

## Лабораторна робота № 3

### "Формальні алгоритмічні системи (ФАС). Машина Тьюрінга (МТ)".

**Мета роботи :** Ознайомлення зі способом зменшення часової складності.

**Зміст роботи:**

#### I. Теоретична частина.

1. Математичні ФАС
2. Структура МТ.
3. Способи зменшення часової складності МТ
4. Обмеженість використання МТ.
5. Послідовність розв'язання задач на МТ.

#### II. Практична частина

##### Практичне заняття

1. Побудувати алгоритм МТ ("слід" МТ, програма) у відповідності до варіанту завдання.
2. Підрахувати часову, програмну та ємкісну складність.

---

#### Лабораторна робота

1. Скласти програму для МТ, користуючись програмою [ALGO2000.EXE](#)
2. Визначити часову, програмну та ємкісну складність алгоритму

#### III. Висновки.

=====

Теза Тьюрінга: Будь-яка обчислювальна функція може бути реалізована на деякій машині Тьюрінга.

#### 1. Математичні ФАС

Основним призначенням математичних ФАС є дослідження проблем розв'язності. Для цієї проблеми вимога елементарності кроку є необхідною. Оскільки ця вимога не може бути математично точно сформульована, вона інтерпретується як умова загальної зрозумілості. Математичні моделі ФАС (вийнятки становлять рекурсивні функції) використовують елементарні операції типу розпізнавання символу, трасування, заміна або зміщення. Всі ці операції нагадують дитячу гру з кубиками, тому можуть вважатись загальнозрозумілими або елементарними.

Прикладом ФАС є машина Тьюрінга.

#### 2. Структура МТ.

Існує низка варіантів детермінованих машин Тьюрінга: однострічкова, багатострічкова, універсальна та ін. Відмінність цих варіантів не принципова, вони зумовлені пошуком способів зменшення часової складності.

Модель однострічкової детермінованої МТ задається шістькою:

$$M = \langle A, Q, q_0, q_f, a_0, p \rangle,$$

де  $A$  – кінцева множина символів зовнішнього алфавіту,

$Q$  – кінцева множина символів внутрішнього алфавіту,

$q_0$  – початковий стан,

$q_f$  – кінцевий стан,

$$q_0, q_f \in Q$$

$a_0$  – позначення порожньої комірки стрічки,

$p$  – така програма, яка не може мати двох команд, у яких би збігалися два перші символи:

$$\{A\} \times \{Q\} \rightarrow \{A\} \{L, R, S\} \{Q\},$$

де  $L$  – зсувати головку вліво,

$R$  – зсувати головку вправо,

$S$  – головка залишається на місці.

Машина Тьюрінга мають одну і ту ж конфігурацію засобів реалізації алгоритму. У конфігурацію входять такі елементи: нескінченна нерухома стрічка, що поділена на окремі комірки, в які можна помістити тільки один символ зовнішнього алфавіту ; рухома головка, яка може стирати, записувати і зчитувати символи зовнішнього алфавіту в комірках стрічки, програма з кінцевою кількістю станів.

Ці елементи і лінії передавання повідомлень, що їх пов'язують, утворюють структуру машини Тьюрінга, яка не залежить від структури алгоритму, що моделюється. [] Ця важлива особливість мМТ дозволяє кількісно порівнювати різні алгоритми з часової, місткісної складності і складності програм.

Машина Тьюрінга як модель алгоритму відповідає визначенню алгоритму. В явному вигляді тут означені всі сім параметрів. Слід машини наочно відображає структуру алгоритму, кількість циклів програми.

Особливості роботи МТ не суперечать властивостям алгоритму. Кроки МТ дискретні і детерміновані, мають властивість масовості. Єдина властивість, яка приймається умовно – це елементарність кроку. У машині Тьюрінга крок алгоритму супроводжується декількома операціями: читання символу в комірці стрічки, пошук необхідної команди, виконання команди – операція зі змістом комірки ( залишити попередній символ, стерти його, записати новий ), операція переміщення головки ( залишити на місці, зсунути ліворуч чи праворуч). Всі ці операції, що складають крок алгоритму, є загальнозрозумілими.

Крок машини Тьюрінга описується виразом  $\{A\} \times \{Q\} \times \{\rightarrow\} \times \{A\} \times \{R, L, S\} \times \{Q\}$ . Звідси, стан це мить, коли читаний із стрічки символ  $a_i$  та новий символ  $q_j$  готові до виконання нової команди.

### *Способи зменшення часової складності МТ*

Часова складність МТ задається послідовністю миттєвих станів машини. Місткісна складність вимірюється кількістю комірок стрічки, яка необхідна для реалізації алгоритму. Складність програми визначається кількістю команд.

