Theories of Computation

Solutions

Mock Examination

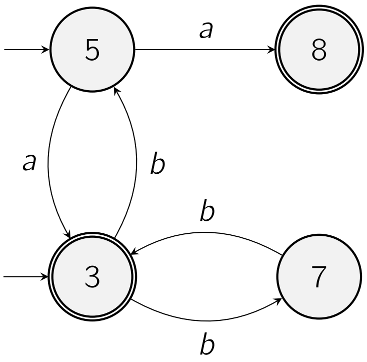
Theories of Computation

Answer ALL questions. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

Mock Exam paper

Question 1 : Automata and Regular Languages

(a) The following is an NFA over the alphabet $\{a, b\}$.



Determinize it using the algorithm taught in lectures. (The DFA that you give need not be total.) Briefly explain the initial state of your DFA, and the a-labelled transition from it. **[7 marks]**

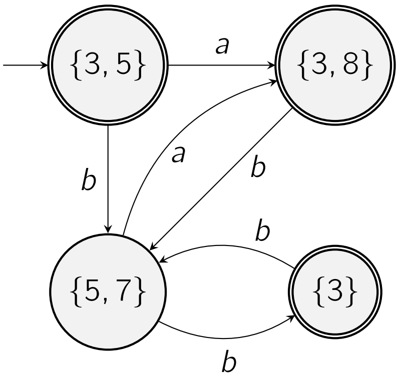
(b) Give a deterministic finite automaton that accepts those words over the alphabet $\{3, 4\}$ for which the sum of all the digits is a multiple of 4.

For example, it should accept ϵ (the empty word) and 44 and 33433, but not 33333. The initial state and final state(s) should be indicated.

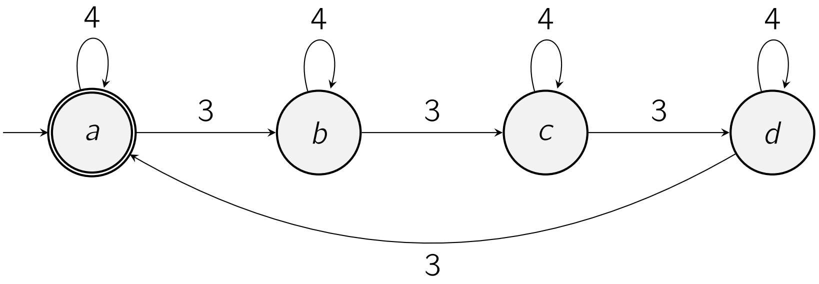
Briefly explain why your automaton is correct. **[8 marks]**

Solutions

(a) Since the NFA has initial states 3 and 5, the DFA has initial state $\{3, 5\}$. Since 5 has a-transitions leading to 3 and 8, and 3 has no a-transitions, the state $\{3, 5\}$ has an a-transition leading to $\{3, 8\}$.



(b) The following finite automaton accepts those words over the alphabet $\{3, 4\}$ for which the sum of all the digits is a multiple of 4:



The automaton accepts any combination of 4's, but if it sees the number 3, it moves to the next state until it has seen 4 instances of 3's (to keep sum divisible by 4). Otherwise, it remains in one of the non-accepting states.

Question 2 : Lambda Calculus and Context-free Languages

(a) Consider λ-calculus with arithmetic, using the following types:

$$A \quad ::= \quad A \to A \mid \texttt{int} \mid (A)$$

In this calculus, here is a term:

$$\lambda y.\, \lambda z.\, \lambda w.\, yz + yw$$

What is the most general type of this term? Explain your answer, giving a suitable type annotation for each bound variable. **[7 marks]**

(b) Consider the following grammar for generating function prototypes in a Java-like language. The alphabet for this language contains the following terminals (words and symbols):

Σ = {public, private, static, void, int, float, double,
main, compute, x, y, z, (,), , , ;}.

$$\begin{array}{rcl}
\textbf{Start} & ::= & \textbf{AccMod}\ \ \textbf{Type}\ \ \textbf{FunName}(\textbf{ArgList}); \\
\textbf{AccMod} & ::= & \mathit{public}\ \ \textbf{AccMod} \mid \mathit{private}\ \ \textbf{AccMod} \mid \textbf{AccMod}\ \ \mathit{static} \mid \varepsilon \\
\textbf{Type} & ::= & \mathit{void} \mid \mathit{int} \mid \mathit{float} \mid \mathit{double} \\
\textbf{FunName} & ::= & \mathit{main} \mid \mathit{compute} \\
\textbf{ArgName} & ::= & x \mid y \mid z \\
\textbf{ArgList} & ::= & \textbf{Arg} \mid \textbf{ArgList},\ \ \textbf{Arg} \\
\textbf{Arg} & ::= & \textbf{Type}\ \ \textbf{ArgName}
\end{array}$$

