

---

UNIVERSITI SAINS MALAYSIA

Second Semester Examination  
2010/2011 Academic Session

April/May 2011

**CPT443/CPT347 – Automata Theory & Formal Languages**  
**[Teori Automata & Bahasa Formal]**

Duration : 3 hours  
[Masa : 3 jam]

---

**INSTRUCTIONS TO CANDIDATE:**  
**[ARAHAN KEPADA CALON:]**

- Please ensure that this examination paper contains **FIVE** questions in **SEVEN** printed pages before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi **LIMA** soalan di dalam **TUJUH** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions.

*[Jawab **SEMUA** soalan.]*

- This is an 'Open Book' Examination.

*[Peperiksaan ini akan dijalankan secara 'Open Book'.]*

- You may answer the questions either in English or in bahasa Malaysia.

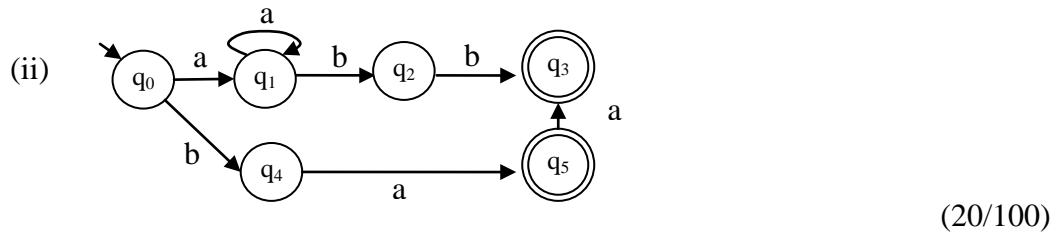
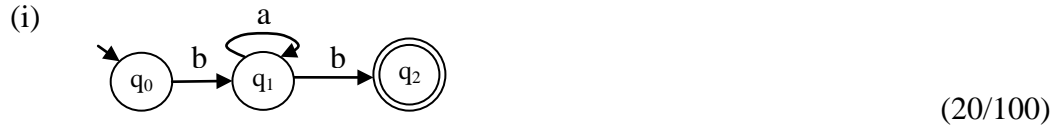
*[Anda dibenarkan menjawab soalan sama ada dalam bahasa Inggeris atau bahasa Malaysia.]*

- In the event of any discrepancies, the English version shall be used.

*[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi bahasa Inggeris hendaklah diguna pakai.]*

1. Answer 1(a), 1(b) and 1(c) below:

(a) Convert the following finite state machines to regular expressions, and give an example of the string it accepts:



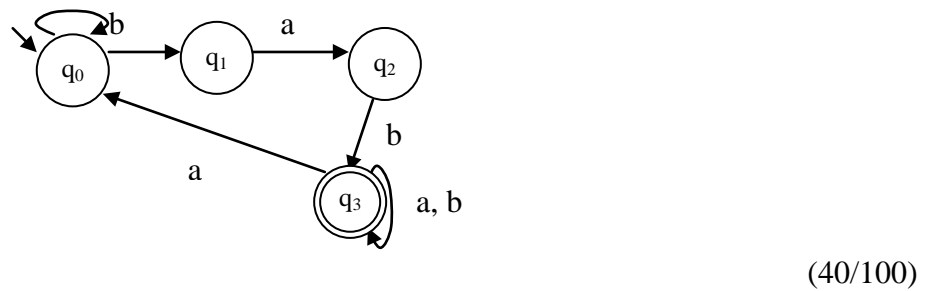
(b) Simplify the regular expressions below:

(i)  $a^+ a^*(b \cup \emptyset) (a \cup b)^*$

(ii)  $b (a^+ \cup ba^*) \cup bab^*$

(20/100)

(c) Convert the following finite state machine to regular grammar:



2. Let  $L_1 = \{w \in \{a, b, c \dots z\}^* : w \text{ are words starting with prefix "mem"}\}$  and  $L_2 = \{w \in \{a, b, c, \dots z\}^* : w \text{ are words which end with suffix "kan"}\}$

(a) Construct a deterministic finite state machine for  $L_1$  and  $L_2$ . (40/100)

(b) Give an example of Malay word in the set  $L' = L_1 - L_2$ . (10/100)

