The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, SPRING SEMESTER 2017–2018

LANGUAGES AND COMPUTATION

ANSWERS

Time allowed TWO hours

Candidates may complete the front cover of their answer book and sign their desk card but must NOT write anything else until the start of the examination period is announced.

Answer ALL THREE questions

No calculators are permitted in this examination.

Dictionaries are not allowed with one exception. Those whose first language is not English may use a standard translation dictionary to translate between that language and English provided that neither language is the subject of this examination. Subject-specific translation directories are not permitted.

No electronic devices capable of storing and retrieving text, including electronic dictionaries, may be used.

Note: ANSWERS

Question 1
The following questions are multiple choice. There is at least one correct answer, but there may be several. To get all the marks you have to list all correct answers and none of the incorrect ones. 1 mistake results in 3 marks, 2 mistakes result in 1 mark, 3 or more mistakes result in zero marks.

Answer: *Note that the answer that should be provided is just a list of the correct alternative(s). Any further explanations below are just for clarification.*

(a) Which of the following statements are correct?

(i) An alphabet is a finite sequence of distinct symbols.

(ii) A language is the set of all possible words over a given alphabet.

(iii) A language is always an infinite set of words.

(iv) A regular language is always infinite.

(v) An infinite language can be regular.

(5)

Answer: *Correct: v*

Incorrect:

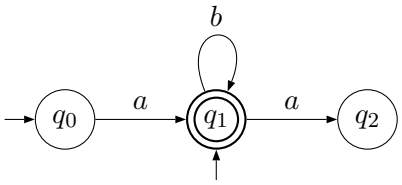
i An alphabet is a set *of symbols, not a sequence.*

ii A language is any *set of words over a given alphabet.*

iii A language can be finite.

iv There are both finite and infinite regular languages. (In fact, any finite language is regular.)

(b) Consider the following nondeterministic finite automaton (NFA) A over $\Sigma = \{a, b\}$:



Which of the following statements about A and the equivalent deterministic finite automaton (DFA) obtained through the subset construction are correct? Consider the entire DFA, even if some states are not reachable.

(i) $\{q_0, q_1\}$ is the initial state of the equivalent DFA.

(ii) Each of $\{q_1\}$, $\{q_2\}$, $\{q_1, q_2\}$ is an accepting state in the equivalent DFA

(iii) The transition function of the equivalent DFA has a transition from the state $\{q_1, q_2\}$ on the symbol b to the state $\{q_1\}$.

(iv) The transition function of the equivalent DFA has a transition from the state $\{q_2\}$ on the symbol a to the state $\emptyset$.

(v) The state $\emptyset$ is a *dead state* of the resulting DFA: no accepting state can be reached from it.

(5)

Answer: *Correct: i, iii, iv, v*

Incorrect:

ii The state $\{q_2\}$ is not accepting.

(c) Consider the following Context-Free Grammar (CFG) G:

$$\begin{array}{lcl} S & \rightarrow & XX \mid Y \\ X & \rightarrow & aXc \mid aYc \\ Y & \rightarrow & Yb \mid \epsilon \end{array}$$

where S, X, Y are nonterminal symbols, S is the start symbol, and a, b, c are terminal symbols.

Which of the following statements about G are correct?

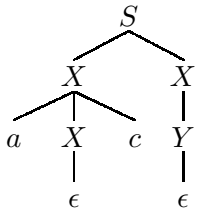
(i) $S \Rightarrow XX \Rightarrow aYcX \Rightarrow acX \Rightarrow acaYc \Rightarrow acac$ is a *left-most* derivation in the grammar G.

(ii) $S \Rightarrow XX \Rightarrow XaYc \Rightarrow Xac \Rightarrow aYcac \Rightarrow acac$ is a *right-most* derivation in the grammar G.

(iii) G is ambiguous.

(iv) $aXcbbb$ is a sentential form for grammar G.

