МАШИНА ТЮРІНГА.

6.1. Означення та конструкція машини Тьюрінга.

Машина Тьюрінга складається з:

1. *Управляючого пристрою,* який може перебувати в одному зі станів, які утворюють скінченну множину $Q = \{q_0, q_1, q_2, \ldots, q_m\}$. Серед станів управляючого пристрою виділені початковий стан q_1 і заключний стан q_0. У початковому стані машина Тьюрінга перебуває перед початком роботи, а, потрапивши в заключний стан, машина зупиняється.

2. Нескінченної *інформаційної стрічки*, яка розбита на секції (комірки), у кожній з яких може бути записаний один із символів абстрактного алфавіту $A = \{s_0, s_1, s_2, \ldots, s_n\}$, де s_0 відповідає порожньої секції.

3. *Головки стирання та запису,* яка у кожний момент часу тримає в поле зору деяку секцію стрічки. Залежно від символу, який оглядається, і стану управляючого пристрою головка може записати в секцію символ (він може співпадати з колишнім символом або бути порожнім), здвигатися уздовж стрічки на секцію вліво або вправо, залишатися на місці. При цьому управляючий пристрій може перейти в новий стан або залишитися в старому.

Скінченна сукупність символів алфавіту, з якими працює машина, називається *зовнішнім алфавітом*. Скінченна сукупність станів пристрою управління – *внутрішнім алфавітом*.

Таким чином, машина Тьюрінга складається із *внутрішньої пам'яті*, яка визначається внутрішнім алфавітом, і стрічки, яку називають *зовнішньою пам'яттю*. Хоча вважається, що стрічка машини нескінченна в обидва боки, однак у початковий момент часу тільки скінченне число її секцій заповнено непустими символами зовнішнього алфавіту, інші секції порожні. У будь-який наступний момент часу, також лише скінченний відрізок стрічки буде заповнений символами. Тому тут, як і для машини Поста, нам важлива не фактична нескінченність стрічки, а її необмеженість, яка дозволяє писати на ній будь-яку кількість як завгодно довгих, але скінченних слів.

Робота машини Тьюрінга складається з елементарних кроків: зчитування або запис символів, переміщення головки на секцію вліво або вправо, перехід управляючого пристрою в наступний стан. Умовимося, що ми будемо розглядати тільки *стандартні* машини, які при переміщенні головки можуть попередньо змінювати стан секції, яка оглядається.

Сукупність, яка утворена послідовністю станів всіх секцій стрічки та станом пристрою управління, називається *конфігурацією* машини Тьюрінга. Конфігурація задається у вигляді слова, яке описує конкретний стан машини в даний момент часу. Нехай у деякий момент часу машина Тьюрінга перебуває в стані, який наведений на рисунку 6.1.

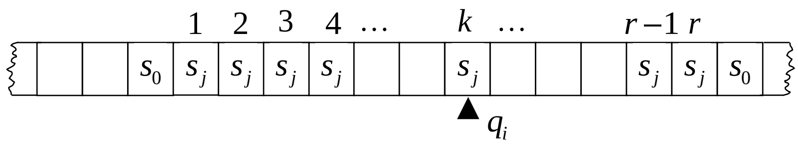


Рисунок 6.1 – Стан машини Тьюрінга

Конфігурація машини в цьому випадку буде зображуватися у вигляді слова: $\ldots\, s_0\, s_{j1}\, s_{j2} \ldots q_i\, s_{jk} \ldots s_{jr}\, s_0 \ldots$, де s_0 – символ, який позначає порожню секцію; r – загальне число заповнених секцій на стрічці; s_{j1} – стан першої лівої непустої секції; s_{jk} – стан секції, яка оглядається в даний момент часу; q_i – стан пристрою управління.

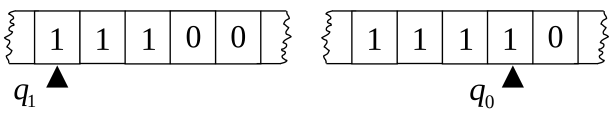
Кожна конфігурація містить лише одне входження символу q_i із внутрішнього алфавіту. Цей символ може бути самим лівим, але не може бути самим правим, тому що праворуч від нього повинен міститися символ секції, яка оглядається.

Якщо стандартна машина Тьюрінга, перебуваючи в стані q_i і оглядаючи символ s_k, записаний на стрічці, переходить у новий стан q_j, здійснюючи при цьому заміну символу, який оглядається, на символ s_m і переміщення головки вліво на одну секцію, то говорять, що машина виконала команду $q_i\ s_k \to q_j\ s_m$ Л. Якщо при переході зі стану q_i в стан q_j заміни символу не відбувається, то s_m в команді може бути відсутнім.

При маніпуляціях з головкою приймаються наступні позначення: Л – рух вліво; П – рух вправо; С – немає руху (стоп).

Часто, при записі команд стандартної машини Тьюрінга стрілка $\to$ опускається, тоді команда зображує собою слово, яке складається з п'яти символів $q_i\ s_k\ q_j\ s_m \begin{bmatrix} \text{Л} \\ \text{П} \\ \text{С} \end{bmatrix}$.

Розглянемо приклад машини Тьюрінга із зовнішнім алфавітом $C = \{0, 1\}$, внутрішнім алфавітом $Q = \{q_1, q_0\}$ і командами $q_1 1 q_1 1$П, $q_1 0 q_0 1$С. Нехай на стрічці записане слово $p = 11100$, а конфігурація машини зображена на рисунку 6.2а).



а) б)

Рисунок 6.2.
а) початковий стан; б) заключний стан

У результаті роботи машини Тьюрінга слово p перетворюється в слово $p' = 11110$, при цьому конфігурація машини стане такою, як показано на рисунку 6.2б).

Стандартна машина Тьюрінга із зовнішнім стандартним алфавітом $C = \{0, 1\}$ називається такою, що *не може стирати,* якщо вона здатна виконувати лише команди вигляду:

$$\left.\begin{matrix} q_\alpha 0 \to q_\beta c_i \begin{bmatrix} \text{Л} \\ \text{П} \\ \text{С} \end{bmatrix}; \ q_\alpha 1 \to q_\beta 1 \begin{bmatrix} \text{Л} \\ \text{П} \\ \text{С} \end{bmatrix}. \end{matrix}\right\}$$

У цьому випадку машина може вписати одиницю в порожню секцію, але не може стерти символ 1, якщо він уже вписаний в секцію.

Сукупність всіх команд, які може виконувати машина Тьюрінга, називається програмою.

Будемо вважати машину Тьюрінга *заданою*, якщо задані її зовнішній і внутрішній алфавіти, програма, початкова конфігурація та зазначено, які із символів позначають порожню секцію і заключний стан.

Нехай машина Тьюрінга задана зовнішнім алфавітом $A = \{a, b, c, d\}$, внутрішнім алфавітом $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$ і сукупністю команд: $q_0\ a\ q_1\ a$П; $q_0\ b\ q_0\ b$П; $q_2\ a\ q_5\ d$Л; $q_0\ c\ q_0\ c$П; $q_1\ d\ q_2\ c$Л; $q_3\ a\ q_4\ d$Л; $q_4\ b\ q_2\ c$Л. Порожню секцію позначає символ s_0, а заключний стан – символ q_5.

Розглянемо тепер роботу даної машини для переробки вхідного слова $p = bcadc$ при заданій початковій конфігурації $q_0\ bcadc$. При запуску машини в хід буде породжена наступна послідовність конфігурацій:

1 крок: $b\,q_0\;cadc\;(q_0\,b \to q_0\,b\,\Pi)$;

2 крок: $bc\,q_0\;adc\;(q_0\,c \to q_0\,c\,\Pi)$;

3 крок: $bca\,q_1\;dc\;(q_0\,a \to q_1\,a\,\Pi)$;

4 крок: $bc\,q_2\;acc\;(q_1\,d \to q_2\,c\,Л)$;

5 крок: $b\,q_5\;cdcc\;(q_2\,a \to q_5\,d\,Л)$.

Таким чином, вхідне слово $p = bcadc$ перероблене машиною у вихідне слово $p' = bcdcc$ і у заключному стані каретка оглядає перший символ c вихідного слова.

Якщо сукупність команд програми машини Тьюрінга відносно велика, то таку програму доцільно записувати у вигляді *таблиці відповідності*. Таблиця відповідності для попередньої задачі має вигляд:

$Q \setminus A$	s_0	a	b	c	d
q_0	—	$q_1\,a\,\Pi$	$q_0\,b\,\Pi$	$q_0\,c\,\Pi$	—
q_1	—	—	—	—	$q_2\,c\,Л$
q_2	—	$q_5\,d\,Л$	—	—	—
q_3	—	$q_4\,d\,Л$	—	—	—
q_4	—	—	$q_2\,c\,Л$	—	—
q_5	С Т О П				

Розглянемо ще кілька прикладів машин Тьюрінга.

6.2 Приклади машин Тьюрінга.

1. Нехай машина Тьюрінга має зовнішній стандартний алфавіт $C = \{0, 1\}$ і внутрішній алфавіт $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$. Для вхідного слова $p = 0010$ визначити порядок виконання команд і породжувану конфігурацію машини, якщо початкова конфігурація має вигляд $q_1 0010$, заключний стан позначає символ q_0, а програма задана таблицею відповідності 6.1.

Таблиця 6.1

$Q \setminus C$	0	1
q_0	$q_0\,0С$	$q_0\,1С$
q_1	$q_2\,\Pi$	—
q_2	$q_3\,1С$	—
q_3	$q_4\,Л$	$q_3\,\Pi$
q_4	—	$q_5\,0С$
q_5	$q_6\,Л$	—
q_6	$q_0\,0С$	$q_6\,Л$

Розв'язання даної задачі зручно зобразити у вигляді наступної таблиці:

№	Команда	Рисунок	Конфігурація
0	Початковий стан	0 0 1 0, q_1▲	$q_1 0010$
1	$q_1 0 \to q_2\,\Pi$	0 0 1 0, q_2▲	$0\,q_2\,010$
2	$q_2 0 \to q_3\,1С$	0 1 1 0, q_3▲	$0\,q_3\,110$
3	$q_3 1 \to q_3\,\Pi$	0 1 0 ⇒, q_3▲ 0 1 1 0, q_3▲	$01\,q_3\,10 \Rightarrow 011\,q_3\,0$
4	$q_3 0 \to q_4\,Л$	0 1 1 0, q_4▲	$01\,q_4\,10$
5	$q_4 1 \to q_5\,0С$	0 1 0 0, q_5▲	$01\,q_5\,00$

6	$q_5 0 \to q_6$ Л	[0][1][0][0], q_6▲ під 1	$0\,q_6\,100$
7	$q_6 1 \to q_6$ Л	[0][1][0][0], q_6▲ під 0	$q_6\,0100$
8	$q_6 0 \to q_0$ 0С	[0][1][0][0], q_0▲ під 0	$q_0\,0100$

2. Скласти програму машини Тьюрінга, яка додає одиницю до довільного числа.

Вхідне слово p складається із цифр цього числа, записаних послідовно в секціях стрічки машини. Припустимо, що в початковий момент часу головка оглядає саму праву секцію запису. Машина повинна додати одиницю до останньої цифри, а якщо це була цифра 9, то замінити її на 0 і поступити аналогічно з попередньою цифрою.

Для здійснення зазначених замін доцільно ввести два стани: q_1 – стан зміни цифр і q_2 – стан зупинки. Якщо в стані q_1 головка бачить одну із цифр діапазону $0 \div 8$, то вона повинна збільшити її на одиницю і перейти в стан q_2. Якщо в стані q_1 головка оглядає секцію із цифрою 9, то 9 повинна замініться на 0, а головка переміститься вліво до попередньої цифри, залишаючись у тім же стані q_1. Так повинне тривати доти, поки не зустрінеться цифра, яка менше, ніж 9. Якщо ж всі цифри виявляться дев'ятками, то головка послідовно замінить їх нулями, а, записавши 0 на місце самої першої цифри запису, необхідно буде здвинутися вліво, побачити там порожню секцію s_0, записати на її місце 1 і, нарешті, перейти в стан q_2.

Таким чином, можна задати машину Тьюрінга, яка визначається: зовнішнім алфавітом $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, s_0\}$; внутрішнім алфавітом $Q = \{q_1, q_2\}$; початковим станом $q_1 a_1$ або $a_1 \ldots q_1 a_n$, де $n > 1$ – довжина вхідного слова, яке складається більш ніж з одного символу; програмою, зображеною у вигляді таблиці відповідності 6.2.

Таблиця 6.2

Q \ A	0	1	2	…	9	s_0
q_1	q_2 1С	q_2 2С	q_2 3С		q_1 0Л	q_2 1С
q_2	С Т О П					

3. Скласти програму машини Тьюрінга, яка призначена для підрахунку кількості розташованих підряд однакових символів. Припустимо, що на стрічці записане слово p довільної довжини, яке складає із символів «↟». Потрібно стерти всі символи і записати на стрічку їхнє число, зображене в десятковій системі.

Будемо формувати шукане число на стрічці ліворуч від заданого слова. У початковий момент машина Тьюрінга може оглядати будь-якій із символів і перебуває в стані q_1.

План алгоритму розв’язання задачі в рамках даної алгоритмічної системи зручно зобразити у вигляді наступних пунктів словесного запису:

1. Знайти правий кінець слова на стрічці й перейти до вказівки 2.
2. Якщо слово закінчується символом «↟», то стерти цей символ і перейти до вказівки 3, інакше – зупинити машину.
3. Додати до числа одиницю й перейти до вказівки 1.

Кожний із цих пунктів можливого плану розв’язання задачі може бути реалізований відповідним станом машини Тьюрінга. У стані q_1 головка буде шукати правий кінець слова p, q_2 буде станом стирання символів, а q_3 буде станом додавання до числа одиниці.

Машина бачить на стрічці цифри, які вона писала сама, і символи «↟», які знаходяться там із самого початку. У стані q_1 ознакою досягнення правого кінця слова служить порожня секція s_0. При цьому головка здвигається по стрічці на крок вліво і переходить у стан q_2. У цьому стані можливі два варіанти: або головка бачить символ (тоді вона стирає його і переходить у стан q_3 зі здвигом вліво); або головка бачить цифру (тоді машина зупиняється, тому що це означає, що всі символи вже стерті). У стані q_3 головка рухається по стрічці вліво, минаючи символи, які залишилися, поки не дійде до числа, і додає до числа одиницю, аналогічно тому, як це робилося в прикладі 2.

Таким чином, машина Тьюрінга визначається: зовнішнім алфавітом $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \dagger, s_0\}$; внутрішнім алфавітом $Q = \{q_1, q_2, q_3\}$, де q_2 – заключний стан; початковим станом q_1, у якому головка може оглядати будь-який із символів вхідного слова; програмою, зображеної у вигляді таблиці відповідності 6.3.

Таблиця 6.3

$Q \backslash A$	s_0	0	1	2	…	9	$\dagger$
q_1	q_2 Л	q_1 П	q_1 П	q_1 П		q_1 П	q_1 П
q_2	С Т О П						q_3 s_0 Л
q_3	q_1 1 П	q_1 1 П	q_1 2 П	q_1 3 П		q_3 0 Л	q_3 Л

Читачу рекомендується самостійно перевірити адекватність роботи заданої машини Тьюрінга для простої початкової конфігурації, наприклад $\dagger\, q_1 \,\dagger\, \dagger\, \dagger$.

6.3 Деякі різновиди машин Тьюрінга.

Дотепер, ми розглядали машини Тьюрінга, які відрізнялися друг від друга тільки числом станів пристрою управління. При цьому існують модифікації машин Тьюрінга, які визначаються ще й кількістю використовуваних стрічок.

Розглянемо б*тострічкову машину Тьюрінга*, схема якої зображена на ри[illegible]дається зі скінченного числа стрічок, розбитих на секції. У кожній секції може міститься один із символів зовнішнього алфавіту $A = \{s_0, s_1, s_2, \ldots, s_n\}$, де s_0 – символ порожньої секції. Пристрій управління машини в кожний момент часу може перебувати в одному зі скінченної множини станів $Q = \{q_0, q_1, q_2, \ldots, q_n\}$, де q_0 – заключний стан машини. У кожний момент часу машина оглядає одну секцію кожної стрічки.

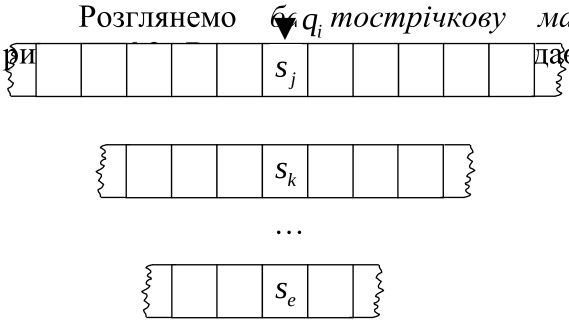


Рисунок 6.3 – Схема багатострічкової машини Тьюрінга

Конфігурація машини вважається заданою, якщо відомі стан пристрою управління, стан всіх секцій стрічок і указані секції, які оглядаються машиною. Для k-стрічкової машини конфігурація її в i-й момент часу описується системою k слів вигляду:

$$\left.\begin{array}{l} s_{i11}s_{i12}\ldots s_{i1z}q_i s_{i1z+1}\ldots s_{i1t}; \\ s_{i21}s_{i22}\ldots s_{i2z}q_i s_{i2z+1}\ldots s_{i2t}; \\ \ldots; \\ s_{ik1}s_{ik2}\ldots s_{ikz}q_i s_{ikz+1}\ldots s_{ikv}. \end{array}\right\} \qquad (1)$$

Кожне слово системи (1) описує конфігурацію кожної стрічки багатострічкової машини. Перший індекс будь-якого символу в слові відповідає моменту часу, другий – номеру стрічки, а третій – номеру секції на стрічці, якщо рахувати зливо на право.

Залежно від внутрішнього стану пристрою управління q_i і вмісту секцій всіх k стрічок, які оглядаються в даний момент $(s_{i1z+1},\ s_{i2z+1},\ \ldots,\ s_{ikz+1})$, машина переходить у наступний стан за допомогою таких дій:

1. Стан кожної секції, яка оглядається головкою, замінюється новим станом (який може співпадати зі старим).
2. Кожна стрічка переміщується на одну секцію вправо, вліво або залишається нерухомою.
3. Стан пристрою управління q_i переводиться в новий стан q_j (який може співпадати зі старим).

Вважається, що якщо пристрій управління в деякий момент часу переходить у заключний стан q_0, то машина зупиняється, і в наступні моменти часу конфігурація машини не змінюється.

Загальний вигляд команди багатострічкової машини Тьюрінга прийнято записувати так:

$$q_i s_{\alpha 1} s_{\alpha 2} \ldots s_{\alpha k} \rightarrow q_j s_{\beta 1} \begin{bmatrix} Л \\ П \\ С \end{bmatrix} s_{\beta 2} \begin{bmatrix} Л \\ П \\ С \end{bmatrix} \ldots s_{\beta k} \begin{bmatrix} Л \\ П \\ С \end{bmatrix}. \tag{2}$$

Таким чином, якщо машина виконала команду вигляду (2), то вона, перебуваючи в стані q_i і оглядаючи у цей момент часу секції стрічок із символами $s_{\alpha 1},\ s_{\alpha 2},\ \ldots,\ s_{\alpha k}$, переходить у стан q_j і заміняє вміст секцій символами $s_{\beta 1},\ s_{\beta 2},\ \ldots,\ s_{\beta k}$ відповідно. При цьому кожна зі стрічок може переміщатися або залишатися нерухомою.

Зрозуміло, що k-стрічкова машина Тьюрінга визначає алгоритм переробки вхідних слів вигляду (1) за допомогою алфавітних операторів (команд) вигляду (2).

Ми вже відзначали, що для будь-якої алгоритмічної системи, велике значення має можливість побудови універсального алгоритму. Незважаючи на те, що різні алгоритми, які реалізовуються в рамках даної алгоритмічної системи, відрізняються набором команд, внутрішнім і зовнішнім алфавітами, можна побудувати *універсальну машину Тьюрінга,* здатну виконувати роботу будь-якої наперед заданої машини Тьюрінга.

Універсальна машина має у своєму розпорядженні фіксований скінченний зовнішній алфавіт та повинна бути пристосована приймати на вхід різні стани пристрою управління і конфігурації із символів абстрактних алфавітів будь-якої довжини. Це досягається шляхом кодування конфігурації і програми будь-якої даної машини Тьюрінга в символах зовнішнього алфавіту універсальної машини. Саме кодування повинне виконуватися в такий спосіб:

1. Різні символи заміняються різними кодовими групами, при цьому один і той же символ заміняється всюди, де б він не зустрівся, однією і тією же кодовою групою.
2. Рядки кодових записів повинні однозначним способом розбиватися на окремі кодові групи.
3. Повинна бути можливість розпізнати, які кодові групи відповідають переміщенням $\{Л, П, С\}$, які символам зовнішнього алфавіту, а які символам внутрішнього алфавіту.

Розглянемо приклад такого кодування для машини Тьюрінга (T), яка має зовнішній алфавіт $A = \{ s_1,\ s_2,\ s_3,\ \ldots,\ s_k \}$ і внутрішній алфавіт $Q = \{ q_1,\ q_2,\ q_3,\ \ldots,\ q_m \}$.

Якщо зовнішнім алфавітом універсальної машини Тьюрінга (T_U) вибрати стандартний двійковий алфавіт $C = \{0, 1\}$, то наведені умови кодування будуть дійсно дотримані при наступному способі кодування:

1. Як кодові групи беруться $3 + k + m$ різних слів вигляду 100…001, де k, m – число символів відповідно зовнішнього та внутрішнього алфавітів машини T, а число 3 визначає кількість здвигів {Л, П, С}. Таким чином, розбивка рядків на кодові групи однозначно виконується шляхом виділення послідовності нулів, які містяться між двома одиницями.

2. Зіставлення кодових груп вхідним символам зовнішнього та внутрішнього алфавітів здійснюється відповідно до наступної таблиці кодування.

	Символ	Кодова група
Здвиги	Л	101
	С	1001
	П	10001
Зовнішній алфавіт	s_1	100001
	s_2	10000001
	s_3	1000000001
	…	…
	s_k	100……001
Внутрішній алфавіт	q_1	1000001
	q_2	100000001
	q_3	10000000001
	…	…
	q_m	1000…001

$\left\{ \begin{array}{l} \text{Парне число } 2(\text{i}+1) \\ \text{нулів, більше, ніж } 2. \end{array} \right.$

$\left\{ \begin{array}{l} \text{Непарне число } 2(\text{i}+1)+1 \\ \text{нулів, більше, ніж } 3\,. \end{array} \right.$

Розглянемо попередній приклад машини Тьюрінга, яка переробляла вхідне слово $p = bcadc$ у вихідне слово $p' = bcdcc$.

Вхідне слово в універсальній машині T_U буде зображене наступним рядком:

$$\overbrace{10000001}^{6\ \textit{нулей}}\overbrace{1000000001}^{8}\overbrace{100001}^{4}\overbrace{100000000001}^{10}\overbrace{1000000001}^{8}$$

Програма має вигляд:

$$\overbrace{1000001}^{5}\overbrace{10000001}^{6}\overbrace{1000001}^{5}\overbrace{10000001}^{6}\overbrace{10001}^{3} \qquad (q_0\, b \to q_0\, b\,\text{П})$$

$$\overbrace{1000001}^{5}\overbrace{1000000001}^{8}\overbrace{1000001}^{5}\overbrace{1000000001}^{8}\overbrace{10001}^{3} \qquad (q_0\, c \to q_0\, c\,\text{П})$$

$$\overbrace{1000001}^{5}\overbrace{100001}^{4}\overbrace{100000001}^{7}\overbrace{100001}^{4}\overbrace{10001}^{3} \qquad (q_0\, a \to q_1\, a\,\text{П})$$

$$\overbrace{100000001}^{7}\overbrace{100000000001}^{10}\overbrace{10000000001}^{9}\overbrace{1000000001}^{8}\overbrace{101}^{1} \qquad (q_1\, d \to q_2\, c\,\text{Л})$$

$$\overbrace{10000000001}^{9}\overbrace{100001}^{4}\overbrace{10000000000000001}^{15}\overbrace{100000000001}^{10}\overbrace{101}^{1} \qquad (q_2\, a \to q_5\, d\,\text{Л})$$

