COMP 2620 Draft Exam Solutions

① Truth Tables

1.

p	q	r	$(p \to q)$	$(p \to q) \to r$	$p \to (q \to r)$	$(q \to r)$	
1	1	1	1	1	1	1	
1	1	0	1	0	0	0	
1	0	1	0	1	1	1	
1	0	0	0	1	1	1	
0	1	1	1	1	1	1	
0	1	0	1	0	1	0	←
0	0	1	1	1	1	1	
0	0	0	1	0	1	1	←

They are different only if both p and r are false

2.

p	q	r	$p \vee q$	$p \vee q \to q \vee r$	$q \vee r$	$q \to \neg p$	$r \to q \to \bot$	$q \to \bot$	$\neg r$	
1	1	1	1	1	1	0	0	0	0	
1	1	0	1	1	1	0	1	0	1	
1	0	1	1	1	1	1	1	1	0	←
1	0	0	1	0	0	1	1	1	1	
0	1	1	1	1	1	1	0	0	0	
0	1	0	1	1	1	1	1	0	1	←
0	0	1	0	1	1	1	1	1	0	←
0	0	0	0	1	0	1	1	1	1	←

The argument becomes valid if we additionally assume q.

② Natural Deduction

1.

α_1	(1)	$(p \to q) \to (q \to p)$	A
α_2	(2)	p	A
α_2, α_3	(3)	q	A
α_3	(4)	$p \to q$	3, $\to$I
α_1, α_3	(5)	$q \to p$	1, 4, $\to$E
α_1, α_3	(6)	p	3, 5, $\to$E
α_1	(7)	$q \to p$	6, $\to$I

2.

α_1	(1)	$\exists x \forall y\, Rxy$	A
α_2	(2)	$\neg Rcc$	A
α_3	(3)	$\forall y\, Ray$	A
α_3	(4)	Raa	3, $\forall$E
α_5	(5)	$a = c$	A
α_3, α_5	(6)	Rcc	5, 4, =E
$\alpha_2, \alpha_3, \alpha_5$	(7)	$\perp$	2, 6, $\neg$E
α_2, α_3	(8)	$\neg(a = c)$	7, $\neg$I
α_2, α_3	(9)	$\neg a = c \wedge Raa$	8, 4, $\wedge$I
α_2, α_3	(10)	$\exists x(\neg x = c \wedge Rxx)$	9, $\exists$I
α_1, α_2	(11)	$\exists x(\neg x = c \wedge Rxx)$	1, 10, $\exists$E

3.

α_1	(1)	$\forall x(Px \vee Q)$	A
α_2	(2)	$\neg(\neg \exists x \neg Px \vee Q)$	A
α_3	(3)	$\neg \exists x \neg Px$	A
α_3	(4)	$\neg \exists x \neg Px \vee Q$	3, $\vee$I1
α_2, α_3	(5)	$\perp$	2, 4, $\neg$E
α_2	(6)	$\neg\neg \exists x \neg Px$	5, $\neg$I
α_2	(7)	$\exists x \neg Px$	6, $\neg\neg$E
could reorder (α_8 ~~α_2~~	(8)	$\neg Pa$	~~[illegible]~~ A
α_1	(9)	$Pa \vee Q$	1, $\forall$E
α_{10}	(10)	Pa	A
α_8, α_{10}	(11)	$\perp$	8, 10, $\neg$E
α_2, α_{10}	(12)	$\perp$	7, 11, $\exists$E
α_{13}	(13)	Q	A
α_{13}	(14)	$\neg \exists x \neg Px \vee Q$	13, $\vee$I2
α_2, α_{13}	(15)	$\perp$	2, 14, $\neg$E
α_1, α_2 ~~[illegible]~~	(16)	$\perp$	9, 12, 15, $\vee$E
α_1	(17)	$\neg\neg(\neg \exists x \neg Px \vee Q)$	16, $\neg$I
α_1	(18)	$\neg \exists x \neg Px \vee Q$	17, $\neg\neg$E

③ Soundness and Completeness

1, Induction Hypothesis 1 : Any row of a truth table in which all of Γ is 1, also has φ as 1.
Induction Hypothesis 2 : Any row in which all of Γ', and also φ, is 1, has ψ as 1.
Consider a row in which all of Γ and Γ' are 1.
By IH1, φ is 1.
Then by IH2, ψ is 1.

2, The main lemma in this case states that $p \vdash \varphi$ can be proved by natural deduction, and so can $\neg p \vdash \varphi$.
$p \vee \neg p$ is a theorem of classical logic, provable in natural deduction. We may apply $\vee E$ to this, and the two proofs from the main lemma, to prove φ without assumption.