(v) The following is a derivation tree in the grammar G:



(5)

Answer: *Correct: i, ii;*

Incorrect:

iii The grammar is unambiguous.

iv No; unless the word consists solely of b's, any string of b's must be preceded by at least one a and followed by equally many c's.

v No; there is no production $X \rightarrow Y$, which is implied by the tree.

(d) Which of the following statements about Complexity Theory is certainly true?

(i) The Satisfiability Problem SAT is NP-complete.

(ii) The Halting Problem is reducible to SAT.

(iii) Every NP-complete problem is reducible to SAT in polynomial time.

(iv) SAT is reducible to every NP-complete problem in polynomial time.

(v) Every NP-complete problem is solvable in polynomial time by a deterministic Turing machine.

(5)

Answer: *Correct: i, iii, iv*

Incorrect:

ii: If the Halting Problem were reducible to SAT, it would be decidable because SAT is. But we know that the Halting Problem is not decidable.

v: We don't know if this is true or not: this statement is equivalent to P=NP, which is at the moment unsolved.

(e) Which of the following statements about the λ-calculus are true?

(i) The Church numeral $\overline{3}$ is $\lambda f.\lambda x.((f\,x)\,x)\,x$.

(ii) The pair of two terms $\langle a, b\rangle$ is represented by $\lambda x.x\,a\,b$.

(iii) Every Turing-computable function can be represented by a λ-term.

(iv) Some λ-terms do not have a normal form.

(v) The problem of determining whether a λ-term has a normal form is decidable.

(5)

Answer: *Correct: ii, iii, iv*

Incorrect:

i: This term applies f to three copies of x, while it should instead apply f three times starting from x: $\overline{3} = \lambda f.\lambda x.f\,(f\,(f\,x))$.

v: Determining whether a λ-term is normalizable is equivalent to the Halting Problem, which is known to be undecidable.

Question 2
Consider the following Context-Free Grammar (CFG):

$$\begin{aligned} S &\rightarrow AS \mid AB \mid AA \\ A &\rightarrow aA \mid \epsilon \\ B &\rightarrow BCDb \mid e \\ C &\rightarrow Dc \mid c \\ D &\rightarrow Cd \mid d \end{aligned}$$

S, A, B, C, and D are nonterminals, a, b, c, d, and e are terminals, and S is the start symbol. The set of nullable nonterminals is $N_\epsilon = \{S, A\}$.

(a) Systematically compute the *first sets* for all nonterminals, i.e., $\mathrm{first}(S)$, $\mathrm{first}(A)$, $\mathrm{first}(B)$, $\mathrm{first}(C)$, and $\mathrm{first}(D)$, by setting up and solving the equations according to the definitions of first sets for nonterminals and strings of grammar symbols, looking for the *smallest* solutions. Recall that an equation of the form $X = X \cup Y$, in the absence of other constraints on X, simplifies to $X = Y$ when we are looking for the smallest solution. Show your calculations.

To get you started, the equation for $\mathrm{first}(A)$, before simplification, resulting from the productions for A ($A \rightarrow aA \mid \epsilon$), is:

$$\mathrm{first}(A) = \mathrm{first}(aA) \cup \mathrm{first}(\epsilon) \tag{8}$$

Answer: *Keeping in mind which non-terminals are nullable, we obtain the following equations:*

$$\begin{aligned} \mathrm{first}(A) &= \mathrm{first}(aA) \cup \mathrm{first}(\epsilon) \\ &= \{a\} \cup \emptyset \\ &= \{a\} \end{aligned}$$

$$\begin{aligned} \mathrm{first}(B) &= \mathrm{first}(BCDb) \cup \mathrm{first}(e) \\ &= (\mathrm{first}(B) \cup \emptyset) \cup \{e\} \\ &= \mathrm{first}(B) \cup \{e\} \end{aligned}$$

$$\begin{aligned} \mathrm{first}(C) &= \mathrm{first}(Dc) \cup \mathrm{first}(c) \\ &= (\mathrm{first}(D) \cup \emptyset) \cup \{c\} \\ &= \mathrm{first}(D) \cup \{c\} \end{aligned}$$

$$\begin{aligned} \mathrm{first}(D) &= \mathrm{first}(Cd) \cup \mathrm{first}(d) \\ &= (\mathrm{first}(C) \cup \emptyset) \cup \{d\} \\ &= \mathrm{first}(C) \cup \{d\} \end{aligned}$$