Show that this grammar is ambiguous by drawing two derivation trees for the following prototype:

public static double compute(float y, double z);

Note: All of the variables (non-terminals) are shown in **bold** in the above grammar. **[8 marks]**

Solutions

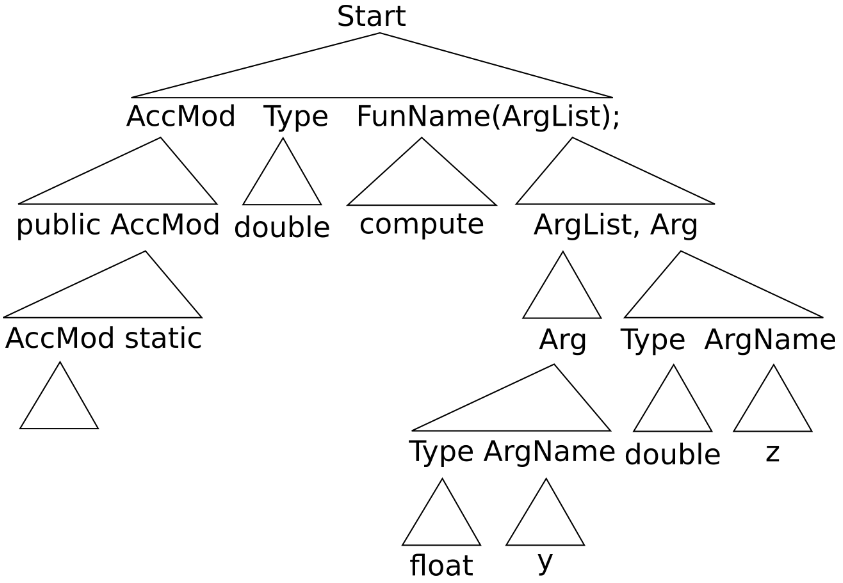
(a) Since y is applied to z, it has type $A \to B$ and z has type A. Also, y is applied to w, so w has type A. Thus yz and yw have type B, but they are added, so $B = \texttt{int}$ and the type of $yz + yw$ is also $\texttt{int}$. So the annotated term is

$$\lambda y_{A \to \texttt{int}}.\, \lambda z_A.\, \lambda w_A.\, yz + yw$$

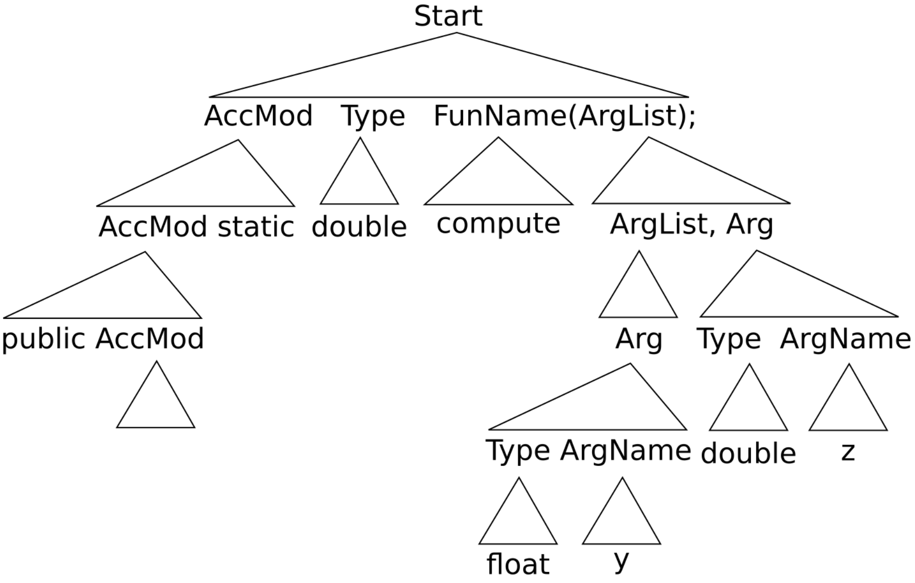
which has the most general type

$$(A \to \texttt{int}) \to \texttt{int} \to \texttt{int} \to \texttt{int}$$

(b) The first possible derivation tree for the given function prototype is shown below:



The second possible derivation tree for the given function prototype is shown below:



Therefore, the given grammar is ambiguous!

Question 3 : Complexity and Computability

(a) Recall that "$f(n)$ is $O(g(n))$" means that there are numbers M and C such that, for all $n \geq M$, we have $f(n) \leq C \times g(n)$. For any $n \in \mathbb{N}$, let $h(n)$ be the maximum of $2n^3 - 7n - 5$ and $n + 8$. Show that $h(n)$ is $O(n^3)$. **[7 marks]**

(b) Recall that Primitive Java is a language with the type `nat` and the following basic facilities.

