University of Toronto
Scarborough Campus
December 19, 2006

CSC B36 Final Examination

Aids allowed: One 8.5×11 handwritten, non-photocopied 'cheat sheet'

Duration: Three hours

- There should be 11 pages in this exam booklet, including this cover page.
- Answer all questions.
- Put all answers in this booklet, in the spaces provided.
- For rough work, use the backs of the pages; ***these will not be marked***.
- Good luck!

Family Name ______________________________ Given Name ________________________

Student Number ____

Problem	Marks Rec'ved	Marks Worth
1.		20
2.		5
3.		10
4.		10
5.		20
6.		10
7.		10
8.		15
9.		20
TOTAL		120

QUESTION 1. (20 marks)

The ***integer logarithm*** (base 2) of a positive integer n, is the largest power of 2 that does not exceed n; i.e., the natural number ℓ such that $2^\ell \leq n < 2^{\ell+1}$. Prove that the following program returns the integer logarithm of n. More precisely, prove that the program is correct with respect to the following pre/postcondition specification.

Precondition : n is a positive integer.

Postcondition : IntLog(n) returns the natural number ℓ where $2^\ell \leq n < 2^{\ell+1}$.

```
IntLog(n)
k := 0
x := 1
while 2 * x <= n do
    k := k + 1
    x := 2 * x
end while
return k
```

Proof of correctness:

Partial correctness:

Loop Invariant Lemma: *For all $i \in \mathbb{N}$, if the loop is executed at least i times then (a) $x_i = 2^{k_i}$ and (b) $x_i \leq n$.*

PROOF THAT LIL ⇒ PARTIAL CORRECTNESS: *Suppose that the program terminates. Thus the loop terminates after some number, say t, of iterations. By parts (a) and (b) of the loop invariant, $2^{k_t} \leq n$; by the loop exit condition $n < 2x_t$, and so by part (a) of the loop invariant $n < 2 \cdot 2^{k_t} = 2^{k_t+1}$. Thus, $2^{k_t} \leq n < 2^{k_t+1}$. Since the program returns k_t, the postcondition is satisfied.*

PROOF OF THE LIL: *Let $P(i)$ be the predicate: "if the loop is executed at least i times then (a) $x_i = 2^{k_i}$ and (b) $x_i \leq n$". We will use induction to prove that $P(i)$ holds for each $i \in \mathbb{N}$.*

BASIS: *$i = 0$. By the program $k_0 = 0$ and $x_0 = 1$, so $x_0 = 2^{k_0}$. By the precondition, n is a positive integer, so $x_0 \leq n$. So, $P(0)$ holds.*

INDUCTION STEP: *Let j be an arbitrary natural number and suppose that $P(j)$ holds. We must prove that $P(j+1)$ also holds. Suppose that the loop is executed at least $j+1$ times (otherwise, $P(j+1)$ trivially holds). We have*

$$\begin{aligned} x_{j+1} &= 2x_j && \text{[by the program]} \\ &= 2 \cdot 2^{k_j} && \text{[by the induction hypothesis]} \\ &= 2^{k_j+1} \\ &= 2^{k_{j+1}} && \text{[by the program, since } k_{j+1} = k_j + 1\text{]} \end{aligned}$$

So, part (a) of $P(j+1)$ holds. Furthermore, since the loop is executed at least $j+1$ times, by the loop exit condition, $2x_j \leq n$. By the program, $x_{j+1} = 2x_j$, so $x_{j+1} \leq n$. So, part (b) of $P(j+1)$ also holds.

Termination: *We claim that the value of $n - x_i$ is a non-negative integer that decreases in each iteration. The Well-Ordering Principle then implies that the loop (and hence the program) terminates.*

That $n - x_i$ is a non-negative integer follows immediately from the LIL. To see that its value decreases in each iteration, observe that, again by the LIL, for each $i \in \mathbb{N}$, if the loop is executed at least $i+1$ times then $x_{i+1} > x_i$, and so $n - x_{i+1} < n - x_i$.

[more space, if needed, is available on the next page]

[this page was deliberately left blank]

[more space, if needed, is available on the next page]

[this page was deliberately left blank]

QUESTION 2. (5 marks)

How many Boolean functions of four variables can be represented by propositional formulas that use only the connectives $\neg$ and $\wedge$? Justify your answer.