The solution of the equation for $\mathrm{first}(A)$ *is manifest. As to the equation for* $\mathrm{first}(B)$*, we need only observe that it has the form* $X = X \cup Y$ *and that there are no other constraints on* $\mathrm{first}(B)$*. The smallest solution is thus given by* $\mathrm{first}(B) = \{e\}$*. We can solve the equations for* $\mathrm{first}(\mathrm{C})$ *and* $\mathrm{first}(D)$ *by substituting the RHS of the equation for* $\mathrm{first}(D)$ *into the equation for* $\mathrm{first}(C)$*, yielding the equation* $\mathrm{first}(C) = (\mathrm{first}(C) \cup \{d\}) \cup \{c\} = \mathrm{first}(C) \cup \{c, d\}$*. This is again an equation of the form* $X = X \cup Y$*. As there are no other constraints on* $\mathrm{first}(C)$*, he smallest solution is just* $\mathrm{first}(C) = \{c, d\}$*. And now we can use that to solve the equation for* $\mathrm{first}(D)$ *yielding* $\mathrm{first}(D) = \{c, d\} \cup \{d\} = \{c, d\}$*.*

Now we can turn to setting up and solving the equation for $\mathrm{first}(S)$*, again keeping in mind which non-termainals are nullable:*

$$\begin{aligned}
\mathrm{first}(S) &= \mathrm{first}(AS) \cup \mathrm{first}(AB) \cup \mathrm{first}(AA) \\
&= (\mathrm{first}(A) \cup \mathrm{first}(S)) \cup (\mathrm{first}(A) \cup \mathrm{first}(B)) \cup (\mathrm{first}(A) \cup \mathrm{first}(A)) \\
&= \mathrm{first}(S) \cup \mathrm{first}(A) \cup \mathrm{first}(B) \\
&= \mathrm{first}(S) \cup \{a\} \cup \{e\} \\
&= \mathrm{first}(S) \cup \{a, e\}
\end{aligned}$$

Again, an equatiom of the form $X = X \cup Y$*, with no further constraints on* $\mathrm{first}(S)$*, meaning that the smallest solution is simply* $\mathrm{first}(S) = \{a, e\}$*.*

(b) Set up the subset constraint system that defines the *follow sets* for all non-terminals; i.e., $\mathrm{follow}(S)$, $\mathrm{follow}(A)$, $\mathrm{follow}(B)$, $\mathrm{follow}(C)$, and $\mathrm{follow}(D)$. Simplify where possible using the law

$$X \subseteq Z \ \wedge \ Y \subseteq Z \quad \Longleftrightarrow \quad X \cup Y \subseteq Z$$

and by removing trivially satisfied constraints such as $\emptyset \subseteq X$ and $X \subseteq X$.

To get you started, the constraints on $\mathrm{follow}(S)$, before simplification, resulting from S being the start symbol and the one production where S occurs in the RHS ($S \to AS$), are:

$$\begin{aligned}
\{\$\} &\subseteq \mathrm{follow}(S) \\
\mathrm{first}(\epsilon) &\subseteq \mathrm{follow}(S) \\
\mathrm{follow}(S) &\subseteq \mathrm{follow}(S)
\end{aligned}$$

(12)

Answer: *Note: Model answer very detailed for clarity. Not all details are necessary for full marks, but systematic simplification with justification of key steps is expected.*

