Практичне заняття № 6

Формальні алгоритмічні системи (ФАС). Машина Тюрінга (МТ).

1. Математичні ФАС

Основним призначенням математичних ***формальних алгоритмічних систем*** (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 (винятки 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 Тюрінга.

2. Структура МТ.

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

Модель одностврічкової детермінованої МТ задається шісткою:

$$\mathbf{M = < A, Q, q0, qf, a0, p >,}$$

де A – кінцева множина символів зовнішнього алфавіту,
Q – кінцева множина символів внутрішнього алфавіту,
q0 – початковий стан,
qf – кінцевий стан,
q0, qf $\in$ Q
a0 – позначення порожньої комірки стрічки,
p – така програма, яка не може мати двох команд, у яких би збігалися два перші символи:

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

де L – зсувати головку вліво,
R - зсувати головку вправо,
S – головка залишається на місці.

Машини Тюрінга мають одну і ту ж конфігурацію засобів реалізації алгоритму. У конфігурацію входять такі елементи: нескінчена 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 алфавиту в комірках стрічки, програма з кінцевою кількістю станів.

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

Машина Тюрінга 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 Тюрінга 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 Тюрінга описується виразом {A}x{Q}x{→}x{A}x{R,L,S}x{Q}. Звідси, стан це мить, коли читаний із стрічки символ a_I та новий символ q_J готові до виконання нової команди.

3. 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 алгоритму руху головки;
- вибір початкового положення головки;
- збільшення символів зовнішнього алфавіту;
- застосування паралелізму (багатострічкова МТ).

4. Обмеженність 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.

5. Послідовність розв’язання задач на МТ.

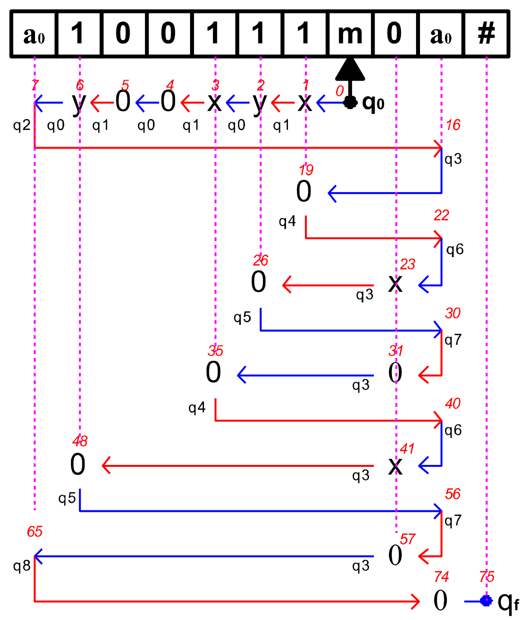
1. Розміщуються дані на стрічці
2. Визначається необхідність використання додаткових символів і місця їх розташування
3. Розробляється алгоритм розв’язання задачі (формується “слід” машини Тюрінга)
4. Будується таблиця програми.
5. У відповідності до сліду машина Тюрінга розробляється набір команд, які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

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

- ***Теза Чьорча***: клас рекурсивно-примітивних функцій співпадає з класом обчислювальних функцій;
- ***Теза Тюрінга***: будь-яка обчислювальна функція може бути реалізована на відповідній машині Тюрінга;
- ***Теза Маркова:*** будь-який довільний потенційно-здійснюваний процес пероброблення слів в деякому алфавіті може бути представлений у вигляді певного нормального алгоритму.

Приклади:

1. Виконати операцію (X mod 3), де X= 100111.
   Визначити часову (L), програмну (P) та місткісну (M) складність алгоритму.