3, Induction Hypothesis : $\Gamma \models \forall x \varphi$
This means that for all M, e satisfying all of Γ, and all $d \in D$, $\models_{M, e(x \mapsto d)} \varphi$
But D is not empty, so there exists $d \in D$ with this property
So $\models_{M,e} \exists x \varphi$, so $\Gamma \models \exists x \varphi$

4, Take unary P and constant c
Let M have universe $\{a, b\}$ with P interpreted as $\{a\}$ and c as a.
$\forall x P x [c/x]$ is $\forall x P c$ according to the bogus definition.
Now $\models_M \forall x P c$ iff for all environments e for x, $\models_{M,e} P c$, which is true because the interpretation of c is in the interpretation of P.
But $\models_{M, (x \mapsto a)} \forall x P x$ requires that for all environments for x, and in particular the map of x to b, $\models_{M, (x \mapsto a)(x \mapsto b)} P x$. But $(x \mapsto a)(x \mapsto b)$ is $(x \mapsto b)$, and it is not that case that $\models_{M, (x \mapsto b)} P x$, because b is not in the interpretation of P.

4 Modelling from English

1, Variables : a1 pass assignment 1

a2 pass assignment 2

p pass the course

Ox offered extra assignment

Px pass extra assignment

- $a1 \wedge a2 \rightarrow \neg Ox \wedge p$
- $(a1 \vee a2) \wedge \neg(a1 \wedge a2) \longrightarrow Ox$ (could also use XOR)
- $Ox \rightarrow (Px \rightarrow p \wedge p \rightarrow Px)$ (could also use IFF / $\leftrightarrow$).
- $p \rightarrow (a1 \wedge a2) \vee Px$ (could replace Px by Ox)

(It is harmless, but not necessary, to also require $Px \rightarrow Ox$)

2, Logic4Fun answer :

Sorts a. (Arguments)

s. (Meaningful Statements)

Vocabulary predicate p(s, a). (1st input is a premise of 2nd input)

function c(a) : s. (map arguments to conclusion)

predicate t(s). (is true)

predicate v(a). (is valid; could be left out if sort a is valid arguments).

~~Constants~~ name X : a.

name Y : a.

Constraints ALL x (ALL y (p(y,x) IMP t(y)) AND v(x) IMP t(c(x))).

v(X) AND v(Y).

t(c(X)) XOR t(c(Y)).

SOME x ((p(x,X) OR p(x,Y)) AND NOT

SOME y (x = c(y) AND v(y) AND

ALL z (p(z,y) IMP t(z)))).

(Note : could add NOT to the last constraint to prove validity, but accept answer either way).

Modelling from English, continued

2, First order logic answer:

constants X and Y

predicates A (unary, 'is an argument'), T (unary, 'is true'), V (unary, 'is valid'), P (binary, '1st ~~argument~~ input is a premise of 2nd input'), C (binary, '1st input is the conclusion of 2nd input')

(note: could also fold A & V together as 'is a valid argument').

$\forall x\,(Ax \rightarrow \exists y\, Cyx)$

$\forall x \forall y \forall z\,(Cyx \wedge Czx \rightarrow y = z)$

$\forall x\,(\forall y\,(Pyx \rightarrow Ty) \wedge Vx \rightarrow \exists z((Czx \wedge Tz)))$

$AX \wedge AY$

$(\exists x\,(CxX \wedge Tx) \vee \exists x\,(CxY \wedge Tx)) \wedge \neg(\exists x((CxX \wedge Tx) \wedge \exists x((CxY \wedge Tx))$

$\therefore\ \exists x\,((PxX \vee PxY) \wedge \neg\exists y\,(Cxy \wedge Vy \wedge \forall z\,(Pzy \rightarrow Tz)))$

3, $i \wedge \varphi\, \mathsf{U} G i$, where φ is the conjunction of

- $i \vee b \vee g \vee s$ (in one of the four regions).
- $i \rightarrow \neg(b \vee g \vee s)$

  $b \rightarrow \neg(i \vee g \vee s)$

  $g \rightarrow \neg(i \vee b \vee s)$ (one place at a time)

  $s \rightarrow \neg(i \vee b \vee g)$
- $Fb \wedge Fg \wedge Fs$ (eventually go to each place)
- $g \rightarrow \neg Xs$

  $s \rightarrow \neg Xg$ (no direct Gungahlin – South road)

(One could also add constraints to prevent staying in the same place, but this is not required)

- $Xi \rightarrow XGi$ (if North is next, we have reached end: prevents multiple stops in i)