Constraints for $\mathrm{follow}(S)$*. Note that* S *only appear in one RHS, of the production* $S \to AS$*, where it appears last; i.e. the string following* S *is*

just ϵ*, and by definition we have* $\mathrm{nullable}(\epsilon)$*. The constraints for* S *are thus:*

$$\begin{aligned} \{\$\} &\subseteq \mathrm{follow}(S) \\ \mathrm{first}(\epsilon) &\subseteq \mathrm{follow}(S) \\ \mathrm{follow}(S) &\subseteq \mathrm{follow}(S) \end{aligned}$$

Constraints for $\mathrm{follow}(A)$ *follow from the productions where* A *occurs in the RHS, i.e.*

$$\begin{aligned} S &\to AS \\ S &\to AB \\ S &\to AA \\ A &\to aA \end{aligned}$$

(note: A *occurs twice in the RHS of* $S \to AA$*, and* $\mathrm{nullable}(S)$, $\mathrm{nullable}(A)$, *and* $\mathrm{nullable}(\epsilon)$*):*

$$\begin{aligned} \mathrm{first}(S) &\subseteq \mathrm{follow}(A) \\ \mathrm{follow}(S) &\subseteq \mathrm{follow}(A) \\ \mathrm{first}(B) &\subseteq \mathrm{follow}(A) \\ \mathrm{first}(A) &\subseteq \mathrm{follow}(A) \\ \mathrm{follow}(S) &\subseteq \mathrm{follow}(A) \\ \mathrm{first}(\epsilon) &\subseteq \mathrm{follow}(A) \\ \mathrm{follow}(A) &\subseteq \mathrm{follow}(A) \\ \mathrm{first}(\epsilon) &\subseteq \mathrm{follow}(A) \\ \mathrm{follow}(A) &\subseteq \mathrm{follow}(A) \end{aligned}$$

Constraints for $\mathrm{follow}(B)$ *follow from the productions where* B *occurs in the RHS, i.e.*

$$\begin{aligned} S &\to AB \\ B &\to BCDb \end{aligned}$$

(note: $\mathrm{nullable}(\epsilon)$, $\neg\mathrm{nullable}(CDb)$*):*

$$\begin{aligned} \mathrm{first}(\epsilon) &\subseteq \mathrm{follow}(B) \\ \mathrm{follow}(S) &\subseteq \mathrm{follow}(B) \\ \mathrm{first}(CDb) &\subseteq \mathrm{follow}(B) \end{aligned}$$

Constraints for $\mathrm{follow}(C)$ *follow from the productions where* C *occurs in the RHS, i.e.*

$$\begin{aligned} B &\to BCDb \\ D &\to Cd \end{aligned}$$

(note: $\neg\text{nullable}(Db)$ *and* $\neg\text{nullable}(d)$*):*

$$\begin{aligned} \text{first}(Db) &\subseteq \text{follow}(C) \\ \text{first}(d) &\subseteq \text{follow}(C) \end{aligned}$$

Constraints for $\text{follow}(D)$ *follow from the productions where* D *occurs in the RHS, i.e.*

$$\begin{aligned} B &\to BCDb \\ C &\to Dc \end{aligned}$$

(note: $\neg\text{nullable}(b)$ *and* $\text{nullable}(\epsilon)$*):*

$$\begin{aligned} \text{first}(b) &\subseteq \text{follow}(D) \\ \text{first}(c) &\subseteq \text{follow}(D) \end{aligned}$$

Using

$$\begin{aligned} \text{first}(S) &= \{a, e\} \\ \text{first}(B) &= \{e\} \\ \text{first}(A) &= \{a\} \\ \text{first}(\epsilon) &= \emptyset \\ \text{first}(CDb) &= \text{first}(C) = \{c, d\} \\ \text{first}(Db) &= \text{first}(D) = \{c, d\} \\ \text{first}(d) &= \{d\} \\ \text{first}(b) &= \{b\} \\ \text{first}(c) &= \{c\} \end{aligned}$$

and eliminating trivial constraints (of the types $\emptyset \subseteq X$ *and* $X \subseteq X$*)*