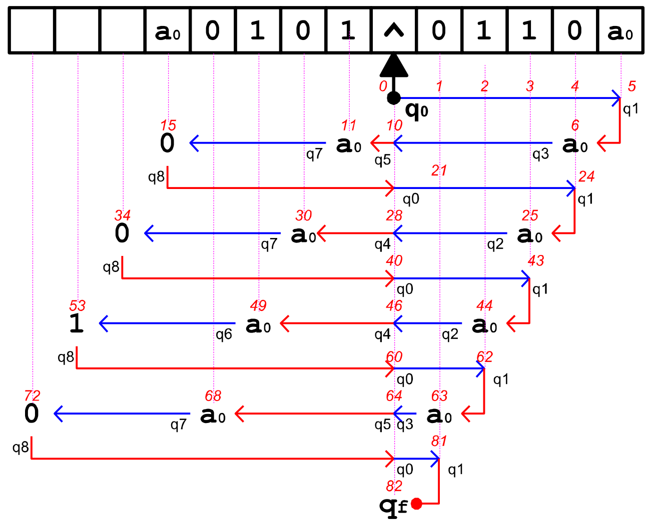
L=75 M=11

Q / A	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}
m	Lq_0		Rq_2	Lq_3	Rq_4	Rq_5			Rq_8			
1	xLq_1	yLq_0	Rq_2	Lq_3		Rq_5						
0	Lq_1	Lq_0	Rq_2	Lq_3	Rq_4	Rq_5	xLq_3	yLq_3	Rq_8			
a_0	Rq_2	Rq_2	Lq_3	Rq_8	Lq_6	Lq_7			$0Rq_{11}$	$1Rq_{11}$	$0Rq_{11}$	
x			Rq_2	$0Rq_4$	Rq_4	Rq_5	yLq_3	$0Lq_3$	$0Rq_9$			
y			Rq_2	$0Rq_5$	Rq_4	Rq_5	$0Lq_3$	xLq_3	$1Rq_{10}$			
#												q_f

P = 44

В алгоритмі використано наступну властивість: якщо C = A + B, то C mod3 =(A mod3 + B mod3) mod3

2. Виконати операцію кон'юнкції: $(X \wedge Y)$, де $X = 0101$, $Y = 0110$.
   Визначити часову (L), програмну (P) та місткісну (M) складність алгоритму.

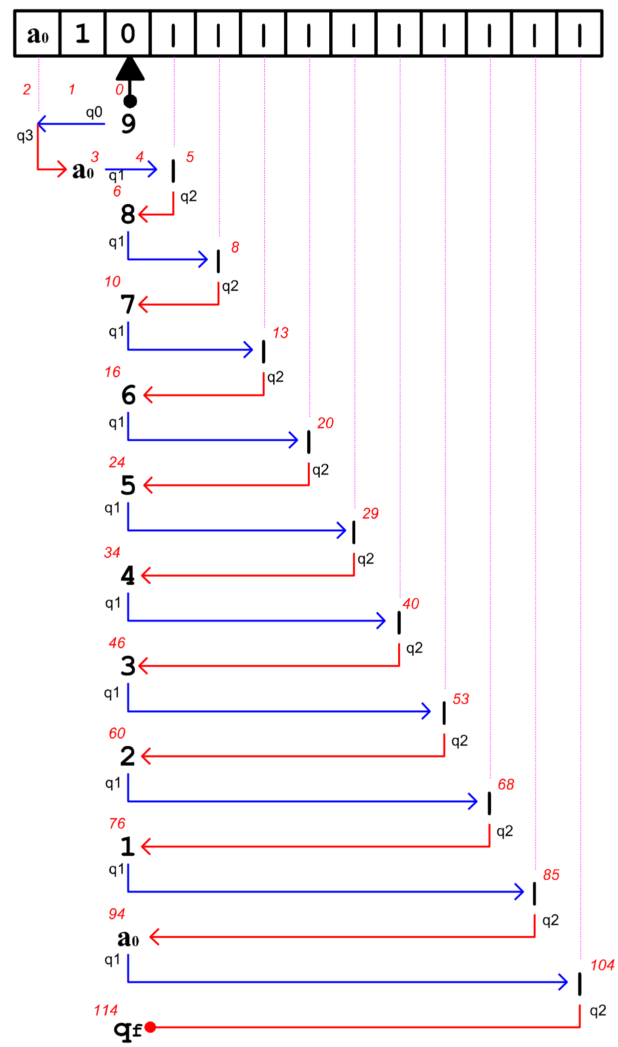


L=82 M=14

A \ Q	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
a_0	Lq_1				Lq_4	Lq_5	$1Rq_8$	$0Rq_8$	Rq_8
0	Rq_0	a_0Lq_3	Lq_2	Lq_3	a_0Lq_7	a_0Lq_7	Lq_6	Lq_7	Rq_8
1	Rq_0	a_0Lq_2	Lq_2	Lq_3	a_0Lq_6	a_0Lq_7	Lq_6	Lq_7	Rq_8
^	Rq_0	q_f	Lq_4	Lq_5					Rq_0

P = 29

3. Виконати операцію перетворення числа із десяткової форми в унарну: X(10) → Y(1), де X= 10. Визначити часову (L), програмну (P) та місткісну (M) складність алгоритму.



L=114 M=13 P=18

A \ Q	q_0	q_1	q_2	q_3
0	$9Lq_0$			
1	Lq_0		a_0Rq_1	a_0Rq_1
2			$1Rq_1$	
3			$2Rq_1$	
4			$3Rq_1$	
5			$4Rq_1$	
6			$5Rq_1$	
7			$6Rq_1$	
8			$7Rq_1$	
9			$8Rq_1$	
a_0	Rq_3	$\vert Lq_2$	q_f	
$\vert$	Rq_1	Rq_1	Lq_2	