Another example solution: take the conjunction of

- i
- XFi
- $X(\neg i\, \mathsf{U}\, p)$ for $p \in \{b, g, s\}$
- $G(p \rightarrow G\neg p)$ for p similarly
- $G(g \rightarrow \neg Xs)$ and $G(s \rightarrow \neg Xg)$
- $G(i \vee b \vee g \vee s)$
- $G(i \rightarrow \neg b \wedge \neg g \wedge \neg s)$
- $G(b \rightarrow \neg i \wedge \neg g \wedge \neg s)$
- $G(g \rightarrow \neg i \wedge \neg b \wedge \neg s)$
- $G(s \rightarrow \neg i \wedge \neg b \wedge \neg g)$

⑤ Tableaux

1.

(1) $T: p \lor q$ ✓

(2) $T: q \to r$ ✓

(3) $T: \neg(r \lor \bot)$ ✓

(4) $F: p$

(5) $T: p$ from (1)	(6) $T: q$ from (1)
$\times$ (1, 5)	(7) $F: r \lor \bot$ from (3) ✓
	(8) $F: r$ from (7)
	(9) $F: \bot$ from (7)

(10) $F: q$ from (2)	(11) $T: r$ from (2)
$\times$ (6, 10)	$\times$ (8, 11)

2. I will give the complete tableau but students are welcome to ignore branches in search of a satisfying model

(1) $T: r \lor s$ ✓

(2) $F: p \to q \to r \land s$ ✓

(3) $T: p$ from (2)

(4) $F: q \to r \land s$ from (2) ✓

(5) $T: q$ from (4)

(6) $F: r \land s$ from (4) ✓

(7) $T: r$ from (1)		(8) $T: s$ from (1)	
(9) $F: r$ from (6)	(10) $F: s$ from (6)	(11) $F: r$ from (6)	(12) $F: s$ from (6)
$\times$ (7, 9)			$\times$ (8, 12)

Either - p, q, r true and s false

or - p, q, s true and r false

3.

(1) $T: \exists x(Px \to Qx)$ ✓

(2) $T: \exists x\, Px$ ✓

(3) $F: \exists x\, Qx$ ✓

(4) $T: Pa$ from (2)

(5) $T: Pb \to Qb$ from (1) ✓

(6) $F: Qa$ from (3)

(7) $F: Qb$ from (3)

(8) $F: Pb$ from (5)	(9) $T: Qb$ from (5)
	$\times$ (7, 9)

The sequent is not valid.

Tableaux Continued

4.

(1) $T : Fp$ ✓

(2) $T : G(p \to XG\neg p)$ ✓

(3) $T : p \to XG\neg p$ from (2) ✓

(4) $T : XG(p \to XG\neg p)$ from (2) ✓

(5) $T : p$ from (1)

(6) $T : XG\neg p$ from (3) ✓

$\ddagger$

(7) $T : G(p \to XG\neg p)$ from (4) ✓

(8) $T : G\neg p$ from (6) ✓

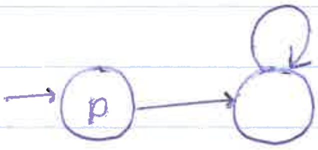
(9) $T : \neg p$ from (8) ✓

(10) $T : XG\neg p$ from (8)

(11) $F : p$ from (9)

(12) $T : p \to XG\neg p$ from (7) ✓ Adds no new formula either way

(13) $T : XG(p \to XG\neg p)$ from (7)



$\ddagger$

[Loop, 1] because exact copy of above node.

5.

(1) $T : G(\neg p \, U \, q)$ ✓

(2) $T : G(\neg q \, U \, p)$ ✓

(3) $T : \neg p \, U \, q$ from (1) ✓

(4) $T : XG(\neg p \, U \, q)$ from (1)

(5) $T : \neg q \, U \, p$ from (2) ✓

(6) $T : XG(\neg q \, U \, p)$ from (2)

(7) $T : q$ from (3)

(8) $T : p$ from (5)