(c) Use the closure property of regular language, and the deterministic finite state machine you have constructed for  $L_1$  and  $L_2$  to build an automaton for  $L' = L_1 - L_2$ . (50/100)

3. A Java object can be defined as below:

```
public class Example{
    public int method(){
        int value = 10;
        return value;
    }
}
```

(a) Design a suitable automaton to verify that the curly brackets are balanced. (30/100)

(b) Design a suitable automaton to verify the name of the variable (e.g. the variable *value* in the example above). Note that variable name in Java is an unlimited-length sequence of Unicode letters and digits, beginning with a letter, the dollar sign "\$", or the underscore character "\_". (30/100)

(c) Propose a way (using the appropriate automaton or grammar) to check the validity of the declaration of the variables. Assume the variables can only be any of the primitive data type (e.g. boolean, char, int, float and double). (40/100)

4.  $L(G) = \{ \{S, a, b\}, \{a, b\}, R, S \}$ , where R is given by:

$$S \rightarrow a T V a \mid a a U$$

$$T \rightarrow a T a \mid b T c \mid \varepsilon \mid V$$

$$U \rightarrow a U b$$

$$V \rightarrow c V c \mid \varepsilon$$

- (a) Simplify the rules for the context-free grammar above. (20/100)
- (b) Show a parse tree for the string 'aacca'. (20/100)
- (c) Convert the rules to Chomsky normal form. (30/100)
- (d) Use the Cocke-Kasami-Younger (CKY) algorithm to determine if the string 'abca' can be generated from the grammar G. (30/100)

5. Answer both 5(a) and 5(b) below:

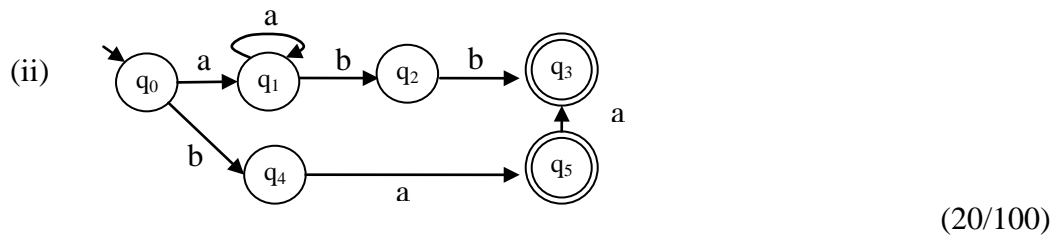
- (a) Draw the changes in the content of the tape(s) of a Turing machine using an appropriate example to show how it compares two binary natural numbers, and return the one with the highest value. The procedure shown must also be workable for other binary natural numbers. (40/100)
- (b) Draw a Turing machine for each question below:
- (i)  $L = \{ w \in \{a, b, c, d\}^* : w \text{ consists of strings with the following pattern } a^n b^n c^n d^n \}$ . (30/100)
- (ii)  $L = \{ w \in \{0, 1, +, =\}^* : w \text{ consists of strings with the following pattern } x+y=z, \text{ where } x, y \text{ and } z \text{ are binary strings, which have a length of three. Example: } 000+000=100, 001+010=100, \dots \}$ . (30/100)

**KERTAS SOALAN DALAM VERSI BAHASA MALAYSIA**

[CPT443/CPT347]

1. Jawab soalan 1(a), 1(b) dan 1(c) berikut:

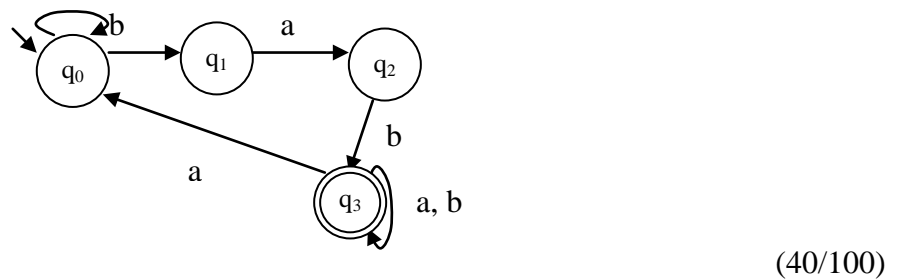
(a) Tukarkan mesin keadaan terhingga yang berikut kepada ungkapan nalar, dan beri satu contoh rangkaian yang diterimanya:



