новими символами,
2. зсув кожної з *m* головок, або деяких з них, вліво, вправо або нейтральний зсув незалежно одна від одної,
3. перехід функціонального механізму в новий стан.

Формально *m*-стрічкова машина Тьюрінга задана сімкою:

$$(Q, S, I, \delta, \Lambda, q_0, q_F),$$

де Q - множина станів; S - множина символів на стрічках; I - множина вхідних символів ($I \subseteq S$); Λ - порожній символ ($\Lambda \in S - I$); q_0 - початковий стан; q_F - заключний стан. Функція переходів δ відображає деяку підмножину множини $Q \setminus \{q_F\} \times S^k$ в $Q \times \left(S \times \{\mathrm{L,R,H}\}\right)^k$, тобто за довільним набором станів та k символів на стрічках вона видає новий стан і k пар, кожна з яких складається з нового символу на стрічці і напряму зсуву головки.

Відповідно до цього команди складаються з $3m+2$ символів і мають такий вигляд:

$$\left\langle s_{i_1}, s_{i_2}, \ldots, s_{i_m}, q_l \right\rangle \to \left\langle s_{j_1}, s_{j_2}, \ldots, s_{j_m}, q_r, Z_1, \ldots, Z_m \right\rangle.$$

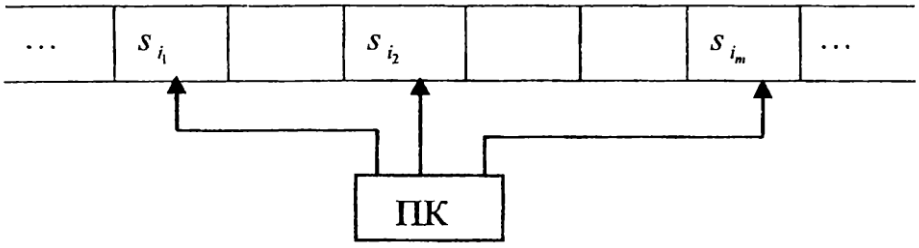

Рис. 11.4.

Багатоголовкова машина - це однострічкова МТ з m головками (рис. 11.4). У кожен момент часу t машина спостерігає m комірок. Крок машини містить три елементарні операції, аналогічні операціям багатострічкової машини, а вигляд команди повністю збігається з виглядом для багатострічкової МТ. Тому за виглядом програми для багатострічкової та багатоголовкової МТ не відрізняються.

У цьому варіанті МТ можливий випадок, коли декілька головок спостерігають одну й ту ж комірку, проте команда передбачає різні записи в цю комірку.

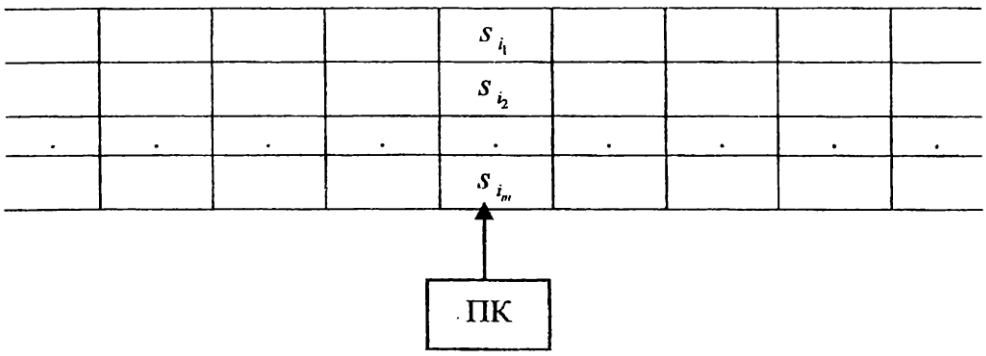

Рис. 11.5

Наприклад, для двоголовкової машини це може бути така команда: $\langle s_{i_1}, s_{i_2}, q_l \rangle \to \langle s_{j_1}, s_{j_2}, q_r, Z_1, Z_2 \rangle$, де $s_{j_1} \neq s_{j_2}$. Для усунення такої неоднозначності вважають, що в комірці зберігається лише той запис, який заданий головкою з найменшим номером (s_{j_1}). Аналогічна домовленість прийнята 1 в загальному випадку - у разі визначення функціонування багатоголовкової машини з кількістю стрічок, меншою від кількості головок.