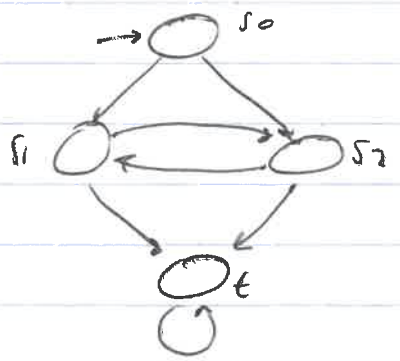
$\ddagger$

[Loop, 1] because exact copy of above node

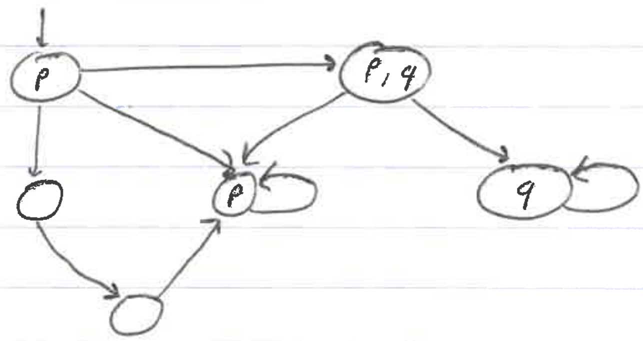
(6) Semantics

1. $e(x) = 0$, $e(y) = 0$
   (do not define e's action on z)

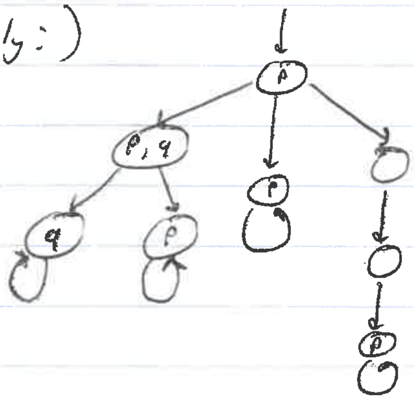
2.



3.



(or, less succinctly but more directly:)

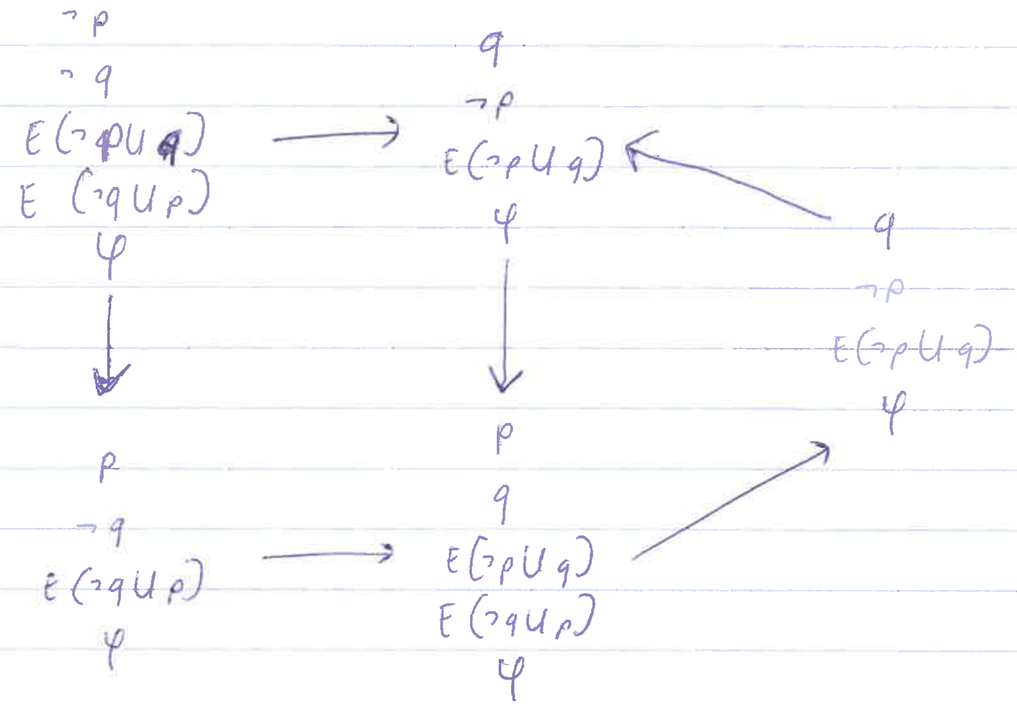


4. $\models_{M,\sigma} X\, G\, \varphi$
   iff $\models_{M,\sigma_{\geq 1}} G\, \varphi$
   iff for all $i \geq 1$, $\models_{M,\sigma_{\geq i}} \varphi$
   iff for all $i \geq 0$, $\models_{M,(\sigma_{\geq i})_{\geq 1}} \varphi$
   iff for all $i \geq 0$, $\models_{M,\sigma_{\geq i}} X\, \varphi$
   iff $\models_{M,\sigma} G\, X\, \varphi$

5. $\models_{M,s} E(A(\varphi))$
   iff there exists σ with $\sigma_0 = s$ and $\models_{M,\sigma} (A(\varphi))$
   iff $\models_{M,\sigma_0} A(\varphi)$
   But $\sigma_0 = s$, so this is just $\models_{M,s} A(\varphi)$.

