EECS 4315 3.0 Mission Critical Systems

A Solution of Midterm

9:00–10:00 on March 2, 2020

1 (2 marks)

Why do we verify systems? (The answer is *not* "to find bugs.")
Answer: It is all about money and safety (slides and page 1 and 2 of the textbook).
Marking scheme: 2 marks for the mention of either money or safety.

2 (4 marks)

Model checking and theorem proving are two approaches to verification.

(a) Mention three advantages of model checking over theorem proving.

1.
2.
3.

Answer:

- Model checking is automatic.
- Model checking is (relatively) fast.
- Model checking provides counter examples.
- Temporal logics can easily express many properties.

(Section 1 of the paper by Clarke.)

Marking scheme: 1 mark for each advantage. (The textbook mentions other strengths of model checking as well, but some of them are shared with theorem proving.)

(b) Mention one disadvantage of model checking.

Answer: The state space explosion problem (slides, Section 1 of the paper by Clarke and page 15 of the textbook).

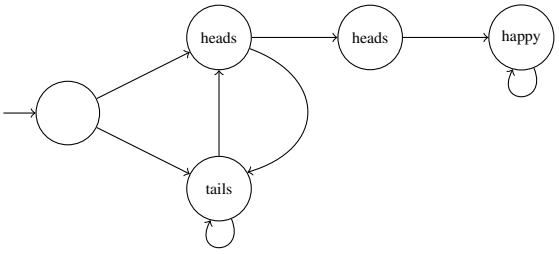
Marking scheme: 1 mark for the mention of the state space explosion problem.

3 (5 marks)

Assume I am flipping a coin and I become happy if I flip heads twice in a row. Model this as a labelled transition system. The labelled transition system has three labels: heads, tails, and happy. Once two heads in a row have occurred, the system should transition to a state that is labelled with happy and that has a transition to itself.

(a) Draw the labelled transition system.

Answer: There are many different ways to capture this system. For example, the following does.



Marking scheme: 0.5 mark for a state labelled happy with a self loop, 0.5 mark for two states labelled heads, 0.5 mark for one state labelled tails, 0.5 mark for correct transitions between the states labelled heads and happy.

(b) Formally define the labelled transition system drawn in part (a).

Answer: $\langle \{1,2,3,4,5\}, \{\text{heads}, \text{tails}, \text{happy}\}, \{1\}, \{(1,2),(1,3),(2,3),(3,2),(3,3),(2,4),(4,5),(5,5)\}, \{1 \mapsto \emptyset, 2 \mapsto \{\text{heads}\}, 3 \mapsto \{\text{tails}\}, 4 \mapsto \{\text{heads}\}, 5 \mapsto \{\text{happy}\}\}\rangle$

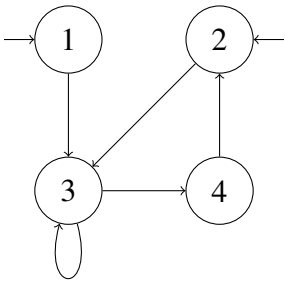
Marking scheme: 0.5 mark if the set of states corresponds to the set of vertices drawn in part (a). 0.5 mark for $\{\text{heads}, \text{tails}, \text{happy}\}$ as the set of labels. 0.5 mark if the transitions form a set of state pairs. 0.5 mark if the set of transitions corresponds to the set of edges drawn in part (a). 0.5 mark if the labelling function maps states to *sets* of labels. 0.5 mark if the labelling function corresponds to the labelling of part (a).

4 (2 marks)

Consider a traffic light. Give an LTL formula that formalizes the requirement that the light must change colour in the following sequence: red, green, and amber.
Answer: $\Box((\text{red} \Rightarrow \bigcirc\text{green}) \wedge (\text{green} \Rightarrow \bigcirc\text{amber}) \wedge (\text{amber} \Rightarrow \bigcirc\text{red}))$
Marking scheme: 0.5 mark for using $\Box$. 0.5 mark for using $\bigcirc$. 1 mark for an LTL formula that captures the requirement correctly.

