

THE AUSTRALIAN NATIONAL UNIVERSITY

Second Semester 2018

**COMP1600/COMP6260
(Foundations of Computation)**

Writing Period: 3 hours duration

Study Period: 15 minutes duration

Permitted Materials: One A4 page with hand-written notes on both sides

Answer ALL questions

Total marks: 100

The questions are followed by labelled blank spaces into which your answers are to be written. Additional answer panels are provided (at the end of the paper) should you wish to use more space for an answer than the space provided. If you require more space on an answer panel, be sure to indicate this by drawing a vertical line down the right-hand side of the panel.

Assignment Project Exam Help
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

The following spaces are for use by the examiners.

Q1 (Logic)	Q2 (ND)	Q3 (SI)	Q4 (HL)	Q5 (FSA)	Q6 (CFL)
------------	---------	---------	---------	----------	----------

Q7 (TM)	Total
---------	-------

QUESTION 1 [14 marks]

Logic

Recall that two formulae are *equivalent* if they have the same truth values for all variable assignments, and consider the following set of formulae:

- $(p \wedge q) \vee (\neg p \wedge \neg q)$
- $\neg(p \vee q)$
- $(q \wedge p) \vee \neg p$
- $(p \vee \neg q) \wedge (q \vee \neg p)$

- (a) Identify two formulae in the above set of formulae that are equivalent, and demonstrate their equivalence by means of a truth table.

QUESTION 1(a)

[4 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

- (b) Identify two formulae in the above set of formulae that are not equivalent, and demonstrate the fact that they are not equivalent by a variable assignment.

QUESTION 1(b)

[4 marks]

(c) State whether the following formulae are true or false where x , y and z range over the integers, and justify your answer briefly.

- (1). $\forall x \exists y (2x - y) = 0$
- (2). $\exists x \forall y (2x - y) = 0$
- (3). $\forall x \exists y (x - 2y) = 0$

QUESTION 1(c)

[6 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

QUESTION 2 [16 marks]**Natural Deduction**

The following questions ask for proofs using natural deduction. Present your proofs in the Fitch style as used in lectures. You may only use the introduction and elimination rules given in Appendix 1. Number each line and include justifications for each step in your proofs.

- (a) Give a natural deduction proof of the formula $p \vee (p \rightarrow q)$. In this proof, you may use the law of excluded middle (*LEM*) $p \vee \neg p$ in addition to the rules provided in the appendix. That is, you may state $p \vee \neg p$ at any line in the proof, where p can stand for an arbitrary formula, and justify this by (LEM).

QUESTION 2(a)

[8 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

(b) Give a natural deduction proof of the rule

$$\frac{\exists x P(x) \wedge \forall x \forall y (R(x, y) \rightarrow P(y))}{\exists x \forall y (R(x, y) \rightarrow P(y))}$$

using only the rules in the appendix.

QUESTION 2(b)

[8 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

QUESTION 3 [16 marks]

Structural Induction

(a) Consider the function `subl` that computes the list of sub-lists of a given list:

```
subl :: [a] -> [[a]]
subl [] = [[]]                                     -- S1
subl (x:xs) = (subl xs) ++ map (pref x) (subl xs) -- S2
```

where the functions `map`, `pref` and `++` are given by:

```
map :: (a -> b) -> [a] -> [b]
map f [] = []                                     -- M1
map f (x:xs) = (f x):(map f xs)                  -- M2

pref :: a -> [a] -> [a]
pref x l = x:l                                    -- P
```

```
(++) :: [a] -> [a] -> [a]
[]      ++ ys      = ys                          -- A1
(x:xs) ++ ys      = x : (xs ++ ys)           -- A2
```

Assignment Project Exam Help

Show, using structural induction, that

`length` (<https://eduassistpro.github.io/>)

for all lists `l` of type `a`.

Here, we assume the standard definition of the `length` fun

Add WeChat `edu_assist_pro`

```
length [] = 0                                     -- L1
length (x:xs) = 1 + length xs                   -- L2
```

and you may use the fact that `map` preserves `length`, and the fact that `length` is compatible with concatenation, that is the equations

```
length (map f xs) = length xs                  -- LM
length (xs ++ ys) = length xs + length ys    -- LA
```

in your proof, with justification as indicated.

