UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR08012 INFORMATICS 1 - COMPUTATION AND LOGIC

Monday 15th August 2016

09:30 to 11:30

INSTRUCTIONS TO CANDIDATES

1. **Note that ALL QUESTIONS ARE COMPULSORY.**
2. **DIFFERENT QUESTIONS MAY HAVE DIFFERENT NUMBERS OF TOTAL MARKS. Take note of this in allocating time to questions.**
3. **CALCULATORS MAY NOT BE USED IN THIS EXAMINATION.**

Convener: D. K. Arvind
External Examiner: C. Johnson

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. (a) For each of the following propositional expressions, use a truth table to determine whether it is tautologous, contradictory, or contingent. [*2 marks*]

    i. $\neg((B \vee \neg A) \to B) \vee B$

    ii. $((A \vee \neg B) \to A) \to (B \to A)$

   (b) This part concerns the 256 possible truth valuations of the following eight propositional letters A, B, C, D, E, F, G, H. For each of the following expressions, say how many of the 256 valuations satisfy the expression, and briefly explain your reasoning. For example, the expression D is satisfied by half of the valuations, that is 128 of the 256, since for each valuation that makes D true there is a matching valuation that makes D false.

    i. $\neg\neg A$

    ii. $E \vee F$

    iii. $B \oplus \neg D$

    iv. $\neg A \oplus (B \wedge C)$

    v.
$$(A \to B) \wedge (B \to C) \wedge (C \to D) \wedge (D \to E)$$

    vi.
$$(A \to B) \wedge (B \to C) \wedge (C \to D) \wedge (D \to E) \\ \wedge (B \to G) \wedge (F \to G) \wedge (G \to H) \wedge (H \to B)$$

    vii.
$$(A \to B) \wedge (B \to C) \wedge (D \to B)$$

    viii.
$$(A \to (B \wedge C)) \wedge ((B \vee C) \to D) \wedge (D \to E) \wedge (E \to F) \wedge (F \to G) \wedge (G \to H)$$

[*8 marks*]

2. You are given the following inference rules, due to Gentzen:

$$\frac{}{\Gamma, A, B \vdash \Delta, A}\ (I)$$

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ (\wedge L) \qquad \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ (\vee R)$$

$$\frac{\Gamma, A \vdash \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ (\vee L) \qquad \frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ (\wedge R)$$

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma, B \vdash \Delta}{\Gamma, A \rightarrow B \vdash \Delta}\ (\rightarrow L) \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta}\ (\rightarrow R)$$

$$\frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ (\neg L) \qquad \frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ (\neg R)$$

Here, A and B are propositional expressions, Γ, Δ are finite sets of expressions, and Γ, A is shorthand for $\Gamma \cup \{A\}$.

An entailment $\Gamma \vdash \Delta$ is *valid* iff every valuation that makes each expression in Γ true makes some expression in Δ true.

(a) i. Explain what it means to say that an entailment, $\Gamma \vdash \Delta$ is **valid**. [*2 marks*]

ii. Show that the rule $(\vee L)$ has the property that its conclusion is invalid iff either one of its assumptions is invalid. [*4 marks*]

(b) Use the Gentzen rules to show that [*4 marks*]

$$P \rightarrow Q \vdash (Q \rightarrow R) \rightarrow (P \rightarrow R)$$

(c) Use the Gentzen rules to build an attempted proof of [*4 marks*]

$$P \rightarrow Q \vdash (P \rightarrow R) \rightarrow (Q \rightarrow R)$$

(d) Derive a counter-example from your attempted proof. (Show your working, and briefly justify your answer in terms of the particular properties of this set of rules.) [*2 marks*]

(e) i. What does it mean to say a set of rules is **sound**? [*2 marks*]

ii. Is this a sound set of rules? Briefly justify your answer. [*2 marks*]

3. (a) Express each of the following expressions in clausal form. [*2 marks*]

   i. $P \to Q$

   ii. $\neg((P \to R) \to (Q \to R))$

(b) Use resolution to determine whether there is a valuation satisfying the conjunction of these two expressions. [*3 marks*]

(c) Using the table provided record the search for a satisfying valuation for the four clauses given, using the two watched literals algorithm.

Start with the empty valuation, watching the first two literals in each clause. At each step of the search use one column of the table provided to record in the first 3 squares the truth values assigned to the atoms, P, Q, R, and in the remaining four, any changed position (1,2,3) of the watched literal for each of the four clauses. When you need to find a new literal to watch always choose the first one available (reading each expression left-to-right). Place a + in the watched literal square if no action is required..