(5) Model Checking

1, Where φ is the main proposition:



2, SCC 1 : S_0

SCC 2 : S_5

SCC 3 : S_7

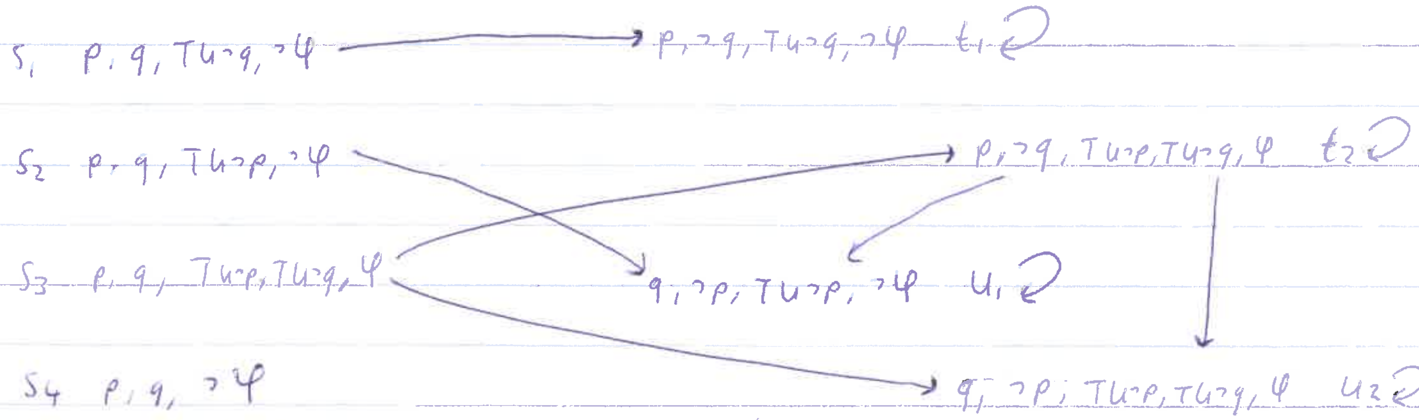
SCC 4 : S_3, S_4

SCC 5 : S_2

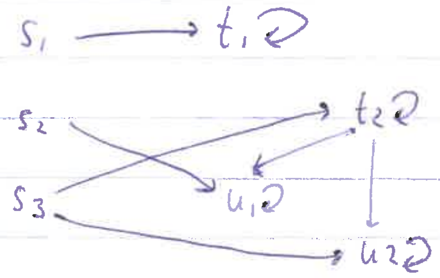
Satisfy $EG\varphi$: S_0, S_3, S_4, S_5, S_7

3, $Gp \vee Gq \equiv \neg(\neg Gp \wedge \neg Gq) \equiv \neg(\neg\neg F\neg p \wedge \neg\neg F\neg q) \equiv \neg(F\neg p \wedge F\neg q)$

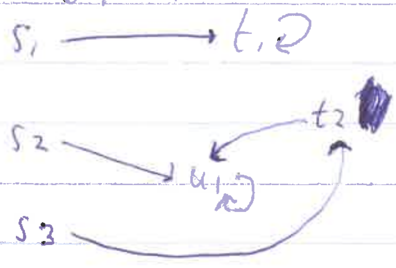
$\equiv \neg((T U \neg p) \wedge (T U \neg q))$. Call the conjunction φ



Deleting dead ends:



Deleting endlessly postponed eventualities:



6262 replacement question

First order logic:

constant s (start state)

binary predicate T (transition)

$\forall x \exists y\ Rxy$

Then add a unary predicate P (satisfies p)

$\forall x (Tsx \longrightarrow Px)$

Logic 4 Fun:

Sorts : state.

Vocabulary : name start (state).

predicate T (state, state). (transition)

Constraints : ~~ALL x (T(start, x) IMP~~

ALL x SOME y T(x,y).

Add new Vocabulary : predicate P(state). (satisfies p)

Constraint : ALL x (T(start, x) IMP P(x)).

Advantages:

- Expressivity: 1st order logic cannot express F, G, U due to compactness / Löwenheim-Skolem
- Computability: 1st order logic ~~[illegible]~~ uncomputable, temporal logic most questions are decidable.

~~Useful class of models~~

(No marks for answers which are true of both systems, e.g. 'has a useful class of models' or 'powerful enough to express useful properties'.