In all proofs indicate the justification (eg, the line of a definition used) for each step.

- (i) State and prove the base case.

QUESTION 3(a)(i)

[2 marks]

- (ii) State the inductive hypothesis.

QUESTION 3(a)(ii)

[1 mark]

- (iii) State and prove the step case goal.

QUESTION 3(a)(iii)

[5 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- (b)** Give an inductive proof the fact that left folding is compatible with list concatenation.
Consider the following definition of left folding:

```
foldl :: :: (b -> a -> b) -> b -> [a] -> b
foldl f z [] = z                                     -- F1
foldl f z (x:xs) = foldl f (f z x) xs           -- F2
```

and consider a fixed function f (of type $b \rightarrow a \rightarrow b$) and a fixed list ys (of elements of type b) and show that

$$P(xs) \equiv \forall z (foldl f z (xs ++ ys) = foldl f (foldl f z xs) ys)$$

holds for all lists xs (of elements of type a).

- (i) State and prove the base case goal

QUESTION 3(b)(i)

[2 marks]

Assignment Project Exam Help

- (ii) State the induction hypothesis

QUESTIONS

[1 mark]

Add WeChat edu_assist_pro

(iii) State and prove the step case goal.

QUESTION 3(b)(iii)

[5 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

QUESTION 4 [16 marks]**Hoare Logic**

- (a) Specify a precondition P and a postcondition Q such that the Hoare-Triple $\{P\} S \{Q\}$ holds precisely for all programs S that *never* terminate.

QUESTION 4(a)

[2 marks]

- (b) The following piece of code is called *Rem*

```
r := x;  
q := 0;  
while (r >= n)  
    r := r - n;  
    q := q + 1;
```

Assignment Project Exam Help

and computes two numbers, q and r , where

- q is the int
- r is the r

<https://eduassistpro.github.io/>

We wish to use H

$\{True\} Rem \{x = n * q + r\}$

In the questions below (and your answers), we may refer to the loop code as *Loop*, the body of the loop (i.e. $r := r - n$; $q := q + 1$) as *Body*, and the initialisation assignments (i.e. $r := x$; $q := 0$) as *Init*.

- (i) Given the desired postcondition $\{x = n * q + r\}$, what is a suitable invariant I for *Loop*?

QUESTION 4(b)(i)

[3 marks]

- (ii) Prove that your answer to the previous question is indeed a loop invariant. That is, if we call your invariant I , show that $\{I\} \text{ Body } \{I\}$. Be sure to properly justify each step of your proof.

QUESTION 4(b)(ii)

[3 marks]

Assignment Project Exam Help

- (iii) Using the previous result and some more proof steps show that

{ <https://eduassistpro.github.io/>

Be sure to p

QUESTION 4(b)(iii)

[2 marks]

Add WeChat edu_assist_pro

- (iv) Explain why the corresponding Hoare-triple for total correctness, that is

$[True] Rem[x = n * q + r]$

is *not* valid by giving a counter-example that shows that the triple above does not hold in general.

QUESTION 4(b)(iv)

[2 marks]

- (v) Identify a precondition P such that the Hoare triple

$[P] Rem[x = n * q + r]$

is valid. Explain why the Hoare-triple now holds (no formal proof in Hoare Logic required).

QUESTION 4(b)(v)

[4 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

QUESTION 5 [13 marks]**Finite State Automata**

- (a) Design a Finite State Automaton that recognises the language of all strings over the alphabet $\Sigma = \{a, b, c\}$ where both the string 'abc' *and* the string 'cba' occurs as a substring.

Here, a string s is a substring of a string w if w can be written as w_1sw_2 .

