

Concordia University Department of Computer Science and Software Engineering

SOEN 331 Section S: Formal Methods
for Software Engineering

Assignment 4

Mohammad Ali Zahir - 40077619

Marwa Khalid - 40155098

November 23, 2022

Date of Submission: December 2, 2022

Contents

1	Our assignment	3
---	----------------	---

1 Our assignment

1. (10 pts) Find a logically equivalent formula for $\phi \mathbf{W} \psi$ and provide a short reasoning to support your answer. Represent this equivalence between the two expressions with the appropriate logical connective, and support your reasoning.

Solution:

The logically equivalent formula for this would be:

$$(\phi \mathbf{W} \psi) \equiv (\phi \mathbf{U} \psi) \vee \Box(\psi)$$

These are equivalent, because the principle of the strong until operator \mathbf{U} . Since the ψ is never guaranteed to be true, we would need to add an extra or statement because if the statement ψ becomes true it would mean that ϕ can't be true. This means that ϕ would be true until a certain condition (ψ is true) is met.

2. (10 pts) Find a logically equivalent formula for $\phi \mathbf{U} \psi$, and provide a short reasoning to support your answer. Represent this equivalence between the two expressions with the appropriate logical connective, and support your reasoning.

Solution:

The logically equivalent formula for this would be:

$$(\phi \mathbf{U} \psi) \equiv (\phi \mathbf{W} \psi) \wedge \bigcirc\Diamond(\psi)$$

These are equivalent, because the principle of the strong until operator \mathbf{U} . We know that this ψ will eventually become true. We then now that we can use the weak until clause with an eventually operator, because the only way that ϕ is not true is when the ψ is not true. We add the and operator, because we want to insure that the next one we actually get ψ as being true, because this may never happen.

3. (10 pts) Find a logically equivalent formula for $\phi \mathbf{R} \psi$ in terms of \mathbf{W} , and provide a short reasoning to support your answer. Represent this equivalence between the two expressions with the appropriate logical connective, and support your reasoning.

Solution:

The logically equivalent formula for this would be:

$$(\phi \mathbf{R} \psi) \equiv (\phi \mathbf{W} \psi) \wedge \Diamond(\psi)$$

These are equivalent as the weak until will make the operation hold until something is triggered. We need the second part of the equation to guarantee that ψ will eventually has to be true, because if it is not then, this will never hold. Hence the and statement.

This paragraph refers to Questions 4 - 5: Consider a railroad with a single rail and a road level-crossing. We introduce the following propositions that represent events:

a : A train is approaching.

b : The barrier is down

c : A train is crossing

l: A light is blinking

4. (15 pts) Express each of the following requirements formally. For each one, proceed to find a logically equivalent formula that captures the safety property of the system (i.e. in terms of “something bad never happens”):

- (a) (5 pts) When a train is crossing, the barrier must be down.

Solution:

$$\Box(c \rightarrow \Box b)$$

- (b) (5 pts) If a train is approaching or crossing, then the light must be blinking.

Solution:

$$\Box(a \vee c \rightarrow \Box l)$$

- (c) (5 pts) If the barrier is up and the light is off, then no train is coming or crossing.

Solution:

$$\Box(\neg(b \wedge l) \rightarrow \neg \Box(a \vee c))$$

5. (10 pts) Express each of the following requirements formally in terms of the liveness property (i.e. in terms of “something good eventually happens”):

(a) (5 pts) When a train is approaching, it will eventually cross..

Solution:

$$\Box(a \rightarrow \Diamond c)$$

(b) (5 pts) When a train is approaching and no train is crossing, then the barrier will eventually go down before the train crosses.

Solution:

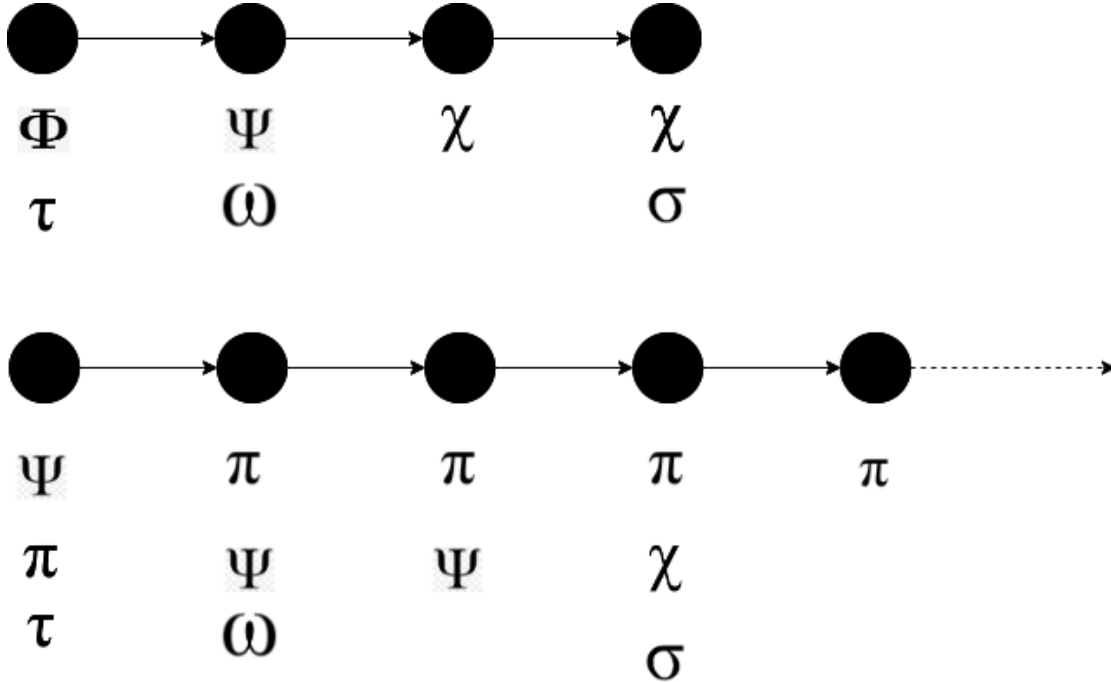
$$\Box((a \wedge \neg c) \rightarrow \Diamond(\neg b \text{ U } c))$$