P											
Q											
R											
$\neg P, Q, R$											
$\neg P, \neg Q, R$											
$\neg P, Q, \neg R$											
$\neg P, \neg Q, \neg R$											

Satisfying valuation: [*10 marks*]

(d) What is the invariant property of the watched literals that has to be maintained? [*2 marks*]

(e) What must you do when it is not possible to maintain the invariant?

(f) Sketch the search tree explored by this search. [*4 marks*]

(g) What invariant must be maintained for the single watched literal algorithm? [*2 marks*]

(h) What advantage is gained by watching two literals? [*2 marks*]

4. This question concerns the design of the user-interaction for an automated check-out machine.

An automated checkout machine has six major states
I: **I**dle –waiting for a user to arrive.
R: **R**eady to scan the next item
C: **C**hecking the weight of the item on the bagging scale.
A: **A**ssistance from a staff member is required because an item fails to scan.
W: **W**arning == alerting a staff member to possible fraud when the weight is not OK
C: **C**heckout – the machine is ready to collect payment.

There are also four minor states
C1:
C2:
R1:
R2:

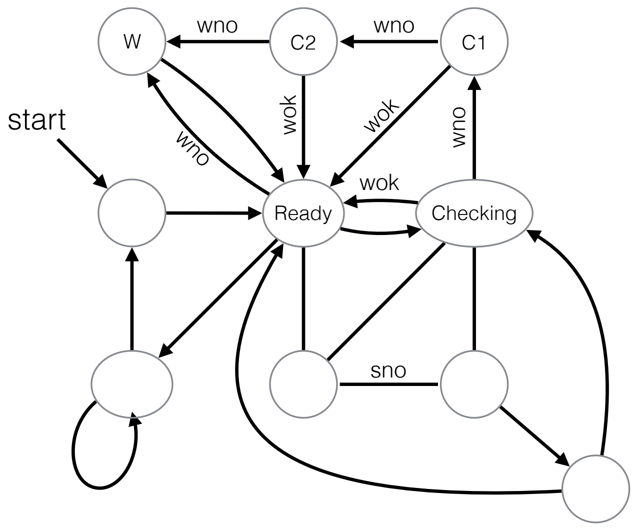
As a result of the user's actions, the machine receives inputs from the scanner, scales, and payment system as follows:
sok: scan ok
sno: scan not ok
wok: weight ok
wno: weight not ok
pok: payment ok
pno: payment failed

As well as these inputs generated by the user's actions, the machine has two buttons that generate the following user inputs
hi: the user wants to start a new session
pay: the user wants to finish scanning and pay

One further input can only come from an authorised staff member
aok: to indicate that a problem has been resolved— all is ok

QUESTION CONTINUES ON NEXT PAGE

The diagram shows a design for the user-interaction, with various details missing.



(a) Add labels for states and transitions, arrowheads, and, if necessary, new transitions to produce a finite state machine that attempts to satify the requirements listed in Q4b. [*8 marks*]

(b) Justify your design by explaining briefly how it meets each of the following requirements. If any of the requirements is ambiguous, or cannot be satisfied without adding further transitions, explain how this could be resolved. [*12 marks*]

i. The machine should allow for the following pattern of interaction.

> The user comes to an idle machine, says "Hi", then repeatedly scans items and places them on the bagging scale. The scale checks the weight of the item against the expected value. Eventually, when the user wants to pay, the machine collects payment.

ii. The **A**ssistance state should be reached if an item fails three time to scan.

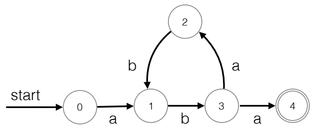
iii. The **W**arning state should be reached if an item fails three time to pass the weight check.

iv. The **W**arning state should be reached if the user removes items from the scale before paying.

v. Only a staff member should be able to make the machine leave the **W**arning state.

vi. A user should be able to leave the **A**ssistance state, if the problem is resolved.

5. (a) Which of the following strings are accepted by the NFA in the diagram? (The start state is indicated by an arrow and the accepting state by a double



border.)

i. abb

ii. ababba

iii. abababababa

iv. aba [*4 marks*]

(b) Write a regular expression for the language accepted by this NFA. [*2 marks*]

(c) Draw a DFA that accepts the same language. Label the states of your DFA to make clear their relationship to the states of the original NFA. [*10 marks*]

(d) For each of the following regular expressions, draw a non-deterministic finite state machine that accepts the language described by the regular expression.

i. $(x|y)(v|w)$

ii. $(x|(yz))^*$

iii. $(xy)^*|(yx)^*$

[*9 marks*]