Мінімізація часової складності МТ пов'язана з використанням наступних способів:

- зміна розташування початкових даних на стрічці;
- вибір місця розташування проміжних результатів;
- вибір стратегії руху головки;
- вибір початкового положення головки;
- збільшення символів зовнішнього алфавіту;
- застосування паралелізму ( багатострічкова МТ ).

### *Обмеженість використання МТ.*

Наведені способи мінімізації часової складності, крім останнього, не мають практичного значення для комп'ютерної реалізації. МТ є ідеалізованою моделлю алгоритму. Основним пунктом її ідеалізації, як і всіх інших математичних ФАС, є неврахування апаратних витрат, необхідних для реалізації алгоритму. Ця особливість математичних ФАС не дозволяє у повній мірі використовувати досягнення теорії ФАС у проектуванні апаратно-програмних засобів. А у деяких випадках цей недолік приводить до практично неприйнятних висновків. Прикладом тому є теорема про лінійне прискорення.

### *Послідовність розв'язання задач на МТ.*

1. Розміщуються дані на стрічці
2. Визначається необхідність використання додаткових символів і місця їх розташування
3. Розробляється стратегія розв'язання задачі (слід машина Тьюрінга )
4. Будується таблиця програми.
5. У відповідності до сліду машина Тьюрінга розробляється набір команд, які розміщуються в клітинах таблиці.
6. Мінімізується кількість станів (команд) не змінюючи стратегії розв'язання задачі

**Основна гіпотеза теорії алгоритмів:** уточнення змісту алгоритму за допомогою рекурсивних функцій, моделей алгоритму: машини Тьюрінга, нормальних алгоритмів Маркова – еквівалентні один одному. Основу гіпотези складають наступні тези:

**Теза Чьорча:** клас рекурсивно – примітивних функцій співпадає з класом обчислювальних функцій.

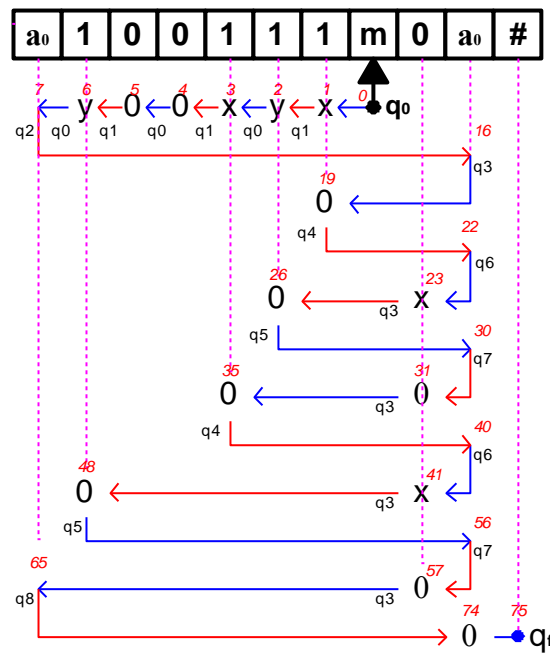
**Теза Тьюрінга:** будь-яка обчислювальна функція може бути реалізована на відповідній машині Тьюрінга.

**Теза Маркова:** будь-який довільний потенційно-здійснюваний процес перероблення слів в деякому алфавіті може бути представлений у вигляді певного нормальних алгоритму.

**Приклади.**

1. Виконати операцію  $(X \bmod 3)$ , де  $X = 100111$ .

Визначити часову (L), програмну (P) та місткісну (M) складність алгоритму.



L=75      M=11

A \ Q	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	q <sub>4</sub>	q <sub>5</sub>	q <sub>6</sub>	q <sub>7</sub>	q <sub>8</sub>	q <sub>9</sub>	q <sub>10</sub>	q <sub>11</sub>
<b>m</b>	Lq <sub>0</sub>		Rq <sub>2</sub>	Lq <sub>3</sub>	Rq <sub>4</sub>	Rq <sub>5</sub>			Rq <sub>8</sub>			
<b>1</b>	xLq <sub>1</sub>	yLq <sub>0</sub>	Rq <sub>2</sub>	Lq <sub>3</sub>		Rq <sub>5</sub>						
<b>0</b>	Lq <sub>1</sub>	Lq <sub>0</sub>	Rq <sub>2</sub>	Lq <sub>3</sub>	Rq <sub>4</sub>	Rq <sub>5</sub>	xLq <sub>3</sub>	yLq <sub>3</sub>	Rq <sub>8</sub>			
<b>a<sub>0</sub></b>	Rq <sub>2</sub>	Rq <sub>2</sub>	Lq <sub>3</sub>	Rq <sub>8</sub>	Lq <sub>6</sub>	Lq <sub>7</sub>			0Rq <sub>11</sub>	1Rq <sub>11</sub>	0Rq <sub>11</sub>	
<b>x</b>			Rq <sub>2</sub>	0Rq <sub>4</sub>	Rq <sub>4</sub>	Rq <sub>5</sub>	yLq <sub>3</sub>	0Lq <sub>3</sub>	0Rq <sub>9</sub>			
<b>y</b>			Rq <sub>2</sub>	0Rq <sub>5</sub>	Rq <sub>4</sub>	Rq <sub>5</sub>	0Lq <sub>3</sub>	xLq <sub>3</sub>	1Rq <sub>10</sub>			
<b>#</b>												q <sub>f</sub>