Number of functions: *65,536.*

Justification: *The set $\{\neg, \wedge\}$ is complete, and so we can represent **all** Boolean functions (and, in particular, all Boolean functions of four variables) by propositional formulas that use only connectives in this set. There are $2^{2^4} = 2^{16} = 65,536$ Boolean functions of four variables.*

QUESTION 3. (10 marks)

Give a CNF and a DNF formula, each of which is logically equivalent to $\neg(x \to y) \vee z$.

In the space following the CNF and DNF formulas, you may show the rough work (such as truth tables or formula manipulations) that you used to obtain your answers. This will be reviewed for possible partial marks, so it should be clear and relevant.

CNF formula: $(x \vee z) \wedge (\neg y \vee z)$.

DNF formula: $(x \wedge \neg y) \vee z$.

Related rough work:

$$
\begin{array}{llll}
\neg(x \to y) \vee z & \textsc{leqv} & \neg(\neg x \vee y) \vee z & \textit{[}{\to}\textit{-Law]} \\
 & \textsc{leqv} & (x \wedge \neg y) \vee z & \textit{[De Morgan and Double Negation]} \\
 & \textsc{leqv} & (x \vee z) \wedge (\neg y \vee z) & \textit{[Distributive Law]}
\end{array}
$$

The penultimate formula is in DNF and the last one is in CNF. CNF and DNF formulas can also be obtained by first writing down the truth table of the given formula, and then applying the procedures discussed in class. This more standard approach is, in this case, a bit more laborious and yields somewhat longer formulas, but is a perfectly correct way to approach the question.

QUESTION 4. (10 marks)

Transform the following first-order formula into a logically equivalent formula in Prenex Normal Form. Demonstrate the steps through which you obtain your answer.

$$\Big(\forall x\, \exists y\, \big(A(x) \wedge C(x,y)\big) \rightarrow \neg\forall y\, \big(A(x) \wedge B(y)\big)\Big)$$

Answer:

$$
\begin{array}{lll}
& \Big(\forall x\, \exists y\, \big(A(x) \wedge C(x,y)\big) \rightarrow \neg\forall y\, \big(A(x) \wedge B(y)\big)\Big) & \\
\text{LEQV} & \Big(\forall x\, \exists y\, \big(A(x) \wedge C(x,y)\big) \rightarrow \exists y\, \neg\big(A(x) \wedge B(y)\big)\Big) & \textit{[Duality]} \\
\text{LEQV} & \Big(\forall u\, \exists v\, \big(A(u) \wedge C(u,v)\big) \rightarrow \exists y\, \neg\big(A(x) \wedge B(y)\big)\Big) & \textit{[2 x Renaming]} \\
\text{LEQV} & \exists u\, \forall v\, \exists y\, \Big(\big(A(u) \wedge C(u,v)\big) \rightarrow \neg\big(A(x) \wedge B(y)\big)\Big) & \textit{[3 x Factoring]}
\end{array}
$$

QUESTION 5. (20 marks)

For each statement below, indicate whether it is true or false by circling the appropriate response, and justify your answer.

Make sure your justification (if any) is sensible: *A correct answer with no justification will receive partial credit, but a correct answer with bogus justification will not.*

a. $\forall x\, A(x) \land \forall x\, B(x)$ LEQV $\forall x\, \big(A(x) \land B(x)\big)$ **$\boxed{\textbf{True}}$ / False**

Justification: *Take any structure where $\forall x\, A(x) \land \forall x\, B(x)$ is satisfied. Then, no element of the structure's domain falsifies $A(x) \land B(x)$ (otherwise at least one of $\forall x\, A(x)$, $\forall x\, B(x)$ would be falsified, and so would their conjunction); therefore, $\forall x\, \big(A(x) \land B(x)\big)$ is also satisfied. Conversely, take any structure where $\forall x\, \big(A(x) \land B(x)\big)$ is satisfied. There is no element of the structure's domain that falsifies $A(x)$ and no element that falsifies $B(x)$. Thus, $\forall x\, A(x) \land \forall x\, B(x)$ is also satisfied.*

b. $\forall x\, A(x) \to \forall x\, B(x)$ LEQV $\forall x\, \big(A(x) \to B(x)\big)$ **True / $\boxed{\textbf{False}}$**

