

THE UNIVERSITY OF MELBOURNE
SCHOOL OF COMPUTING AND INFORMATION SYSTEMS
COMP30026 MODELS OF COMPUTATION

”You’re gonna have a bad time.”

Q1 A

Multiple approaches we can use:

Truth Tables:

Prove logical consequence via: ”Everywhere we find a 1 in ψ we should find in γ ”

Answer is:

$$Q \Rightarrow R$$

Q1 B

From the information provided, we can write:

$$\begin{aligned} r &\Leftrightarrow s \\ (\neg r \wedge \neg s) &\Rightarrow \neg p \\ r \vee s \vee \neg p \\ q &\Rightarrow r \oplus s \\ r \wedge s &\Rightarrow q \end{aligned}$$

From these we can derive:

$$\begin{aligned} r \vee \neg q \\ \neg p \\ \neg q \\ r \Rightarrow q \\ \neg r \end{aligned}$$

Note: We do not need to show working out for this, but showing working out can result in more generous marks.

Important Quotes from Harold:

”If there is a result we have shown in the lectures, we can take that for granted”.

”If there is a language that we’ve come across - you don’t have to prove that language.”

Answer is: MacGuffin cannot show films that week.

Q2 A

Approach: Find an interpretation which makes this formula true and one that makes it false.
For true F, we can say it represents the equal predicate.
For false F, we can say it represents the less than predicate.
Because it's $F \wedge G$, we don't even have to interpret G .

Note: Make sure you define your domain as well!

Typical answer syntax:

Consider dom Z, let $P(x, y)$ be $x < y$.

Q2 B

Approach: Same as above.

Q2 C

Approach: Negate G, and we get $\neg P(x, y) \vee P(y, x)$, Negate H and we get $\exists(\neg P(x, x) \wedge \forall y P(y, x))$. When we try and resolve after skolemization then we get something unsatisfiable instantly, which means it is valid.

Answer: $G \models H$

Q3 A

Approach: Figure out the connectives, and then build up the formula from that.

"No cat is a friend of x": $\neg \exists y (C(y) \Rightarrow F(y, x))$

Push negation of above in : $\forall y (C(y) \Rightarrow \neg F(y, x))$

"x eats pasta": $\exists z (P(z) \Rightarrow E(x, z))$

Now we put it together:

"If dog does thing then do that thing" :

$$\forall xyz (D(x) \wedge \forall y (C(y) \Rightarrow \neg F(y, x))) \Rightarrow \exists z (P(z) \Rightarrow E(x, z))$$

Note: If the terms inside the predicate formula is \wedge , then you can push them in, otherwise you can't.

Q3 B

Approach: This is all mechanical, so just do the formulas we learnt with substitution and pushing negations in.

Q3 C

Approach: First things first, negate the end statement! Then we can just resolve.

Begin with $\neg L(b, b)$, and start resolving e.g. with $L(b, x) \vee \neg L(x, c)$ etc.

Q4 A

Tick the following: *abba*, *ababba*, *bababa*

Q4 B

Skipped.

Q4 C

Approach 1: Write the DFA for a^*b^* and b^*a^* separately, and then combine with the product. (This will probably take too long.)

Approach 2: Just look at it and consider each of the parts separately.

Answer: $a^* \cup b^*$

Q5 A

Skipped.

Q5 B

Q: How do you see the smallest regular expression?

A: You just do.

Q5 C

Approach: Note how regular expressions are a subset of context free grammars, so we can express any particular string in a language in regexp.

This is a trick question:

S can generate literally anything since it goes to T (the first two T rules generate any string), so we can disregard the rest.

The regular language is $(a \cup b)^*$.

Q6 C

You won't get a question like this at all? wat.

Note: Induction will not be tested.

Q6 B

Approach: Simply create the same string from this language using alternative pathways.

In this case we can just use both the S's in different order to generate the string "a a a a a...."

Q6 C

Answer:

$$S \rightarrow \epsilon |aaaa|a^2|a^3|....|a^{16}|a^{18}T$$

and

$$T \rightarrow \epsilon |aT$$

Note: Make sure you actually write all the rules in the exam, or alternatively write " a^k where $k \dots$ "

Be careful with your linting.

Q7 A

Approach: Every time we want to show a subset, show that for everything F it's also a member of G . Here in the example they give, y is at least one of the sets in F and z is the member of all sets in G .

Q7 B

Approach: This looks really complicated, but you need to look past the illusion 100 skills. We can interpret this as something that it's something that's definitely in the first part $(L \ M)^*$ but definitely not in the second part $(L * \ M^*)$.

Answer: In this case, it's actually the empty set as $(L * \ M^*)$ can't be empty.

Q7 C

Approach: This looks really hard, but just start trying with small cases, e.g.

$$L = a, b \quad M = b$$

From this we can see that the first part $(a \cup b * \ b^*)$, and the second part $ab \ b$ which results in strings with only a .

Q8 A

Approach: It's a function if the first number isn't mapped to multiple things, and it's total if the entire relation maps their first numbers to stuff.

Q8 B

I missed this.

Q9

Approach: Draw the first 4+1 states, and then draw the reject states going back off the tape.

Move along the tap and read b's and a's until we get to 4th last symbol, and accept an a, then start moving back until we can reject.

This is definitely decidable as it's a regular expression.