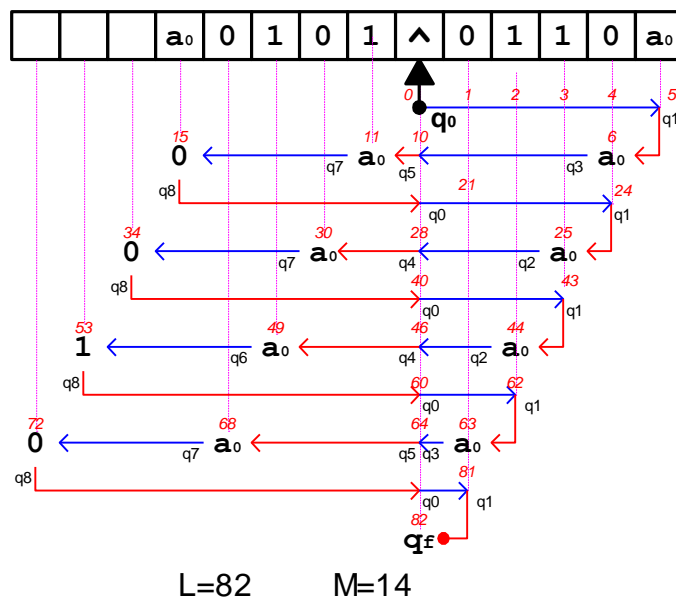
**P = 44**

- В алгоритмі використано наступну властивість :

Якщо  $C = A + B$ , то  $C \bmod 3 = (A \bmod 3 + B \bmod 3) \bmod 3$

2. Виконати операцію кон'юнкції:  $(X \wedge Y)$ , де  $X = 0101$ ,  $Y = 0110$ .

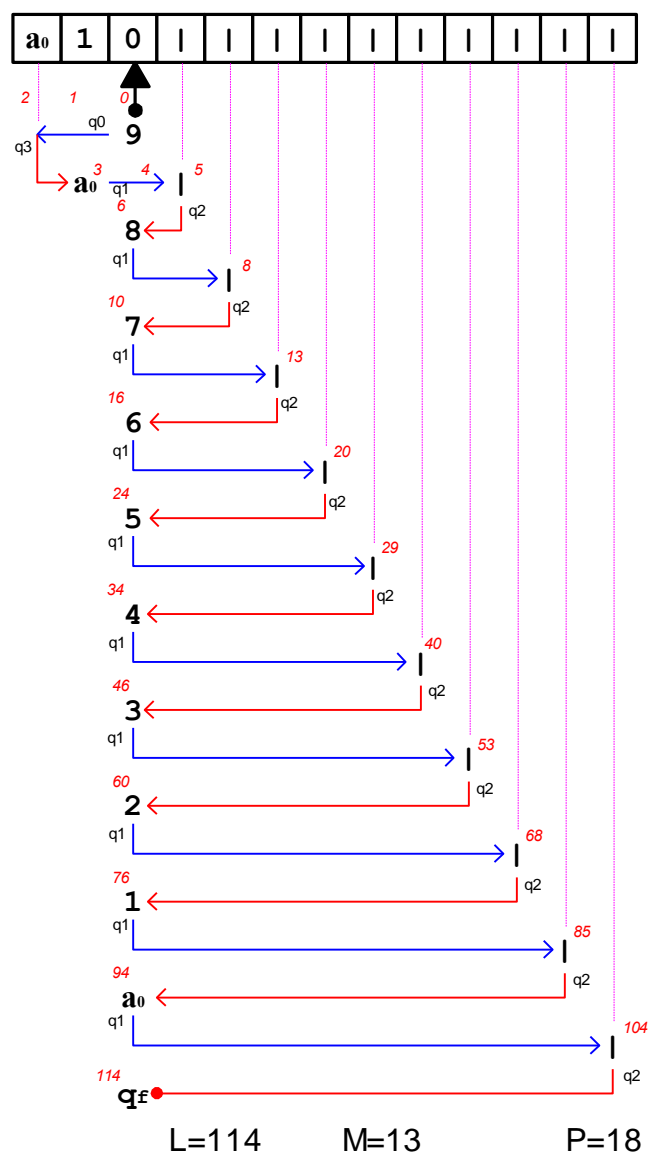
Визначити часову (L), програмну (P) та місткісну (M) складність алгоритму.