Justification: *Consider the structure with domain the natural numbers, where $A(x)$ means "x is a multiple of 2" and $B(x)$ means "x is a multiple of 4". In this structure, the formula $\forall x\, A(x) \to \forall x\, B(x)$ says that if every natural number is a multiple of 2, then every natural number is a multiple of 4; this is vacuously true because the antecedent is false: not all natural numbers are multiples of 2. On the other hand, the formula $\forall x\, \big(A(x) \to B(x)\big)$ says that every natural number that is a multiple of 2 is a multiple of 4; this is fase: 2 is a counterexample (a multiple of 2 that is not a multiple of 4). Since there is a structure that satisfies one formula and falsifies the other, the two formulas are not logically equivalent.*

c. $\forall x\, \exists y\, \big(A(x) \to B(y)\big)$ LEQV $\exists y\, \forall x\, \big(A(x) \to B(y)\big)$ **$\boxed{\textbf{True}}$ / False**

Justification: *We have:*

$$\big(\exists x\, A(x) \to \exists y\, B(y)\big) \text{ LEQV } \forall x\, \big(A(x) \to \exists y\, B(y)\big) \text{ LEQV } \forall x\, \exists y\, \big(A(x) \to B(y)\big) \quad \textit{and}$$
$$\big(\exists x\, A(x) \to \exists y\, B(y)\big) \text{ LEQV } \exists y\, \big(\forall x\, A(x) \to B(y)\big) \text{ LEQV } \exists y\, \forall x\, \big(A(x) \to B(y)\big)$$

By factoring the $\exists x$ and $\exists y$ quantifiers from $\big(\exists x\, A(x) \to \exists y\, B(y)\big)$ in different orders, we have shown that $\forall x\, \exists y\, \big(A(x) \to B(y)\big)$ and $\exists y\, \forall x\, \big(A(x) \to B(y)\big)$ are both logically equivalent to $\big(\exists x\, A(x) \to \exists y\, B(y)\big)$. Thus, they are logically equivalent to each other.

d. $\forall x\, \exists y\, \big(A(x) \to C(x, y)\big)$ LEQV $\exists y\, \forall x\, \big(A(x) \to C(x, y)\big)$ **True / $\boxed{\textbf{False}}$**

Justification: *Consider the structure with domain the natural numbers, where $A(x)$ means "$x \neq 0$" and $C(x, y)$ means "y is the predecessor of x" (i.e., $y = x - 1$). In this structure, the formula $\forall x\, \exists y\, \big(A(x) \to C(x, y)\big)$ says that every non-zero natural number has a predecessor, which is true; while the formula $\exists y\, \forall x\, \big(A(x) \to C(x, y)\big)$ says that there is a natural number that is the predecessor of every non-zero natural number, which is false. Since there is a structure that satisfies one formula and falsifies the other, the two formulas are not logically equivalent.*

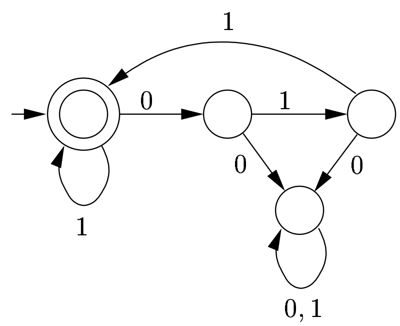
This is very similar to the "everyone loves somebody" versus "somebody is loved by everyone" example we discussed in class.

QUESTION 6. (10 marks)

Let $L = \{x \in \{0,1\}^* : \text{every 0 in } x \text{ is immediately followed by at least two consecutive 1s}\}$.

a. (5 marks) Give the diagram of a DFSA that accepts L. You need not prove that your automaton is correct.

DFSA diagram:

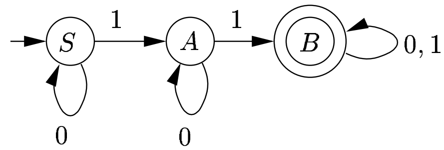


b. (5 marks) Give a regular expression R that denotes L. You need not prove that your regular expression is correct.

