Module Title: INF1-CL
Exam Diet (Dec/April/Aug): Aug 2015
Brief notes on answers:

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

A	B	$\neg A$	$B \vee \neg A$	$(B \vee \neg A) \to B$	$\neg((B \vee \neg A) \to B)$	$\neg((B \vee \neg A) \to B) \vee B$
$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$

   contingent

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

A	B	$A \vee \neg B$	$((A \vee \neg B) \to A)$	$(B \to A)$	$((A \vee \neg B) \to A) \to (B \to A)$
$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$
$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$
$\top$	$\top$	$\top$	$\top$	$\top$	$\top$

   tautology

(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. $\neg\neg A$ *128*

ii. $E \vee F$ *192*

iii. $B \oplus \neg D$ *128*

iv. $\neg A \oplus (B \wedge C)$ *128*

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

$48 = 2^3 \times 6$

vi.
$$\begin{aligned}(A \to B) \wedge (B \to C) \wedge (C \to D) \wedge (D \to E)\\ \wedge (B \to G) \wedge (F \to G) \wedge (G \to H) \wedge (H \to B)\end{aligned}$$

6

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

$96 = 2^4 \times 6$

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

10

[*8 marks*]

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

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

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

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

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

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

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]*

Every valuation V that makes every expression in Γ True, makes some expression in Δ True.

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]*

A counterexample to the conclusion makes everything in Γ True, and everything in Δ False. It also makes $A \vee B$ true, so it either makes A True or B True. In the first case, the valuation is a counterexample to the first assumption; in the second case, to the second assumption. Contrariwise, a counterexample to either assumption makes all of Γ True and all of Δ False. For the first assumption it makes A True; for the second, it makes B True. So, in either case $A \vee B$ is true, and so we have a counterexample to the conclusion.

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

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

$$\dfrac{\dfrac{\dfrac{\dfrac{}{Q \to R, P \vdash P, R}\ (I) \qquad \dfrac{\dfrac{}{Q, R, P \vdash R}\ (I) \quad \dfrac{}{Q, P \vdash Q, R}\ (I)}{Q, Q \to R, P \vdash R}\ (\to L)}{P \to Q, Q \to R, P \vdash R}\ (\to L)}{P \to Q, Q \to R \vdash P \to R}\ (\to R)}{P \to Q \vdash (Q \to R) \to (P \to R)}\ (\to R)$$

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

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

$$\dfrac{\dfrac{}{P \to Q, R, Q \vdash R}\,(I) \quad \dfrac{Q \vdash P, R \quad Q \vdash P, R}{P \to Q, Q \vdash P, R}\,(\to L)}{\dfrac{P \to Q, P \to R, Q \vdash R}{\dfrac{P \to Q, P \to R \vdash Q \to R}{P \to Q \vdash (P \to R) \to (Q \to R)}\,(\to R)}\,(\to R)}\,(\to L)$$

(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*]
Each rule has the property shown for $(\vee L)$ in 2(a)ii. So if either of the (identical) open assumptions in is invalid then each conclusion derived from it will be invalid. $\bar{P}Q\bar{R}$ provides a counterexample.

(e) i. What does it mean to say a set of rules is **sound**? *It means that any derivable sequent is valid.* [*2 marks*]

ii. Is this a sound set of rules? Briefly justify your answer. *Yes, since every rule has the property shown for $(\vee L)$ in 2(a)ii.* [*2 marks*]

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

i. $P \to Q$

$\{\neg P, Q\}$

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

$\{\neg P, R\}, \{Q\}, \{\neg R\}$

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

$$
\begin{array}{c|c|c|}
P \to Q & \neg P, Q & \neg P \\
\neg((P \to R) \to (Q \to R)) & \neg P, R & \\
 & Q & \\
 & \neg R & \\
\end{array}
$$

Resolve first on R. All remaining literals are pure. We cannot derive the empty clause, so the conjunction is satisfiable. [*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		$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$				
Q			$\top$	$\bot$		$\top$	$\top$				
R							$\top$				
$\neg P, Q, R$	*1,2*	*2,3*		R							
$\neg P, \neg Q, R$	*1,2*	*2,3*	R			*1,3*					
$\neg P, Q, \neg R$	*1,2*	*2,3*		$\neg R$			+				
$\neg P, \neg Q, \neg R$	*1,2*	*2,3*	$\neg R$			*1,3*	+				

Satisfying valuation: $P \wedge Q \wedge \neg R$ [*10 marks*]

(d) What is the invariant property of the watched literals that has to be maintained?