A	Q	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>	q <sub>4</sub>	q <sub>5</sub>	q <sub>6</sub>	q <sub>7</sub>	q <sub>8</sub>
a <sub>0</sub>		Lq <sub>1</sub>				Lq <sub>4</sub>	Lq <sub>5</sub>	1Rq <sub>8</sub>	0Rq <sub>8</sub>	Rq <sub>8</sub>
0		Rq <sub>0</sub>	a <sub>0</sub> Lq <sub>3</sub>	Lq <sub>2</sub>	Lq <sub>3</sub>	a <sub>0</sub> Lq <sub>7</sub>	a <sub>0</sub> Lq <sub>7</sub>	Lq <sub>6</sub>	Lq <sub>7</sub>	Rq <sub>8</sub>
1		Rq <sub>0</sub>	a <sub>0</sub> Lq <sub>2</sub>	Lq <sub>2</sub>	Lq <sub>3</sub>	a <sub>0</sub> Lq <sub>6</sub>	a <sub>0</sub> Lq <sub>7</sub>	Lq <sub>6</sub>	Lq <sub>7</sub>	Rq <sub>8</sub>
^		Rq <sub>0</sub>	q <sub>f</sub>	Lq <sub>4</sub>	Lq <sub>5</sub>					Rq <sub>0</sub>

**P = 29**

3. Виконати операцію перевodu формату числа із десятичного в унарний :  $X_{(10)} \rightarrow Y_{(1)}$ , де  $X=10$ .  
Визначити часову (L), програмну (P) та місткісну (M) складність алгоритму.



A	Q	q <sub>0</sub>	q <sub>1</sub>	q <sub>2</sub>	q <sub>3</sub>
0	9Lq <sub>0</sub>				
1	Lq <sub>0</sub>			a <sub>0</sub> Rq <sub>1</sub>	a <sub>0</sub> Rq <sub>1</sub>
2				1Rq <sub>1</sub>	
3				2Rq <sub>1</sub>	
4				3Rq <sub>1</sub>	
5				4Rq <sub>1</sub>	
6				5Rq <sub>1</sub>	
7				6Rq <sub>1</sub>	
8				7Rq <sub>1</sub>	
9				8Rq <sub>1</sub>	
a <sub>0</sub>	Rq <sub>3</sub>	Lq <sub>2</sub>	q <sub>f</sub>		
	Rq <sub>1</sub>	Rq <sub>1</sub>	Lq <sub>2</sub>		



Розташування даних довільне.

Без збереження вхідних даних.

Без збереження вхідних даних.

Зі збереженням вхідних даних.

Зі збереженням вхідних даних.

Результат розташувати на місці вхідних даних

Без збереження вхідних даних. Числа представлені в прямому коді.

Без збереження вхідних даних. Числа представлені в прямому коді.

		x <sub>3</sub>	x <sub>2</sub>	x <sub>1</sub>	x <sub>0</sub>	-	y <sub>3</sub>	y <sub>2</sub>	y <sub>1</sub>	y <sub>0</sub>	=	z <sub>3</sub>	z <sub>2</sub>	z <sub>1</sub>	z <sub>0</sub>					
--	--	----------------	----------------	----------------	----------------	---	----------------	----------------	----------------	----------------	---	----------------	----------------	----------------	----------------	--	--	--	--	--

[illegible]

							x1	x0	-	y1	y0	=	z1	z0									
--	--	--	--	--	--	--	----	----	---	----	----	---	----	----	--	--	--	--	--	--	--	--	--



						x1	x0	-	y1	y0	=		z1	z0						
--	--	--	--	--	--	----	----	---	----	----	---	--	----	----	--	--	--	--	--	--



						x0	=			.	.	.							
--	--	--	--	--	--	----	---	--	--	---	---	---	--	--	--	--	--	--	--

[illegible][illegible][illegible]



Розташування даних довільне.

Розташування даних довільне.



$X < 10, Y < 10$  Розташування результату довільне.



- початкове положення рухомої головки задано



- виберіть початкове положення рухомої головки таким чином, щоб часова складність була найменшою