Regular Expression: $(1 + 011)^*$ *is one of several (equivalent) possibilities.*

QUESTION 7. (10 marks)

Consider the DFSA whose diagram is shown below.



a. (5 marks) Give a regular expression that denotes the language accepted by this DFSA. You need not prove that your regular expression is correct.

Regular expression: $0^*10^*1(0+1)^*$.

b. (5 marks) List the productions of a context-free grammar that generates the language accepted by this DFSA. You need not prove that your grammar is correct.

Context-free grammar:

$$
\begin{aligned}
S &\to Z1Z1A \\
Z &\to 0Z \\
Z &\to \epsilon \\
A &\to 0A \\
A &\to 1A \\
A &\to \epsilon
\end{aligned}
$$

Other correct answers are, of course, possible.

QUESTION 8. (15 marks)

Consider the language $L = \{0^m 1^m 2^n : \ m, n \in \mathbb{N}\}$ — i.e., the set of strings consisting of some number (perhaps zero) of 0s, followed by ***the same*** number of 1s, followed by ***any*** number (perhaps zero) of 2s.

a. (5 marks) List the productions of a context-free grammar that generates L.

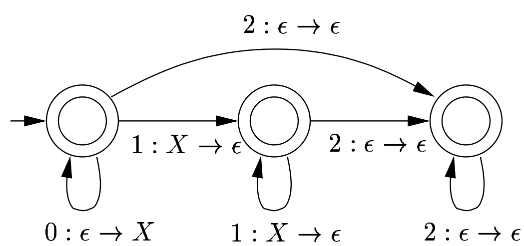
CFG:

$$
\begin{aligned}
S &\rightarrow AB \\
A &\rightarrow 0A1 \\
A &\rightarrow \epsilon \\
B &\rightarrow 2B \\
B &\rightarrow \epsilon
\end{aligned}
$$

Other correct answers are, of course, possible.

b. (5 marks) Show the diagram of a pushdown automaton that accepts L.

PDA:



c. (5 marks) Prove that there is no right-linear context-free grammar that generates L.

Proof: *Since right-linear CFGs generate only regular languages, it suffices to prove that L is not regular.*

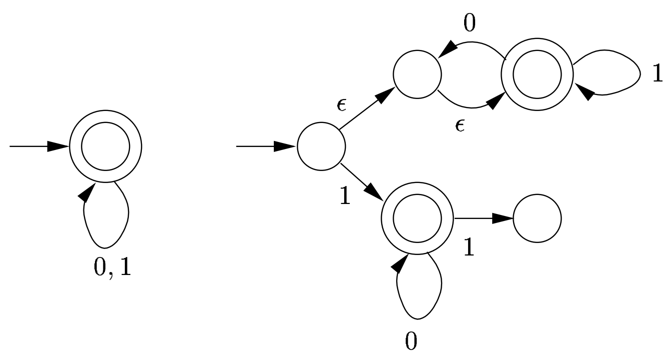
Suppose, for contradition, L is regular. Then (because the intersection of regular languages is regular), so is $L \cap \mathcal{L}(0^*1^*)$. But $L \cap \mathcal{L}(0^*1^*) = \{0^m 1^m : \ m \in \mathbb{N}\}$, which we have proved is not regular. Thus, L is not reguar.

QUESTION 9. (20 marks)

Indicate whether each of the following statements is true or false, by circling the appropriate response. Do not justify your answers.

Do not guess: *4 for each correct answer, 0 for no answer, −2 for wrong answer.*

(a) The two FSA below accept the same language. $\boxed{\textbf{True}}$ / **False**



(b) $(0^*1^*)^*(00)^*(0+\epsilon) \equiv (0+1)^*$. $\boxed{\textbf{True}}$ / **False**

(c) For all languages L and L', if L and L' are both non-regular, then $L \cup L'$ is also non-regular. **True** / $\boxed{\textbf{False}}$

(d) For every language L, if L is regular and $x \in L$ then $L - \{x\}$ is also regular. $\boxed{\textbf{True}}$ / **False**

(e) For every language L, if L is generated by a right-linear grammar with 10 variables, then L is accepted by a DFSA with at most 1024 states. $\boxed{\textbf{True}}$ / **False**

THE END