5 (5 marks)

Consider the following labelled transition system.



Note that states 1 and 2 are both initial. States 1 and 2 have label a. States 2 and 3 have label b. State 4 has label c. For each of the following LTL formulas, determine if that formula holds for the above labelled transition system. A simple yes or no suffices.

(a) a

(b) b

(c) $\bigcirc b$

(d) $\bigcirc \bigcirc c$

(e) $\bigcirc \bigcirc \neg a$

(f) $\Diamond b$

(g) $\Box(a \vee c)$

(h) $a \,\mathrm{U}\, b$

(i) $b \,\mathrm{U}\, a$

(j) $a \,\mathrm{U}\, (b \,\mathrm{U}\, c)$

Answer:

(a) yes

(b) no

(c) yes

(d) no

(e) yes

(f) yes

(g) no

(h) yes

(i) yes

(j) no

Marking scheme: 0.5 mark for each correct answer.

6 (3 marks)

Let f be an arbitrary LTL formula. Which of the following equivalences hold? If the equivalence holds, give a proof. If the equivalence does not hold, provide a labelled transition system (and a specific choice for f) and argue that one of the two LTL formulas holds and the other LTL formula does not hold.

(a) $\Box f \equiv f \wedge \bigcirc \Box f$

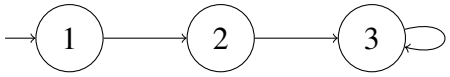
Answer: The formulas are equivalent. Let TS be an arbitrary labelled transition system. Let p be a path starting in an initial state. Then

$$
\begin{aligned}
& p \models \Box f \\
\text{iff } & \forall i \geq 0 : p[i..] \models f \\
\text{iff } & p[0..] \models f \text{ and } \forall i \geq 1 : p[i..] \models f \\
\text{iff } & p \models f \text{ and } \forall i \geq 0 : p[(i+1)..] \models f \\
\text{iff } & p \models f \text{ and } \forall i \geq 0 : p[1..][i..] \models f \\
\text{iff } & p \models f \text{ and } p[1..] \models \Box f \\
\text{iff } & p \models f \text{ and } p \models \bigcirc \Box f \\
\text{iff } & p \models f \wedge \bigcirc \Box f
\end{aligned}
$$

Marking scheme: 0.5 mark for the correct characterization of $\Box f$. 0.5 mark for the correct manipulation of $p[(i+1)..]$ and $p[1..][i..]$. 0.5 mark extra if the overall proof is correct.

(b) $\Diamond f \equiv f \vee \bigcirc \Diamond \bigcirc f$

Answer: The formulas are not equivalent. For f we choose the atomic propostion a. We consider the following labelled transition system.



and the following labelling function

$$\begin{array}{lcl} \ell(1) & = & \emptyset \\ \ell(2) & = & \{a\} \\ \ell(3) & = & \emptyset \end{array}$$

This labelled transition system satisfies $\Diamond a$ since state 2 has label a. However, the labelled transition system does not satisfy $a \vee \bigcirc\Diamond \bigcirc a$ since the first and third state are not labelled a.

Marking scheme: 0.5 mark for making an appropriate choice for f. 0.5 mark for an appropriate labelled transition system. 0.5 mark for arguing why the labelled transition system satisfies the one property but not the other.

7 (4 marks)

We extend the syntax of CTL with state formulas of the form $\exists(f \mathrm{R} g)$, where f and g are state formulas. For a state s, we define

$$s \models \exists(f \mathrm{R} g) \text{ iff } \exists p \in \mathit{Paths}(s) : \forall i \geq 0 : (p[i] \models g \text{ or } \exists 0 \leq j < i : p[j] \models f)$$

(a) Characterize $\mathit{Sat}(\exists(f \mathrm{R} g))$ in terms to $\mathit{Sat}(f)$, $\mathit{Sat}(g)$, succ and $\mathit{Sat}(\exists(f \mathrm{R} g))$. Provide a derivation of your characterization (marks will only be given for the derivation).

