Prof. Dr. A. Podelski, Dr. B. Westphal
L. Musliu

Wintersemester 2017/18

Real-Time Systems

http://swt.informatik.uni-freiburg.de/teaching/WS2017-18/rtsys

Exercise Sheet 3

Early submission: Monday, 2017-11-20, 14:00 Regular submission: Tuesday, 2017-11-21, 14:00

Exercise 1 — Validity (5/20 Points)

Choose one of the following blocks (i) or (ii).

Let P, Q, R be state assertions. Which of the formulae (a) – (c) (in your chosen block) is valid? Explain your claim or give a counterexample. Prove that (d) (in your chosen block) is valid.

(i)
a) $\neg\lceil P\rceil \implies \lceil\neg P\rceil$
b) $(\lceil P\rceil \wedge \lceil Q\rceil) \impliedby \lceil P \wedge Q\rceil$
c) $\Diamond(\lceil P\rceil \wedge \lceil Q\rceil) \implies (\Diamond\lceil P\rceil \wedge \Diamond\lceil Q\rceil)$
d) $\lceil\rceil \implies \int P = 0$

(ii)
a) $\neg\lceil P\rceil \impliedby \lceil\neg P\rceil$
b) $(\lceil P\rceil \wedge \lceil Q\rceil) \implies \lceil P \wedge Q\rceil$
c) $\Diamond(\lceil P\rceil \wedge \lceil Q\rceil) \impliedby (\Diamond\lceil P\rceil \wedge \Diamond\lceil Q\rceil)$
d) $\lceil\neg P\rceil \implies \int P = 0$

Exercise 2 — Writing DC Formulae (10/20 Points)

A traffic light for pedestrians is modelled by the observables 'Light' of data type $\{\mathsf{red}, \mathsf{yellow}, \mathsf{green}\}$ and 'Button' of data type $\{\mathsf{press}, \mathsf{release}\}$.
Formalise the following requirements using Duration Calculus:

(i) The button is never pressed when the lights show green. (2)

(ii) If the button is pressed, it takes at most 125 time units until green is shown. (2)

(iii) Green phases are at least 25 time units long. (3)

(iv) Within 3600 time units, the lights should not show green for more than 1000 time units. (3)

Hint: Explain your understanding of the requirement in natural language as precise as you can; formalise your understanding; explain your formalisation. That is, for each task, "test" your formula on some evolutions, i.e. give at least one positive and one negative example evolution and argue that your formula behaves adequately.
[What is an adequate meaning of "behaves adequately"? ;-)]

Exercise 3 — A Correct Level Crossing Controller [OD08] (5/20 Points)

We can abstractly model a rail-road level crossing by the observables

- 'Track' with domain $\{\mathsf{empty}, \mathsf{appr}, \mathsf{cross}\}$,
- 'Gate' with domain $\{\mathsf{open}, \mathsf{moving}, \mathsf{closed}\}$.

The track observable represents the presence of the train with two logical regions of the crossing. It is 'empty' if there is no train near or on the crossing, it is 'appr' if a train is near the crossing (approaching), and 'cross' if the train is in the area where road and tracks intersect.
The gate can be open, moving (up or down), or closed.
We use the following abbreviations:

E stands for $\mathsf{Track} = \mathsf{empty}$
A stands for $\mathsf{Track} = \mathsf{appr}$
X stands for $\mathsf{Track} = \mathsf{cross}$

O stands for $\mathsf{Gate} = \mathsf{open}$
C stands for $\mathsf{Gate} = \mathsf{closed}$

Consider the following DC properties:

$$\Box(\lceil X \rceil \implies \lceil C \rceil) \qquad \text{('Safety')}$$
$$(\lceil E \rceil \,;\, \mathit{true}) \vee \lceil \rceil \qquad \text{('Init')}$$
$$\Box((\lceil E \rceil \,;\, \mathit{true} \,;\, \lceil X \rceil) \implies \ell \geq \varepsilon) \qquad \text{('T-Fast')}$$
$$\Box((\lceil \neg E \rceil \wedge \ell \geq \varepsilon) \implies \mathit{true} \,;\, \lceil C \rceil) \qquad \text{('G-Close')}$$

(i) Explain informally the meaning of each of these formulae. (1/5)

Hint: don't just "read them out" using the names of the operators, but relate them to the behaviour of entities of the real world.

(ii) We distinguished requirements, design decisions, and assumptions.
Which of the formulae serves which purpose here? Briefly explain. (1/5)

(iii) Prove the validity of the following implication by using the DC semantics: (3/5)

$$\mathsf{Init} \wedge \mathsf{T\text{-}Fast} \wedge \mathsf{G\text{-}Close} \implies \mathsf{Safety}$$

Hint: the last task may be more difficult than 3 "exercise point units"; just see how far you get — if you like, nobody stops you from constructing a nice proof.

Exercise 4 — Writing More DC Formulae (5 Bonus)

One of the proposals for Exercise 1 on Exercise Sheet 1 was the anti-lock braking system (ABS).
A simplified model considers the observables:

- $P : \{0, 1\}$ – 1 models "brake pedal pressed"
- $L : \{0, 1\}$ – 1 models "locked wheel detected"
- $B : \{0, 1\}$ – 1 models "brake applied"

An example evolution is shown in Figure 1 on page 3: Braking is requested and the brake activates (disregarding reaction times for the moment), some time later, a blocking wheel is detected, and the brake is deactivated for short durations as long as the lock persists. While the wheel is blocked, the brakes should be deactivated and reactivated 10 times per second such that the brakes should be deactivated for exactly half of the time.

(i) A first, nice (and wrong) attempt on formalising the requirement yields the following formula:

$$\Box(\lceil L \rceil \implies \textstyle\int B = 0.5\ell)$$

Explain, using Figure 1, that this formula will not be satisfied by a correct controller. (1)

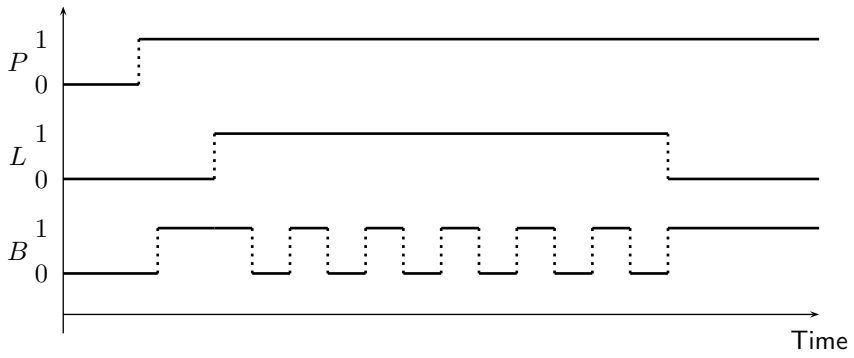

Figure 1: Anti-lock braking system: example evolution.

(ii) Based on the analysis of the previous task, the requirement is refined by adding "phases of activation and deactivation should be of equal length".

Propose a formalisation of the refined requirement. (2)

(iii) Your proposal from the previous task may still not be satisfied by evolutions which one would consider to be examples of acceptable controller behaviour.

Explain the problem (or why there is none), and either refine the requirement further to match your proposal or add an assumption on the plant behaviour (inputs are pedal and lock sensor). (2)

References

[OD08] Ernst-Rüdiger Olderog and Henning Dierks. *Real-Time Systems - Formal Specification and Automatic Verification*. Cambridge University Press, 2008.