6. (45 pts) The behavior of a program is expressed by the following temporal formula:

$$\Box \left[\begin{array}{l} \text{start} \rightarrow (\phi \oplus \psi) \\ \\ \text{start} \rightarrow \tau \\ \\ \phi \rightarrow \bigcirc(\psi \text{ U } \chi) \\ \\ \psi \wedge \tau \rightarrow \bigcirc(\psi \text{ W } \chi) \\ \\ \tau \wedge \bigcirc\psi \rightarrow \bigcirc\omega \\ \\ \psi \wedge \omega \rightarrow \bigcirc^2\chi \\ \\ \omega \wedge \bigcirc^2\chi \rightarrow \bigcirc^2\sigma \\ \\ \psi \wedge \bigcirc\sigma \rightarrow \bigcirc^2\pi \\ \\ \psi \wedge \tau \rightarrow \sigma \text{ R } \pi \\ \\ \phi \wedge \bigcirc\psi \rightarrow \bigcirc^2\chi \end{array} \right]$$

(a) (20 pts) Visualize all models of behavior.

Solution:



(b) (10 pts) Is the set of requirements satisfiable in all models of behavior? Explain why or why not.

Solution:

This set of requirements is not satisfiable in all behaviors. The first timeline, the top one terminates fine, hence it has no problems. The second one however has the operation **R** which means strong release. The specific condition that is using the strong release, would be the: $\psi \wedge \tau \rightarrow \sigma \text{ R } \pi$

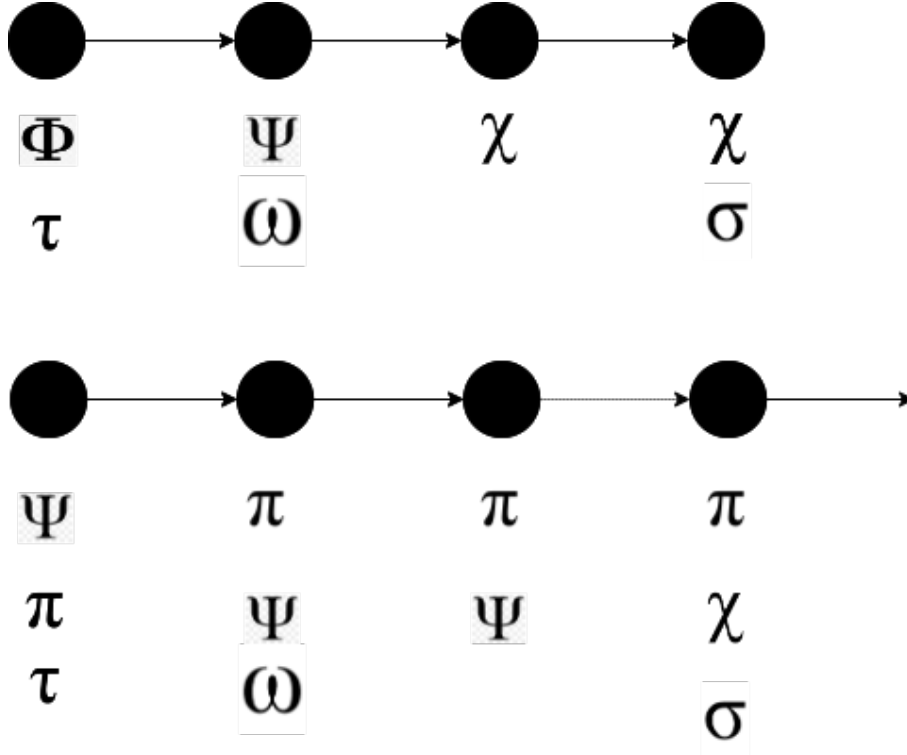
Hence at $i = 3$, is when our strong release should have ended, but at $i = 4$, we see an extra π which is not supposed to be there. **Hence this is not valid for all requirements.**

(c) (10 pts) In the case where the set of requirements is not satisfiable, what modification(s) to the requirements would you make (you may temporarily assume the role of a stakeholder) in order to achieve satisfiability.

Solution:

As the stakeholder, we are allowed to do change the program requirements how we want them to be. Hence the best way to acheive satisfiability would be the remove the condition that cause the problem in the second timeline which would be: $\psi \wedge \bigcirc\sigma \rightarrow \bigcirc^2\pi$

Hence after we remove that condition, our new timelines look like:



Here we can see now that $i+4$ is removed and $i + 3$ is where the strong release ends and it **hence satisfies the behaviour for all requirements now**.

- (d) (5 pts) Having resolved any possible conflicts in requirements, specify conditions (models of behavior), if any exist, under which the program can terminate. If none exist, please indicate so.

Solution: