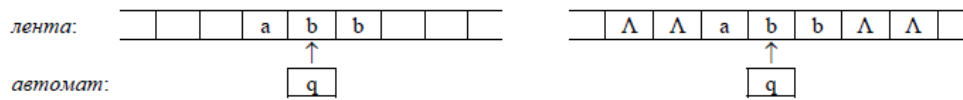


Машини Тюрінга: лекційний матеріал та завдання

Машина Тюрінга (МТ) складається з двох частин - стрічки і автомата:



Стрічка використовується для зберігання інформації. Вона нескінченна в обидва боки і розбита на клітинки, які ніяк не нумеруються і не іменуються. У кожній клітинці може бути записаний один символ або нічого не записано. Вміст клітинки може змінюватися - в неї можна записати інший символ або витерти символ, що знаходиться там.

Домовимося порожню клітинку позначати знаком Λ («лямбда»). Операцію видалення літери в деякій клітинці можна розглядати як запис в цю клітинку символу Λ .

Автомат - це активна частина МТ. У кожен момент він розміщується під однією із клітинок стрічки і бачить її вміст; це видима клітинка, а той символ, що знаходиться в ній - це видимий символ; вміст сусідніх та інших клітинок автомат не бачить. Крім того, в кожен момент автомат знаходиться в одному з станів, які будемо позначати буквою **q** з номерами: q_1 , q_2 і т.п. Перебуваючи в деякому стані, автомат виконує якусь певну операцію (наприклад, переміщується направо по стрічці, замінюючи всі символи *b* на *a*, перебуваючи в іншому стані - іншу операцію).

Пару видимого символу (*S*) і поточного стану автомата (*q*) будемо називати конфігурацією і позначати $\langle S, q \rangle$.

Автомат може виконувати три елементарних дії:

- 1) записати у видиму клітинку новий символ (міняти вміст інших клітинок автомат не може);
- 2) рухатись на одну клітинку вліво або вправо («перестрибувати» відразу через кілька клітинок автомат не може);
- 3) переходити в новий стан.

Нічого іншого робити автомат не вміє, тому всі більш складні операції так чи інакше повинні бути зведені до цих трьох елементарних дій.

Такт роботи машини Тюрінга

МТ працює тактами, які виконуються один за іншим. На кожному такті автомат МТ виконує три наступних дії, причому обов'язково в зазначеному порядку:

- 1) записує деякий символ S' у видиму клітинку (зокрема, може бути записаний той же символ, що і був у ній, тоді вміст цієї клітинки не змінюється);
 - 2) зсувається на одну клітинку вліво (позначення - L), або на одну клітинку вправо (позначення - R), або залишається нерухомим (позначення - N , в деяких джерелах використовується знак S - від слова «стоп»);
 - 3) переходить в деякий стан q' (зокрема, може залишитися в попередньому стані).
- Формально дії одного такту будемо записувати у вигляді трійки:

$S', [L, R, N], q'$

де конструкція з квадратними дужками означає можливість запису в цьому місці будь-який з букв L , R або N .

Програма для машини Тюрінга (МТ)

Програма записується у вигляді наступної таблиці:

	S_1	S_2	...	S_i	...	S_n	Λ
q_1							
...							
q_j				$S', [L, R, N], q'$			
...							
q_m							

Зліва перераховуються всі стани, в яких може перебувати автомат (q_i), зверху - всі символи (у тому числі і Λ), які автомат може бачити на стрічці. (Які саме символи та стани вказувати в таблиці – визначає автор програми.) На перетинах (в осередках таблиці) вказуються ті такти, які повинен виконати автомат, коли він знаходиться у відповідному стані і бачить на стрічці відповідний символ.

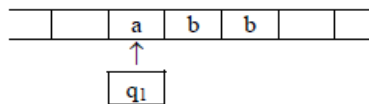
В цілому таблиця визначає дії МТ при всіх можливих конфігураціях і тим самим повністю задає поведінку МТ. Описати алгоритм у вигляді МТ - значить побудувати таку таблицю.

Правила виконання програми

До виконання програми потрібно виконати наступні попередні дії.

По-перше, треба записати на стрічку вхідне слово, до якого буде застосована програма. Вхідне слово - це скінченна послідовність символів, записаних в сусідніх клітинках стрічки; всередині вхідного слова порожніх клітин бути не повинно, а ліворуч і праворуч від нього повинні бути тільки порожні клітини. Порожнє вхідне слово означає, що всі клітини стрічки порожні.

По-друге, треба встановити автомат в стан q_1 (вказане в таблиці першим), і розмістити його під першим символом вхідного слова:



Якщо початкове слово порожнє, то автомат може розпочинати роботу з будь-якої клітинки, тому що всі вони порожні.

Після цих попередніх дій починається виконання програми. У таблиці відшукується клітинка на перетині першого рядка (тому що автомат знаходиться в стані q_1) і того стовпця, який відповідає першому символу вхідного слова (це необов'язково лівий стовпчик таблиці), і виконується такт, зазначений в даній клітинці. У результаті автомат опиниться в новій конфігурації. Тепер такі ж дії повторюються, але вже для нової конфігурації: в таблиці відшукується клітинка, що відповідає стану і символу цієї конфігурації, і виконується такт з цієї клітинки. І так далі.

Коли завершується виконання програми? Введемо поняття такту зупинки q_0 . Потрапивши на такт зупину, МТ, за визначенням, зупиняється, завершуючи свою роботу.

В цілому можливі два результати роботи МТ над вхідним початковим словом:

1) Перший результат - це коли в якийсь момент МТ зупиняється (потрапляє на такт зупинки). У такому випадку говорять, що МТ застосовна до заданого вхідного слова. А те слово, яке до цього моменту отримано на стрічці, вважається вихідним словом, тобто результатом роботи МТ, відповіддю.

У момент зупинки повинні бути виконані такі обов'язкові умови:

- Всередині вихідного слова не повинно бути порожніх клітин (відзначимо, що під час виконання програми всередині слова порожні клітинки можуть бути, але в кінці їх вже не повинно залишитися);

- Автомат зобов'язаний зупинитися під одним із символів вихідного слова (під яким саме - не грає ролі), а якщо слово пусте - під будь-якою клітинкою стрічки.

2) Другий можливий результат («поганий»): це коли МТ зациклюється, ніколи не потрапляючи на такт зупинки (наприклад, автомат на кожному кроці зсувається вправо і тому не може зупинитися, тому що стрічка нескінченна). У цьому випадку говорять, що МТ незастосовна до заданого вхідного слова. Ні про який результат в такому випадку не може йти й мови.

Відзначимо, що один і той же алгоритм (програма МТ) може бути застосовним до одних вхідних слів (тобто зупинятися) і незастосовним до інших (тобто зациклюватися). Таким чином, застосовність / незастосовність залежить не тільки від самого алгоритму, але і від вхідного слова.

Домовленості для скорочення запису

1) Якщо в такті не змінюється видимий символ, або не змінюється стан автомата, то у відповідній позиції такту ми не будемо нічого писати.

Наприклад, при конфігурації $\langle a, q1 \rangle$ такі записи тактів еквівалентні:

$$\begin{array}{lcl} a, R, q3 & \equiv & , R, q3 \\ b, N, q2 & \equiv & b, , q2 \\ a, L, q1 & \equiv & , L, \\ a, N, q1 & \equiv & , , \end{array}$$

Зауваження. Коми в тактах бажано не опускати, так як інакше можлива плутанина, якщо серед символів на стрічці можуть зустрітися букви L і R.

2) Якщо треба вказати, що після виконання деякого такту МТ повинна зупинитися, то в такій позиції цього такту будемо писати знак «!».

Наприклад, такт $\langle b, L, ! \rangle$ означає наступні дії: запис символу b в клітинку стрічки, рух вліво і зупинку.

3) Якщо заздалегідь відомо, що в процесі виконання програми не може з'явитися деяка конфігурація, тоді, щоб підкреслити це явно, будемо в відповідній комірці таблиці малювати хрестик. (Формально цей хрестик вважається тактом зупинки.)

ПРИКЛАДИ СКЛАДАННЯ ПРОГРАМ МТ

Розглянемо приклади на складання програм для МТ, щоб продемонструвати деякі типові прийоми програмування на МТ.

Для скорочення формулювання завдань введемо наступні дві умови:

- Буквою Р будемо позначати вхідне слово;
- Буквою А будемо позначати алфавіт вхідного слова, тобто набір тих символів, з яких і тільки яких може складатися Р (відзначимо, однак, що в проміжних і вихідному словах можуть з'являтися й інші символи).

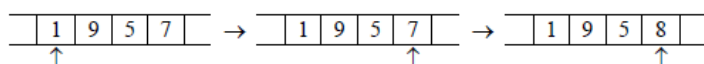
Приклад 1 (переміщення автомата, заміна символів)

$A = \{ 0,1,2,3,4,5,6,7,8,9 \}$ Нехай Р - непорожнє слово; значить, Р - це послідовність з десятичних цифр, тобто запис невід'ємного цілого числа в десятичній системі числення. Потрібно задати на стрічці запис числа, яке на 1 більше числа Р (тобто задати наступне число до вхідного)

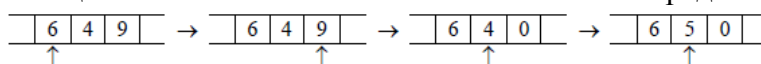
Розв'язок.

Для вирішення цього завдання пропонується виконати наступні дії:

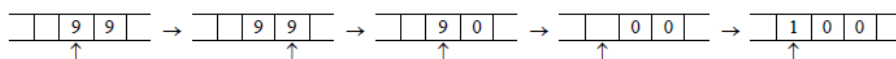
1. Перегнати автомат під останню цифру числа.
2. Якщо це цифра від 0 до 8, то замінити її цифрою на 1 більше і зупинитися; наприклад:



3. Якщо ж це цифра 9, тоді замінити її на 0 і повернути автомат до попередньої цифри, після цього таким же способом збільшити на 1 передостанню цифру; наприклад:



Особливий випадок: у Р тільки дев'ятки (наприклад, 99). Тоді автомат буде рухатися вліво, замінюючи дев'ятки на нулі, і врешті-решт опиниться під порожній клітинці. У цю порожню клітинку треба записати 1 і зупинитися (відповіддю буде 100):



У виді програми для МТ ці дії описуються наступним чином:

	0	1	2	3	4	5	6	7	8	9	Λ
q1	0,R,q1	1,R,q1	2,R,q1	3,R,q1	4,R,q1	5,R,q1	6,R,q1	7,R,q1	8,R,q1	9,R,q1	Λ,L,q2
q2	1,N,!	2,N,!	3,N,!	4,N,!	5,N,!	6,N,!	7,N,!	8,N,!	9,N,!	0,L,q2	1,N,!

Пояснення.

q1 - це стан, в якому автомат «біжить» під останню цифру числа. Для цього він весь час рухається вправо, не змінюючи видимі цифри і залишаючись в тому ж стані. Але тут є одна особливість: коли автомат перебуває під останньою цифрою, то він ще не знає про це (адже він не бачить, що записано в сусідніх клітинах) і визначить це лише тоді, коли потрапить на порожню клітинку. Тому, дійшовши до першої порожньої клітинки, автомат повертається назад під останню цифру і переходить в стан q2 (вправо рухатися вже не треба).

q2 - це стан, в якому автомат додає 1 до тієї цифри, яку бачить в даний момент. Спочатку це остання цифра числа; якщо вона - в діапазоні від 0 до 8, то автомат замінює її цифрою, яка на 1 більше, і зупиняється. Але якщо це цифра 9, то автомат замінює її на 0 і зсувається вліво, залишаючись в стані q2. Тим самим, він буде тепер додавати 1 до

попередньої цифри. Якщо і ця цифра дорівнює 9, то автомат замінює її на 0 і зсувається вліво, залишаючись в стані q2, тому повинен виконати ту саму дію – збільшити на 1 цифру. Якщо ж автомат зрушився вліво, а у клітинці немає цифри (тобто є «порожньо»), то він записує сюди 1 і зупиняється.

Відзначимо, що для порожнього вхідного слова наше завдання не визначене, тому на цьому слові МТ може вести себе як завгодно. У нашій програмі, наприклад, при порожньому вхідному слові МТ зупиняється і видає відповідь 1.

Вище ми привели запис програми в повному, нескороченому вигляді. Тепер же наведемо запис програми в скороченому, більш наочному вигляді:

	0	1	2	3	4	5	6	7	8	9	Λ
q1	.R,	.R,	.R,	.R,	.R,	.R,	.R,	.R,	.R,	.R,	.L,q2
q2	1, !	2, !	3, !	4, !	5, !	6, !	7, !	8, !	9, !	0,L,	1, !

Приклад 2 (аналіз символів)

$A=\{a,b,c\}$. Перенести перший символ непорожнього слова Р в його кінець.

наприклад:

a	b	c	b			→		b	c	b	a	
---	---	---	---	--	--	---	--	---	---	---	---	--

Розв'язок

Для вирішення цього завдання пропонується виконати наступні дії:

1. Запам'ятати перший символ слова Р, а потім стерти цей символ.
2. Перейти вправо під першу порожню клітину за Р і записати в неї символ, який запам'ятали.

Як «бігати» вправо, ми вже знаємо з попереднього прикладу. А ось як запам'ятати перший символ? Адже в МТ немає іншого запам'ятовуючого пристрою, крім стрічки, а запам'ятовувати символ в якійсь клітці на стрічці безглуздо: як тільки автомат зрушиться вліво або вправо від цієї клітини, він тут же забуде даний символ. Що робити?

Вихід тут такий - треба використовувати різні стани автомата. Якщо перший символ - це a , то треба перейти в стан q2, в якому автомат біжить вправо і записує в кінці a . Якщо ж першим був символ b , тоді треба перейти в стан q3, де робиться все те ж саме, тільки в кінці записується символ b . Якщо ж першим був символ c , тоді переходимо в стан q4, в якому автомат дописує за вхідним словом символ c . Отже, те, яким був перший символ, ми фіксуємо переходом автомата в різні стану. Це типовий прийом при програмуванні на МТ.

З урахуванням сказаного програма буде такою:

	a	b	c	Λ	
q1	Λ .R,q2	Λ .R,q3	Λ .R,q4	.R,	аналіз 1-го символу, удаление его, разветвление
q2	.R,	.R,	.R,	a, !	запись a справа
q3	.R,	.R,	.R,	b, !	запись b справа
q4	.R,	.R,	.R,	c, !	запись c справа

Приклад 3 (порівняння символів, стирання слова)

$A = \{a, b, c\}$. Якщо перший і останній символи (непорожнього) слова Р однакові, то це слово не міняти, а інакше замінити його на порожнє слово.

Розв'язок

Для вирішення цього завдання пропонується виконати наступні дії:

1. Запам'ятати перший символ вхідного слова, не стираючи його.
2. Перемістити автомат під останній символ і порівняти його з тим, що запам'ятали. Якщо вони рівні, то більше нічого не робити.

3. В іншому випадку знищити все вхідне слово.

Як запам'ятовувати символ і як переганяти автомат під останній символ слова, ми вже знаємо з попередніх прикладів. Стирання ж вхідного слова реалізується заміною всіх його символів на символ Λ . При цьому, якщо вже автомат опинився в кінці слова, будемо переміщувати автомат справа наліво до першої порожньої клітинки. Ці дії описуються наступною програмою для МТ (нагадаємо, що хрестик в комірці таблиці означає неможливість появи відповідної конфігурації при виконанні програми):

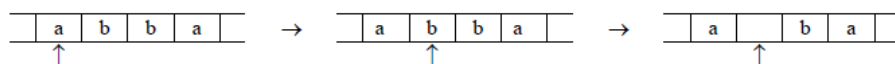
	a	b	c	Λ	
q1	,,q2	,,q4	,,q6	,,!	анализ 1-го символа, разветвление
q2	,R,	,R,	,R,	,L,q3	идти к последнему символу при 1-м символе a
q3	,,!	,,q8	,,q8	×	сравнить посл. символ с a, не равны – на q8 (стереть P)
q4	,R,	,R,	,R,	,L,q5	аналогично при 1-м символе b
q5	,,q8	,,!	,,q8	×	
q6	,R,	,R,	,R,	,L,q7	аналогично при 1-м символе c
q7	,,q8	,,q8	,,!	×	
q8	Λ ,L,	Λ ,L,	Λ ,L,	,,!	стереть всё слово, двигаясь справа налево

Приклад 4 (видалення символу з слова)

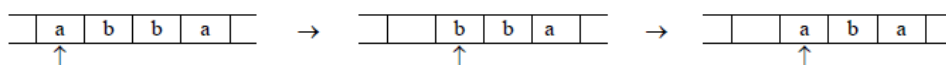
$A = \{a, b\}$. Видалити з слова P його другий символ, якщо такий є.

Розв'язок

Здавалося б, це завдання вирішити просто: треба зрушити автомат під клітинку з другим символом і потім очистити цю клітинку:



Однак нагадаємо, що всередині вихідного слова не повинно бути порожніх клітинок. Тому після видалення другого символу треба «стиснути» слово, перенісши перший символ на одну клітинку вправо. Для цього автомат повинен повернутися до першого символу, запам'ятати його і стерти, а потім, знову зрушившись вправо, записати його в клітинку, де був другий символ. Однак початковий «похід» вправо до другого символу, щоб його стерти, і наступне повернення до першого символу є зайвими діями: яка різниця - переносити перші символ в порожню клітинку або в клітинку з якимось символом? Тому відразу запам'ятовуємо перший символ, стираємо його і записуємо замість другого символу:



Запишемо програму:

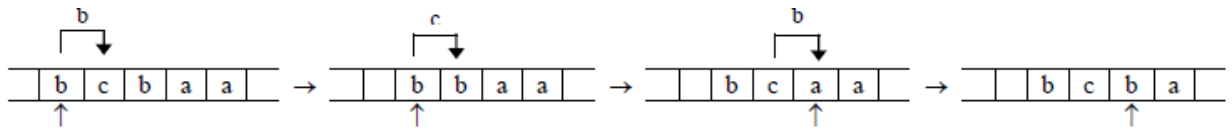
	a	b	Λ	
q1	Λ ,R,q2	Λ ,R,q3	,,!	анализ и удаление 1-го символа, разветвление
q2	,,!	a,!	a,!	замена 2-го символа на a
q3	b,!	,,!	b,!	замена 2-го символа на b

Приклад 5 (стиснення слова)

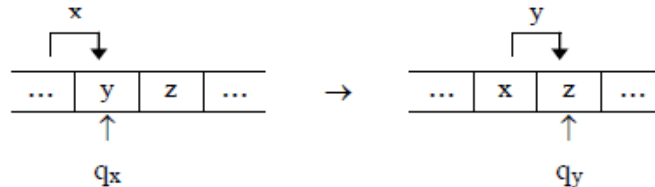
$A = \{a, b, c\}$. Видалити з слова P перше входження символу a, якщо таке є.

Розв'язок

У попередньому прикладі ми переносили на позицію вправо тільки один символ. У даному ж прикладі ми будемо в циклі переміщувати вправо всі початкові символи b і c вхідного слова - до першого символу або до порожньої клітинки:



Центральний момент тут - як перенести символ x з лівої клітинки в наступну клітинку, де знаходиться деякий символ y , щоб потім цей символ можна було перенести в праву клітинку? Якщо через q_x позначити стан, в якому у видиму клітинку треба записати символ x , що знаходився раніше в клітинці зліва, тоді це дію можна зобразити так:



Для цього пропонується виконати такт $\langle x, R, q_y \rangle$, в якому об'єднані наступні три дії: по-перше, у видиму клітинку записується символ x , який взятий з клітинки зліва; по-друге, автомат зсувається вправо - під клітинку, в яку потім треба буде записати щойно замінений символ y ; по-третє, автомат переходить в стан q_y , в якому він і буде виконувати цей запис.

Повторення таких тактів в циклі і призведе до зрушення вправо на одну позицію початкових символів вхідного слова. Цей цикл повинен закінчитися, коли в наступній клітинці виявиться символ a або Λ ($y = a$ або $y = \Lambda$), а на початку циклу можна вважати, що на місце першого символу зліва переноситься символ Λ ($x = \Lambda$). У підсумку виходить наступна програма для МТ:

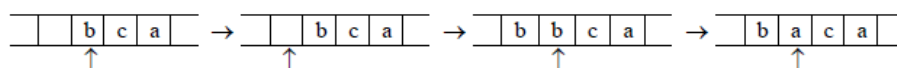
	a	b	c	Λ	
q1	$\Lambda, R, !$	Λ, R, q_2	Λ, R, q_3	$,, !$	q_Λ : стерти 1-й символ и перенести его вправо
q2	$b, , !$	$, R,$	b, R, q_3	$b, , !$	q_b : запись b , перенос ранее видимого символа вправо
q3	$c, , !$	c, R, q_2	$, R,$	$c, , !$	q_c : запись c , перенос ранее видимого символа вправо

Приклад 6 (вставка символу в слово)

$A = \{a, b, c\}$. Якщо P - непорожнє слово, то за його першим символом вставити символ a .

Розв'язок

Ясно, що між першим і другим символами слова P треба звільнити клітинку для нового символу a . Для цього треба перенести на одну позицію вліво перший символ (на старому місці його можна поки що не видаляти), а потім, повернувшись на старе місце, записати символ a :



Перенесення символу на одну позицію вліво аналогічне перенесенню символу вправо, про що говорилося в двох попередніх прикладах, тому наведемо програму для МТ без додаткових коментарів. Відзначимо лише, що в станах q_2, q_3 і q_4 автомат може бачити тільки порожню клітинку, а в стані q_5 він обов'язково бачить перший символ вхідного слова, але не порожню клітинку.

	a	b	c	Λ	
q1	$, L, q_2$	$, L, q_3$	$, L, q_4$	$,, !$	анализ 1-го символа для переноса его влево
q2	\times	\times	\times	a, R, q_5	приписать a слева
q3	\times	\times	\times	b, R, q_5	приписать b слева
q4	\times	\times	\times	c, R, q_5	приписать c слева
q5	$,, !$	$a, , !$	$a, , !$	\times	заменить бывший 1-й символ на a