If one of the watched literals is set to False, then the other must be True [*2 marks*]

(e) What must you do when it is not possible to maintain the invariant? *Backtrack or fail.*

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

(g) What invariant must be maintained for the single watched literal algorithm? *No watched literal is false. Every watched literal is either True or Undefined* [*2 marks*]

(h) What advantage is gained by watching two literals? *Unit clauses are propagated* [*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

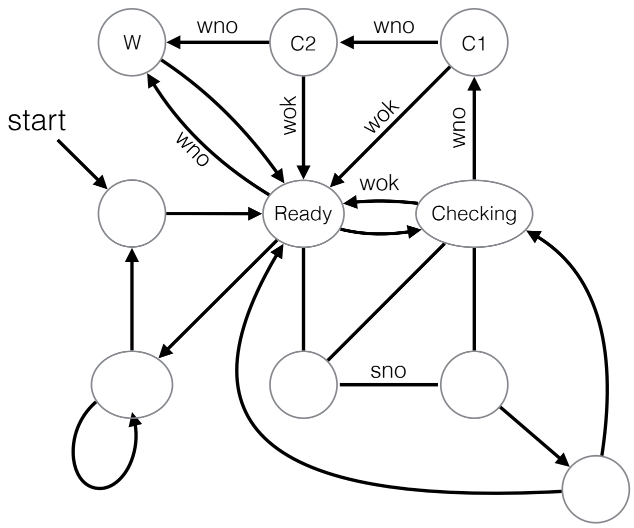
As well as these inputs generated by the user's actions, the machine has two buttons that generate the following user inputs

- 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

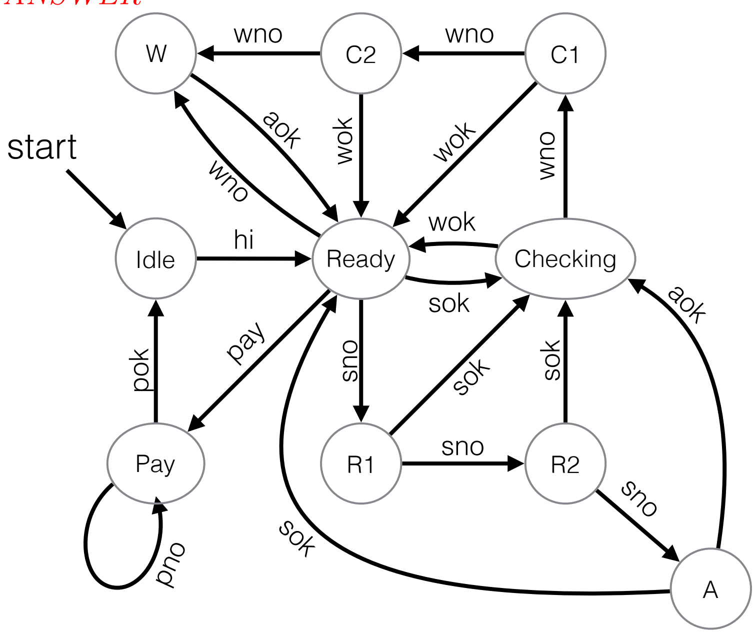
- aok: to indicate that a problem has been resolved— all is ok

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*]

ANSWER



(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.

Normal operation has the form ***(hi)((sok)(wok))*(pay)(pok)***

ii. The **A**ssistance state should be reached if an item fails three time to scan. *The only trace with three consecutive* ***sno*** *reaches state* ***W***

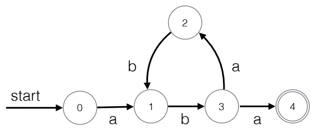
iii. The **W**arning state should be reached if an item fails three time to pass the weight check. *The only trace with three consecutive* ***wno*** *reaches state* ***W***

iv. The **W**arning state should be reached if the user removes items from the scale before paying. *This requirement introduces ambiguity, it conflicts the normal operation of the balance scale. The* ***wno*** *transition from* ***Ready*** *to* ***W*** *is certainly insufficient as the user could remove items when in states,* ***R1, R2, P, A****. Transitions labelled* ***wno*** *could be added from these to state* ***W****, but the ambiguity over required behaviour should be resolved.*

v. Only a staff member should be able to make the machine leave the **W**arning state. *The only transition leaving the state requires authorised access.*

vi. A user should be able to leave the **A**ssistance state, if the problem is resolved. *The* ***sok*** *transition achieves this.*

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 *No*

ii. ababba *Yes*

iii. abababababa *No*

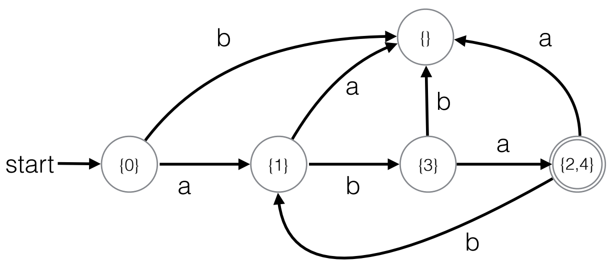
iv. aba *Yes*

[*4 marks*]

(b) Write a regular expression for the language accepted by this NFA. *ab(abb)*a*

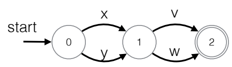
[*2 marks*]

(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.

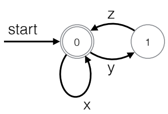


[*10 marks*]

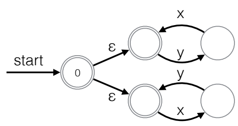
(d) For each of the following regular expressions, draw a non-deterministic finite state machine that accepts the language described by the regular expression.



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



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



iii. $(xy)^*|(yx)^*$ [*9 marks*]