Машина з багатоповерховою стрічкою - це однострічкова, одноголовкова МТ (рис. 11.5). Стрічка має m поверхів і в кожний момент часу t машина оглядає відразу m символів, які розташовані на поверхах комірки, що є активною в цей момент.

Крок машини полягає в заміні m символів і стану, а також зсуві головки, а команда має такий вигляд:

$$\langle s_{i_1}, s_{i_2}, \ldots, s_{i_m}, q_l \rangle \to \langle s_{j_1}, s_{j_2}, \ldots, s_{j_m}, q_r, Z \rangle.$$

Одноголовкова машина з однією двовимірною стрічкою. Двовимірна стрічка - це площина, вистелена комірками. Для кожної комірки є сусідні праворуч, ліворуч, знизу і зверху, відповідно до чого і зсуви головок будуть чотирьох видів.

Команда має вигляд

$$\langle s_i, q_l \rangle \to \langle s_j, q_r, \pi \rangle,$$

де $\pi = \{\text{L,R,S,U,D}\}$, (*U* - уверх, *D* - униз).

Одностороння машина - це одноголовкова, однострічкова МТ, в якої стрічка є безмежною в один бік. Розглянемо випадок, коли стрічка безмежна вправо і має лівий кінець. Тоді команди з правим зсувом або без зсуву мають вигляд, як і в класичній МТ, та виконувані звичайним способом.

Вигляд команд з лівим зсувом і Характер їхнього виконання інший. Команда

$$\langle s_i, q_l \rangle \to \begin{cases} \langle s_{j_1} \text{L} q_{r_1} \rangle \\ \langle s_{j_2} \text{H} q_{r_2} \rangle \end{cases}$$

означає таке: якщо видима комірка не є крайньою лівою, то виконується те, що задане у верхньому рядку, в протилежному випадку - у нижньому. Отже, машина здатна відчувати крайню комірку і робити відповідний вибір (тобто команда є умовною).

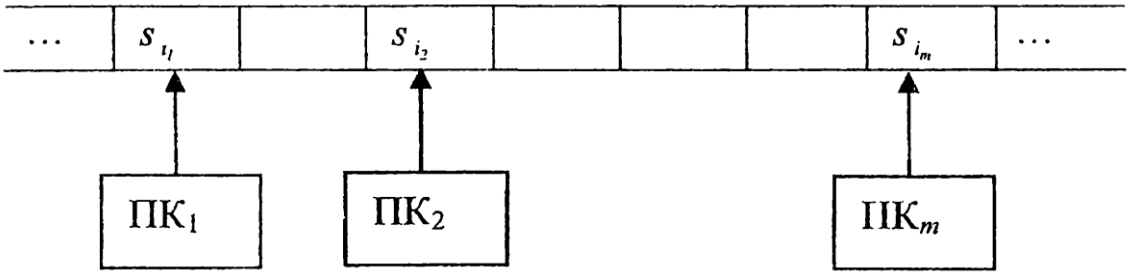

Рис. 11.6.

Машина з декількома функціональними механізмами - це однострічкова, багатоголовкова МТ, у якій головки пристроїв керування переміщаються вздовж стрічки та виконують свої команди незалежно одна від

одної (рис. 11.6). Оскільки може трапитись, що кілька головок спостерігають одну й ту ж комірку, і відповідні пристрої керування передбачають різні записи в цю комірку, то вважають, що пристрої керування пронумеровані, а в комірці зберігається запис, змінений пристроєм з найменшим номером.

Є також інші варіанти МТ, які поєднують у собі ті чи інші компоненти перерахованих вище машин. Проте всі такі машини не виводять з класу функцій, обчислюваних за Тьюрінгом.

Операції композиції, виконувані над алгоритмами, дають змогу утворювати нові, складніші алгоритми з відомих простих.

Над усіма МТ можна виконувати такі операції композиції: *суперпозиція, об'єднання, розгалуження, ітерація*. Розглянемо деякі з них.

Нехай задано дві машини Тьюрінга T_1 та T_2, які мають спільний зовнішній $S=\{s_1,s_2,\ldots,s_m\}$ і внутрішні $Q_1=\{q_0,q_1,\ldots,q_n\}$ та $Q_2=\{q_0',q_1',\ldots,q_t'\}$ алфавіти.