yields:

$$\begin{aligned} \{\$\} &\subseteq \text{follow}(S) \\ \\ \{a, e\} &\subseteq \text{follow}(A) \\ \text{follow}(S) &\subseteq \text{follow}(A) \\ \{e\} &\subseteq \text{follow}(A) \\ \{a\} &\subseteq \text{follow}(A) \\ \\ \text{follow}(S) &\subseteq \text{follow}(B) \\ \{c, d\} &\subseteq \text{follow}(B) \\ \\ \{c, d\} &\subseteq \text{follow}(C) \\ \{d\} &\subseteq \text{follow}(C) \\ \{b\} &\subseteq \text{follow}(D) \\ \{c\} &\subseteq \text{follow}(D) \end{aligned}$$

This is equivalent to:

$$\begin{aligned} \{\$\} &\subseteq \text{follow}(S) \\ \{a, e\} \cup \text{follow}(S) \cup \{e\} \cup \{a\} &\subseteq \text{follow}(A) \\ \text{follow}(S) \cup \{c, d\} &\subseteq \text{follow}(B) \\ \{c, d\} \cup \{d\} &\subseteq \text{follow}(C) \\ \{b\} \cup \{c\} &\subseteq \text{follow}(D) \end{aligned}$$

which can be further simplified to the final constraints:

$$\begin{aligned} \{\$\} &\subseteq \text{follow}(S) \\ \{a, e\} \cup \text{follow}(S) &\subseteq \text{follow}(A) \\ \{c, d\} \cup \text{follow}(S) &\subseteq \text{follow}(B) \\ \{c, d\} &\subseteq \text{follow}(C) \\ \{b, c\} &\subseteq \text{follow}(D) \end{aligned}$$

(c) Solve the subset constraint system for the follow sets from the previous question by finding the *smallest* sets satisfying the constraints. (5)

Answer: *The smallest set satisfying the constraint for* $\text{follow}(S)$ *is obviously just* $\{\$\}$*. Substituting this into the remaining constraints makes the*

smallest sets satisfying those obvious too. Thus:

$$\begin{aligned}
\text{follow}(S) &= \{\$\} \\
\text{follow}(A) &= \{a, e\} \cup \{\$\} = \{a, e, \$\} \\
\text{follow}(B) &= \{c, d\} \cup \{\$\} = \{c, d, \$\} \\
\text{follow}(C) &= \{c, d\} \\
\text{follow}(D) &= \{b, c\}
\end{aligned}$$

Question 3

(a) Write the λ-terms that represent the following values:

- The Church numerals $\overline{0}$ and $\overline{2}$;
- The exponential function exp such that $(\text{exp}\ \overline{n}\ \overline{m}) = \overline{n^m}$;
- The Boolean values true and false;
- The *'not and'* Boolean operator nand, such that

$$\begin{array}{ll} \text{nand true true} \rightsquigarrow^* \text{false}, & \text{nand false true} \rightsquigarrow^* \text{true}, \\ \text{nand true false} \rightsquigarrow^* \text{true}, & \text{nand false false} \rightsquigarrow^* \text{true}. \end{array} \tag{8}$$

Answer:

$$\begin{array}{l} \overline{0} = \lambda f.\lambda x.x \qquad \overline{2} = \lambda f.\lambda x.f\,(f\,x) \\ \text{exp} = \lambda n.\lambda m.m\,n \\ \text{true} = \lambda x.\lambda y.x \qquad \text{false} = \lambda x.\lambda y.y \\ \text{nand} = \lambda\, u.\lambda v.u\,(v\,\text{false}\,\text{true})\,\text{true} \end{array}$$