(b) Mudahkan ungkapan nalar berikut:

- (i)  $a^+ a^*(b \cup \emptyset) (a \cup b)^*$
  - (ii)  $b (a^+ \cup ba^*) \cup bab^*$
- (20/100)

(c) Tukarkan mesin keadaan terhingga yang berikut kepada nahu nalar:



2. Katakan  $L_1 = \{w \in \{a, b, c \dots z\}^* : w \text{ ialah perkataan yang bermula dengan prefix "mem"}\}$  dan  $L_2 = \{w \in \{a, b, c, \dots z\}^* : w \text{ ialah perkataan yang berakhir dengan suffix "kan"}\}$

(a) Bina satu mesin keadaan terhingga penentu untuk  $L_1$  dan  $L_2$ .

(40/100)

(b) Beri satu contoh perkataan bahasa Melayu yang terdapat dalam set  $L' = L_1 - L_2$ .

(10/100)

(c) Gunakan ciri-ciri tertutup bahasa nalar dan mesin keadaan terhingga penentu yang kamu bina untuk  $L_1$  and  $L_2$  sebelum ini untuk membina satu mesin keadaan terhingga untuk  $L' = L_1 - L_2$ .

(50/100)

3. Satu objek Java boleh didefinisikan seperti berikut:

```
public class Example{
    public int method(){
        int value = 10;
        return value;
    }
}
```

(a) Bina satu automata yang sesuai untuk mengesahkan yang kurungan kerinting adalah seimbang.

(30/100)

(b) Bina satu automata yang sesuai untuk mengesahkan nama pemboleh ubah (contohnya pemboleh ubah *value* dalam contoh di atas). Ambil perhatian bahawa nama pemboleh ubah dalam Java ialah satu urutan panjang tidak terhad bagi huruf dan digit, bermula dengan huruf, tanda dolar "\$", atau aksara "\_".

(30/100)

(c) Cadangkan satu cara (sama ada menggunakan automata atau tatabahasa) untuk memeriksa kesahihan deklarasasi bagi pemboleh ubah. Andaikan bahawa pemboleh ubah hanya boleh terdiri daripada jenis data primitif sahaja (contoh: boolean, char, int, float and double).

(40/100)

4.  $L(G) = \{ \{S, a, b\}, \{a, b\}, R, S \}$ , di mana R diberikan sebagai:

$$S \rightarrow a T V a \mid a a U$$

$$T \rightarrow a T a \mid b T c \mid \varepsilon \mid V$$

$$U \rightarrow a U b$$

$$V \rightarrow c V c \mid \varepsilon$$

- (a) Mudahkan petua untuk nahu bebas konteks di atas. (20/100)
- (b) Tunjukkan satu pohon huraian untuk rangkaian 'aaacca'. (20/100)
- (c) Tukarkan petua di atas ke bentuk normal Chomsky. (30/100)
- (d) Gunakan algoritma Cocke-Kasami-Younger (CKY) untuk menentukan sama ada rangkaian 'abca' boleh dijana daripada G nahu di atas. (30/100)

5. Jawab kedua-dua soalan 5(a) dan 5(b) di bawah:

- (a) Lakarkan perubahan dalam kandungan pita bagi satu mesin Turing dengan menggunakan contoh yang sesuai untuk menunjukkan bagaimana ia membandingkan dua nilai binari, dan hasil nilai binari terbesar. Prosedur yang ditunjukkan mestilah boleh juga digunakan untuk nilai binari lain. (40/100)
- (b) Lakarkan satu mesin Turing untuk setiap soalan di bawah:
- (i)  $L = \{ w \in \{a, b, c, d\}^* : w \text{ terdiri daripada rangkaian dengan corak berikut } a^n b^n c^n d^n \}$ . (30/100)
- (ii)  $L = \{ w \in \{0, 1, +, =\}^* : w \text{ terdiri daripada rangkaian dengan corak } x+y=z, \text{ di mana } x, y \text{ dan } z \text{ ialah rangkaian binari yang mempunyai tiga huruf panjang. Contoh: } 000+000=100, 001+010=100, \dots \}$ . (30/100)