Таким чином, якщо яка-небудь машина T розв’язує деяку задачу, то машина T_U здатна розв’язати цю задачу за умови, що крім кодів вхідних даних цієї задачі на її стрічку буде поданий код програми машини T. Результатом роботи T_U буде код вихідного слова, у яке машина T переробляє вхідне слово.

Відзначимо, що при побудові машини T_U ми обійшлися всього двома символами на стрічці, при цьому не переслідували цілей оптимізації та не жалкували ні символів стрічки, ні станів, а лише прагнули до наочності такої побудови. Те, що машину T_U можна побудувати всього із двома символами на стрічці – факт досить очевидний. Шеннон установив менш очевидний факт – він побудував машину T_U із двома станами. У той же час Мінським і Боброу було показано, що універсальна машина із двома символами та двома станами неможлива. Взагалі, в певних рамках зменшення числа символів зовнішнього алфавіту спричиняє збільшення числа символів внутрішнього алфавіту й навпаки. Оскільки величини k та m зв'язані між собою, Шеннон запропонував оцінювати складність T_U як добуток числа символів зовнішнього алфавіту k і числа відмінних від заключного станів пристрою управління m'.

ЕОМ будуються як універсальні машини T_U, оскільки в запам'ятовувальний пристрій поряд з вхідними даними поставленої задачі вводиться також і програма її розв'язання. При цьому необхідно замітити, що основні роботи з теорії алгоритмів, у тому числі й роботи А. Тьюрінга з'явилися раніше, ніж сучасні електронні обчислювальні машини.

6.4. Композиції машин Тьюрінга.

За програмою заданої машини Тьюрінга іноді буває важко зрозуміти, як конкретна машина вирішує дану задачу. Як правило, виникає необхідність у поясненнях, типа таких, які наводилися в п. 6.2. Крім того, для більш складних задач виявляється достатньо складно будувати таблиці відповідності.

Операції композиції, які можна виконувати над алгоритмами, застосовні й до машин Тьюрінга, оскільки ці машини зображують собою алгоритми для переробки слів.

Таким чином, доцільно сформувати набір базових машин Тьюрінга, які призначені для розв'язання простих задач і, за допомогою операцій композиції, утворювати нові, більш складні машини. Розглянемо деякі композиції машин Тьюрінга.

Нехай задані дві машини Тьюрінга T_1 і T_2, які мають загальний зовнішній алфавіт $A = \{ s_0, s_1, s_2, \dots, s_k \}$ і внутрішні алфавіти, відповідно $Q_1 = \{ q_0, q_1, q_2, \dots, q_n \}$, $Q_2 = \{ q'_0, q'_1, q'_2, \dots, q'_t \}$. *Композитом* або *добутком* машин T_1 і T_2 називають машину T з тим же зовнішнім алфавітом A, внутрішнім алфавітом $Q = \{ q_0, q_1, q_2, \dots, q_n, q_{n+1}, q_{n+2}, \dots, q_{n+t} \}$ і програмою, яка дістається у такий спосіб. У всіх командах T_1 символ заключного стану q_0 заміняють символом q_{n+1}. Всі інші символи в командах T_1 залишаються незмінними. У командах T_2, навпаки, символ заключного стану залишається незмінним, але інші символи q'_i заміняють символами q'_{i+n}. Добуток машин T_1 і T_2 позначається так $T = T_1 \cdot T_2$, а програма машини T буде зображувати сукупність всіх команд, змінених описаним способом.