Суперпозицією, або ***добутком***, машин Тьюрінга T_1, та T_2 називають машину T з тим же зовнішнім алфавітом S та внутрішнім $Q=\{q_0,q_1,\ldots,q_n,q_{n+1},\ldots,q_{n+t}\}$ і програмою, яку отримують так. У всіх командах T_1 які містять заключний стан q_n, його заміняють на початковий стан машини T_2: $q_{n+1}=q_0'$. Всі інші символи в командах T_1 залишають без зміни. Кожен зі станів $q_j'\ (j=1,2,\ldots,t)$ заміняють станом q_{n+j}.

Сукупність усіх команд T_1 та T_2, змінена таким способом, і утворює програму суперпозиції машин T_1 та T_2, яку позначають T_1*T_2. Таблиця програми T складається з двох частин: верхня описує T_1, а нижня – T_2.

Якщо одна з машин має більше одного заключного стану, то обов'язково повинно бути визначено, який із заключних станів попередньої машини ототожнюється із початковим станом наступної. Наприклад,

$$T=T_1*\begin{cases}(1)T_2\\(2)\end{cases}\text{, або } T=T_1*\begin{cases}(1)\\(2)T_2\end{cases}.$$

Машина T у цьому випадку має також два заключні стани: один заключний стан T_1, інший - заключний стан T_2.

Операцію ітерації застосовують до однієї машини. Суть операції полягає в такому. Нехай машина T_1, має декілька заключних станів. Виберемо деякий *r*-й заключний стан і ототожнимо його в основній таблиці машини T_1 з її початковим станом. Отриману машину позначимо

$$T = \dot{T}_1 \begin{cases} (1) \\ \dots \\ (r) \\ \dots \\ (s) \end{cases}.$$

Вона є результатом ітерації машини $\dot{T}_1$. Тут символ '.' над T_1 означає ототожнення заключного стану з початковим станом машини T_1. Якщо T_1 має лише один заключний стан, то після ітерації отримують машину, яка не має заключного стану взагалі.

Проблема розпізнавання самозастосованості алгоритмів

Алгоритмічну систему Тьюрінга використовують для доведення нерозв'язності задач. Розглянемо одну з алгоритмічно нерозв'язних проблем.

Нехай система правил P алгоритму $A = \langle \varphi, P \rangle$ закодована певним способом у вхідному алфавіті алгоритму A. Позначимо цей а код системи P через P^{cod}.

Сформулюємо ***проблему розпізнавання самозастосовності*** алгоритмів так: *знайти алгоритм, здатний за кодом P^{cod} довільного алгоритму $A = \langle \varphi, P \rangle$ визначити, чи є А самозастосовним.*

Теорема. *Проблема розпізнавання самозастосовності алгоритму є алгоритмічно нерозв'язною.*

Доведення. Цю теорему доводять на підставі алгоритмічної системи Тьюрінга.

Нехай на стрічці деякої МТ T_k, зображено її власний код T_k^{cod} (тобто код програми і початкової конфігурації), записаний в алфавіті машини T_k. Якщо T_k

застосовна до T_k^{cod}, то будемо називати її самозастосовною, в іншому випадку - несамозастосовною. Нагадаємо, що застосовність МТ до вхідного слова означає зупинку цієї машини через скінченну кількість кроків.

Припустимо, що існує МТ M, яка розпізнає самозастосовність довільної машини T_k. Тоді в M кожен самозастосовний код T_k^{cod}, перетворюється в деякий символ v (який позначає позитивну відповідь на поставлене питання про самозастосовність), а кожен несамозастосовний код - у символ τ (який позначатиме негативну відповідь).

Звідси випливає, що можна побудувати і таку машину M_1, яка перероблятиме несамозастосовні коди в τ, тоді як до самозастосовних кодів машина M_1, вже буде незастосовною. Цього можна досягти шляхом такої зміни програми M, щоб після появи символу v машина, замість зупинки, нескінченно довго повторювала б цей символ.

Отже, M_1 буде застосовною до кожного несамозастосовного коду (у цьому разі видає символ τ) і незастосовна до самозастосовних кодів T_k^{cod} (нескінченно видає символ v). Однак це приводить до суперечності. Справді:

1. нехай машина M_1, самозастосовна, тоді вона застосовна до свого коду M_1^{cod} і переробляє його в символ τ. Однак поява цього символу саме й повинна означати, що M_1, несамозастосовна;
2. нехай M_1, несамозастосовна, тоді вона застосовна до M_1^{cod}, а це означає, що M_1 є самозастосовною.

Отримана суперечність доводить теорему, Оскільки припущення про існування машини M, яка розпізнає самозастосовність, не правильне.

Нерозв'язність цієї проблеми часто використовують для доведення нерозв'язності й інших масових проблем.