QUESTION 5(a)

[3 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

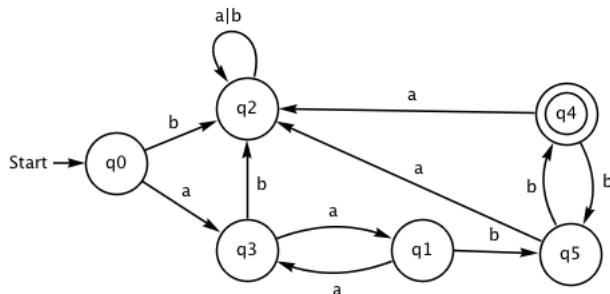
Add WeChat edu_assist_pro

- (b) Is your Finite State Automaton (above) deterministic or non-deterministic? Explain.

QUESTION 5(b)

[1 mark]

(c) What language is recognised by the following Finite State Automaton?



Describe the language in English, and give a regular expression defining the language.

QUESTION 5(c)

[3 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

(d) Consider the statement

$\forall w \in \Sigma^*. N^*(q_0, w) = q_5$

Express this property in English. Why might it be relevant?

QUESTION 5(d)

[2 marks]

(e) For the Finite State Automaton above, prove that

$$\forall n \in \mathbb{N} . N^*(q_1, (aa)^n) = q_1$$

and hence, or otherwise, conclude that

$$\forall n \in \mathbb{N} . N^*(q_0, (aa)^{n+1}) = q_1$$

QUESTION 5(e)

[4 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

QUESTION 6 [13 marks]**Context-Free Grammars**

- (a) Design a push-down automaton that recognises precisely the language

$$\{a^m b^n \mid n > m > 0\}$$

QUESTION 6(a)

[4 marks]

- (b) Is your automaton deterministic, or non-deterministic? Briefly justify your answer.

QUESTION 6

[1 mark]

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- (c) Demonstrate, using the pigeon hole principle or other bove
is *not* regular.

QUESTION 6(c)

[4 marks]

(d) Give a context-free grammar that generates precisely the language given above.

QUESTION 6(d)

[4 marks]

Assignment Project Exam Help

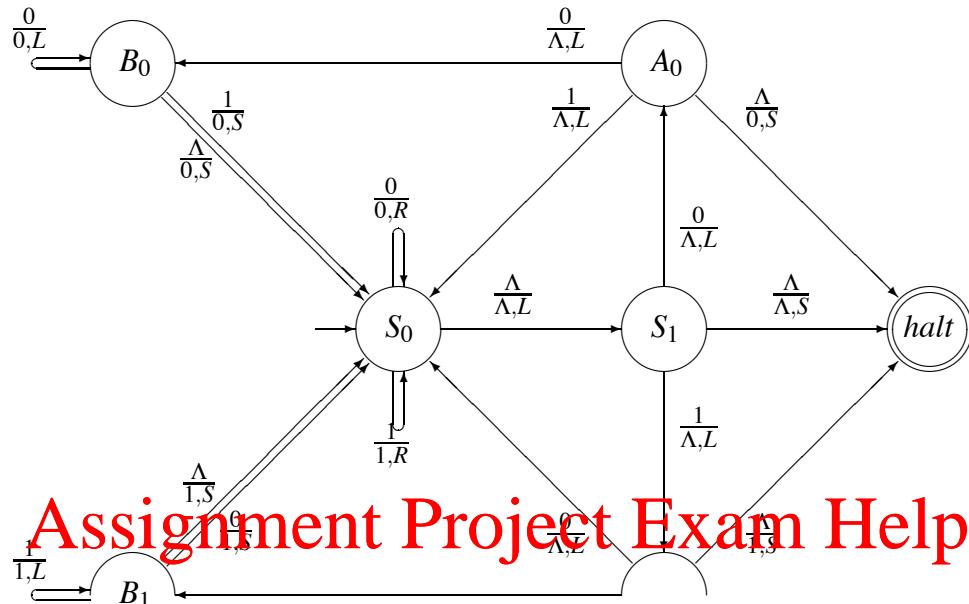
<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

QUESTION 7 [12 marks]

Turing Machines

- (a) The following diagram shows a Turing machine, whose purpose is either to accept or reject the input string. The input string is a string of 0's and 1's and the tape is blank to the left and to the right of the input string. Initially the head is somewhere on the input string.