Машина T_1

$Q \backslash C$	0	1
q_1	q_2 0П	q_1 1П
q_2	q_2 0П	q_0 1С

Машина T_2

$Q \backslash C$	0	1
q_1	q_0 0С	q_1 0П

Машина $T = T_1 \cdot T_2$

$Q \backslash C$	0	1
q_1	q_2 0П	q_1 1П
q_2	q_2 0П	q_3 1С
q_3	q_0 0С	q_3 0П

Як приклад, розглянемо машини T_1 і T_2, які задані простими таблицями відповідності. Видно, що ці машини застосовні до вхідних слів, які задані в стандартному алфавіті C, а в початковому стані q_1 головка повинна оглядати саму ліву з позначених на стрічці секцій. Також, легко зрозуміти, що машина T_1 для будь-якого можливого початкового стану відшукує найближчу праворуч групу одиниць після групи нулів, а T_2 – стирає всі одиниці, якщо вони є, до найближчого праворуч нуля.

Таким чином, машина T, таблиця відповідності для якої складена відповідно до описаних вище правил, спочатку відшукує найближчу групу одиниць, а потім стирає їх. Отже, машина T є добутком машин T_1 і T_2, якщо їхня послідовна робота еквівалентна роботі однієї машини T.

У такий же спосіб визначається *композиція піднесення до степеня*: n-им степенем машини T називається добуток $T \cdot T \cdot T \cdot \ldots \cdot T$ з n співмножниками.

У тому випадку, коли одна з машин, що перемножуються, має кілька заключних станів, множення визначається зовсім аналогічно, але при цьому, обов'язково повинна бути присутня вказівка на те, який заключний стан попереднього співмножника необхідно сполучити з початковим станом наступного. Так, наприклад, якщо машина T_1 має два заключних стани, то добуток T_1 і T_2 визначається залежно від того, чи ототожнюється початковий стан машини T_2 з першим або з другим заключним станом машини T_1:

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

Ще однією операцією композиції, яку ми розглянемо, є *операція ітерації*. Операція ітерації застосовується до однієї машини, а суть її полягає в наступному. Нехай машина T_1 має кілька заключних станів. Виберемо яке-небудь r-ий заключний стан і ототожнимо його в таблиці відповідності машини T_1 з її заключним станом. Отримана машина позначається через

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

і є результатом ітерації машини T_1. Тут крапка вказує на ототожнення заключного стану з початковим станом машини T_1. Зрозуміло, що якщо T_1 має всього один заключний стан, то застосування операції ітерації призведе до побудови машини T, яка не має заключного стану зовсім, і внаслідок цього буде працювати нескінченно.

Описуючи різні алгоритми для своїх машин і доводячи результативність різноманітних композицій алгоритмів, Тьюрінг переконливо продемонстрував багатство можливостей запропонованої їм конструкції. Це дозволило йому виступити із твердженням, яке називається *теза Тьюрінга* і формулюється так: «Будь-який алгоритм може бути реалізований відповідною машиною Тьюрінга».

Ця основна гіпотеза теорії алгоритмів у формі Тьюрінга є одночасно формальним визначенням алгоритму. Як і аналогічні визначення і формулювання, які ми неодноразово наводили в рамках інших алгоритмічних систем, тезу Тьюрінга довести не можна, оскільки саме поняття алгоритму (або ефективної процедури) є нестрогим. Теза Тьюрінга не є постулатом математичної теорії, вона представляє собою твердження, яке зв'язує теорію з тими об'єктами, для опису яких вона створена. За своїм характером теза Тьюрінга нагадує гіпотези фізики про адекватність математичних моделей фізичним явищам і процесам.

Підтвердженням тези Тьюрінга є дві обставини. По-перше, це математична практика, яка підтвердила можливість здійснювати або імітувати на машині Тьюрінга всі алгоритмічні процеси, які коли-небудь описувалися математиками. По-друге, та обставина, що формальний опис алгоритму в рамках інших алгоритмічних систем може бути зведений до його опису в рамках машини Тьюрінга. Таким чином, питання про можливість або неможливість розв'язувального алгоритму для задачі того або іншого типу слід розуміти як питання про існування або неіснування машини Тьюрінга, яка володіє необхідними властивостями.