(b) For every pairs of natural numbers $n > 0$ and i such that $1 \leq i \leq n$, consider the following λ-term:

$$\pi_i^n = \lambda x_1.\lambda x_2.\ldots.\lambda x_n.x_i$$

where $x_1, \ldots, x_n$ are distinct variables. For example

$$\pi_1^1 = \lambda x_1.x_1, \qquad \pi_1^3 = \lambda x_1.\lambda x_2.\lambda x_3.x_1, \qquad \pi_3^4 = \lambda x_1.\lambda x_2.\lambda x_3.\lambda x_4.x_3.$$

If a, b and c are any terms, what values do the following terms reduce to?

$$\begin{array}{ll} \pi_1^2\,a\,b \rightsquigarrow^* ? & \pi_2^3\,a\,b\,c \rightsquigarrow^* ? \\ \pi_1^3\,a\,b\,c \rightsquigarrow^* ? & \pi_3^3\,a\,b\,c \rightsquigarrow^* ? \end{array}$$

Now consider the following two functions shift and slide:

$$\text{shift} = \lambda f.\text{true}\,f, \qquad \text{slide} = \lambda f.\lambda u.\lambda v.f\,u.$$

What values do the following terms reduce to (show the steps in the reductions)?

$$\text{shift}\,\pi_1^2 \rightsquigarrow^* ? \qquad \text{slide}\,\pi_1^2 \rightsquigarrow^* ?$$

In general, when we apply shift and slide to a term of the form π_i^m, we obtain a term in the same form. Determine what the indexes of the result are:

$$\text{shift}\,\pi_i^m \rightsquigarrow^* \pi_?^? \qquad \text{slide}\,\pi_i^m \rightsquigarrow^* \pi_?^?$$

[Hint: Remember that the names of the variables are arbitrary; you can rename them freely as long as you keep them distinct.] (10)

Answer:

$$\pi_1^2\, a\, b \rightsquigarrow^* a \qquad \pi_2^3\, a\, b\, c \rightsquigarrow^* b$$
$$\pi_1^3\, a\, b\, c \rightsquigarrow^* a \qquad \pi_3^3\, a\, b\, c \rightsquigarrow^* c$$

$$\mathsf{shift}\,\pi_1^2 \rightsquigarrow^* \lambda y.\lambda x_1.\lambda x_2.x_1 = \pi_2^3$$
$$\mathsf{slide}\,\pi_1^2 \rightsquigarrow^* \lambda u.\lambda v.\lambda x_2.u = \pi_1^3$$

$$\mathsf{shift}\,\pi_i^m \rightsquigarrow^* \pi_{i+1}^{m+1} \qquad \mathsf{slide}\,\pi_i^m \rightsquigarrow^* \begin{cases} \pi_i^{m+1} & \text{if } i = 1 \\ \pi_{i+1}^{m+1} & \text{if } i > 1 \end{cases}$$

(c) In the context of complexity theory, give a definition of the class of NP-complete problems.

Name two examples of NP-complete problems, with brief informal definitions.

Explain what the open question P=NP means and why it is important in computer science. (7)

Answer:

- *A problem is NP-complete if it belongs to NP (it is decidable in non-deterministic polynomial time) and every NP problem can be reduced in polynomial time to it.*
- *Some well known NP-complete problems are:*
  - **The Satisfiability Problem:** *Determine if there is an assignment of truth values to the Boolean variables of a Boolean formula that makes it true.*
  - **The Travelling Salesman Problem:** *Determine if there is a route through all the vertices of a weighted graph with total weight less than a given number.*
  - **The Subset Sum Problem:** *Determine if there is a non-empty subset of a set of numbers that has sum 0.*
  - **The Graph Colouring Problem:** *Determine if a certain number of colours is sufficient to colour the vertices of a graph so that no adjacent vertices have the same colour.*
- *The question P=NP means determining if the classes of polynomially solvable and non-deterministically polynomially solvable problems are the same. In other words, if a problem can be solved in polynomial time by a non-deterministic Turing machine, is there always also a deterministic Turing machine that solves the problem in polynomial time?*

A positive solution to the question would mean that a large class of important problems for which at present we know only algorithms that run in exponential time could be solved in polynomial time.