- `nat i = 0`
- `i++`
- `i--`, which does nothing if `i==0`
- `if (i == 0)` $\{M\}$ `else` $\{N\}$
- `repeat i times` $\{M\}$

The following facilities are derivable from the above basic facilities and may be used in answering this question.

- `nat j = i`
- `j = 0`
- `j = i`
- `if (i <= j)` $\{M\}$ `else` $\{N\}$
- `if (i < j)` $\{M\}$ `else` $\{N\}$
- `if (i == j)` $\{M\}$ `else` $\{N\}$
- `i = j + k`
- `i = max(j-k,0)`

To show that the squaring function is primitive recursive, give a Primitive Java encoding of:

$$\texttt{i = j}^2$$

[8 marks]

Solutions

(a) For $n > 0$, the quotient $\frac{h(n)}{n^3}$ is the maximum of $\frac{2n^3-7n-5}{n^3}$, which is ≤ 2, and $\frac{n+8}{n^3} = \frac{1}{n^2} + \frac{8}{n^3}$, which is ≤ 9, so $\frac{h(n)}{n^3} \leq 9$. Hence $h(n)$ is $O(n^3)$.

Alternative solution: we show that, for $n \geq 3$, we have $2n^3 - 7n - 5 \geq n + 8$, which implies $h(n) = 2n^3 - 7n - 5$ and therefore $\frac{h(n)}{n^3} = \frac{2n^3-7n-5}{n^3} \leq 2$. One way to show this is to use calculus. Another is to put $n = m + 3$, and then we have

$$\begin{aligned}
(2n^3 - 7n - 5) - (n + 8) &= [2(m+3)^3 - 7(m+3) - 5] - [(m+3) + 8] \\
&= [2(m^3 + 3 \times 3m^2 + 3 \times 3^2 m + 3^3) - (7m + 26)] - (m + 11) \\
&= 2(m^3 + 9m^2 + 27m + 27) - (8m + 37) \\
&= 2m^3 + 18m^2 + 46m + 17 \\
&\geq 17 \qquad \text{since } m \geq 0.
\end{aligned}$$

(b) The Primitive Java encoding is given below:

```
i = 0;
repeat j times {
  i = i + j;
}
```

Question 4 : Fancy Turing Machines

Jim designs a Turing machine for the input alphabet {a, b} using extra symbols

c, d, e, f, g, h, k, m, n

as well as the blank symbol (␣). Marie wants to convert Jim's machine to a Turing machine that does not use the extra symbols. She adopts the following convention. Characters are translated as follows:

Jim's machine	**Marie's machine**
a	aaa
b	aab
c	aa␣
d	aba
e	abb
f	ab␣
g	a␣a
h	a␣b
k	a␣␣
m	baa
n	bab
␣	␣␣␣

The head of Marie's machine is the middle of the three symbols corresponding to the symbol on Jim's machine where the head is located.

(a) Complete the following table of tape-configurations. The head position is indicated by a dot over the symbol.

Jim's machine	**Marie's machine**
ȧhf␣f	
	a␣␣a␣␣a␣␣␣␣␣␣␣̇␣

[6 marks]

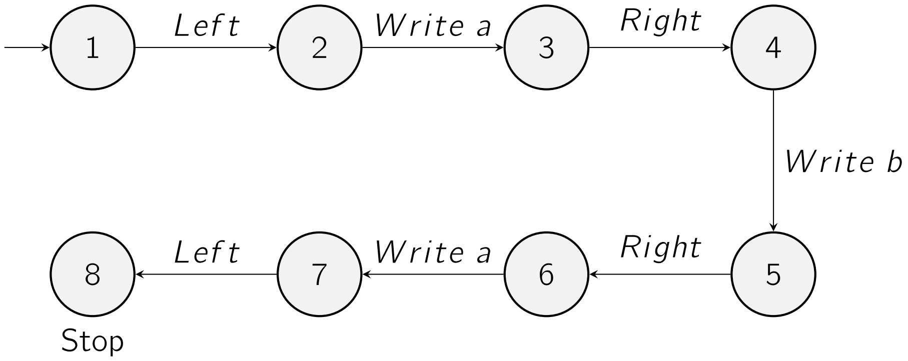
(b) Give a program for Marie's machine that simulates the instruction "Write d" on Jim's machine. Your program must have running time $O(1)$, though you are not required to show this. **[9 marks]**

Solutions

(a) Table of tape-configurations is shown below:

Jim's machine	Marie's machine
ȧhf␣f	aȧaa␣bab␣␣␣␣ab␣
kkk␣␣̇	a␣␣a␣␣a␣␣␣␣␣␣␣̇␣

(b) A program for Marie's machine that simulates the "Write `d`" instruction on Jim's machine is given below:


1
Left
2
Write a
3
Right
4
Write b
5
Right
6
Write a
7
Left
8
Stop