- (i) For each of the strings 010101, 101100
mine the content of
the tape after the machine has terminated with the g
QUESTION 7(a)(i) Add WeChat edu_assist_pro [3 marks]

- (ii) Given an input string s , describe the output after the machine has terminated on input string s .

QUESTION 7(a)(ii)

[3 marks]

- (iii) Explain (no formal proof required) why the machine will always terminate, regardless of the given input string.

QUESTION 7(a)(iii) **Assignment Project Exam Help** [3 marks]

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- (b)** Design a Turing Machine that will take a pair of binary numbers, separated by a hash symbol (#), and reverses their order. That is, 0101#1111 should be replaced by 1111#0101 on the tape. Assume that the tape is empty apart from the input and that the tape head is somewhere over the input initially. Include a brief description of the purpose of the individual states.

QUESTION 7(b)

[3 marks]

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Additional answers. Clearly indicate the corresponding question and part.

Assignment Project Exam Help

Additional answers.

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Additional answers. Clearly indicate the corresponding question and part.

Assignment Project Exam Help

Additional answers.

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Additional answers: deliberately left like this for use in landscape mode. Clearly indicate the corresponding question and part.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Additional answers: deliberately left like this for use in landscape mode. Clearly indicate the corresponding question and part.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Appendix 1 — Natural Deduction Rules

Propositional Calculus

$$(\wedge I) \quad \frac{p \quad q}{p \wedge q} \qquad (\wedge E) \quad \frac{p \wedge q}{p} \quad \frac{p \wedge q}{q}$$

$$(\vee I) \quad \frac{p}{p \vee q} \quad \frac{p}{q \vee p} \qquad (\vee E) \quad \frac{p \vee q \quad r \quad r}{r}$$

$$(\rightarrow I) \quad \frac{q}{\frac{p \rightarrow q}{[p]}} \qquad (\rightarrow E) \quad \frac{p \quad p \rightarrow q}{[p]}$$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

$$(\neg I) \quad \frac{}{\neg p} \qquad (\text{PC}) \quad \frac{}{}$$

Add WeChat edu_assist_pro

$$(\neg E) \quad \frac{p \quad \neg p}{F} \qquad (\text{T}) \quad \frac{}{T}$$

Predicate Calculus

$$(\forall I) \quad \frac{P(a) \quad (a \text{ arbitrary})}{\forall x. P(x)} \qquad (\forall E) \quad \frac{\forall x. P(x)}{P(a)}$$

$$(\exists I) \quad \frac{P(a)}{\exists x. P(x)} \qquad (\exists E) \quad \frac{\exists x. P(x) \quad q \quad (a \text{ arbitrary})}{q \quad (a \text{ is not free in } q)}$$

Appendix 2 — Truth Table Values

p	q	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$\neg p$	$p \Leftrightarrow q$
T	T	T	T	T	F	T
T	F	T	F	F	F	F
F	T	T	F	T	T	F
F	F	F	F	T	T	T

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Appendix 3 — Hoare Logic Rules

- Precondition Strengthening:

$$\frac{P_s \rightarrow P_w \quad \{P_w\} S \{Q\}}{\{P_s\} S \{Q\}}$$

- Postcondition Weakening:

$$\frac{\{P\} S \{Q_s\} \quad Q_s \rightarrow Q_w}{\{P\} S \{Q_w\}}$$

- Assignment:

$$\{Q(e)\} x := e \{Q(x)\}$$

- Sequence:

$$\frac{\{P\} S_1 \{Q\} \quad \{Q\} S_2 \{R\}}{\{P\} S_1; S_2 \{R\}}$$

- **Assignment Project Exam Help**

$$\frac{\{P \wedge b\}}{\{ \text{https://eduassistpro.github.io/} \}}$$

- While Loop:

$$\frac{\{P \wedge b\} \vdash \{P\}}{\{P\} \text{ while } b \text{ do } S \{P \wedge \neg b\}}$$