Answer:

$s \in \mathit{Sat}(\exists(f \mathrm{R} g))$
iff $s \models \exists(f \mathrm{R} g)$
iff $\exists p \in \mathit{Paths}(s) : p \models f \mathrm{R} g$
iff $\exists p \in \mathit{Paths}(s) : \forall i \geq 0 : (p[i] \models g \text{ or } \exists 0 \leq j < i : p[j] \models f)$
iff $\exists p \in \mathit{Paths}(s) : p[0] \models g \text{ and } \forall i \geq 1 : (p[i] \models g \text{ or } \exists 0 \leq j < i : p[j] \models f)$
iff $\exists p \in \mathit{Paths}(s) : p[0] \models g \text{ and } \forall i \geq 1 : (p[i] \models g \text{ or } p[0] \models f \text{ or } \exists 1 \leq j < i : p[j] \models f)$
iff $\exists p \in \mathit{Paths}(s) : p[0] \models g \text{ and } (p[0] \models f \text{ or } \forall i \geq 1 : (p[i] \models g \text{ or } \exists 1 \leq j < i : p[j] \models f))$
iff $\exists p \in \mathit{Paths}(s) : s \models g \text{ and } (s \models f \text{ or } \forall i \geq 0 : (p[i+1] \models g \text{ or } \exists 0 \leq j < i : p[j+1] \models f))$
iff $s \models g \text{ and } (s \models f \text{ or } \exists p \in \mathit{Paths}(s) : \forall i \geq 0 : (p[i+1] \models g \text{ or } \exists 0 \leq j < i : p[j+1] \models f))$
iff $s \models g \text{ and } (s \models f \text{ or } \exists s \to t : \exists p \in \mathit{Paths}(t) : \forall i \geq 0 : (p[i] \models g \text{ or } \exists 0 \leq j < i : p[j] \models f)$
iff $s \models g \text{ and } (s \models f \text{ or } \exists s \to t : t \models \exists(f \mathrm{R} g))$

Hence,

$$\mathit{Sat}(\exists(f \mathrm{R} g)) = \mathit{Sat}(g) \cap (\mathit{Sat}(f) \cup \{\, s \in S \mid \mathit{succ}(s) \cap \mathit{Sat}(\exists(f \mathrm{R} g)) \neq \emptyset \,\}).$$

Marking scheme: 2 marks for a correct derivation. 1 mark for a derivation that seems correct but lacks some detail (several fewer steps than in the sample solution). 0.5 mark for a derivation that contains several correct steps.

(b) Extend the algorithm to compute Sat with the case for $\exists(f \,\mathrm{R}\, g)$.

```
Sat(f):
switch (f) {
  case a :       return { s ∈ S | a ∈ ℓ(s) }
  case f ∧ g :   return Sat(f) ∩ Sat(g)
  ...
  case ∃(f R g) :
```

Answer:

$T = \emptyset$
`while` $T \neq F(T)$
  $T = F(T)$
`return` T

where

$$F(T) = Sat(g) \cap (Sat(f) \cup \{\, s \in S \mid succ(s) \cap T \neq \emptyset \,\}).$$

Marking scheme: 0.5 mark for a loop. 0.5 mark extra if the answer is correct.

(c) Provide the relevant definition(s) and theorem(s) that are needed to prove that your algorithm for the case $\exists(f \,\mathrm{R}\, g)$ is correct. (You do not have to prove the theorem(s).)

Answer: The function $F : 2^S \to 2^S$ is defined by

$$F(T) = Sat(g) \cap (Sat(f) \cup \{\, s \in S \mid succ(s) \cap T \neq \emptyset \,\}).$$

The function F is monotone, that is, for all $T, U \in 2^S$, if $T \subseteq U$ then $F(T) \subseteq F(U)$.

Marking scheme: 0.5 mark for the definition of F. 0.5 mark for the monotonicity of F. If F was already provided in part (b) and is not provided here, then 1 mark for the monotonicity of F.

Definitions

Definition 1. A labelled transition system is a tuple $\langle S, L, I, \to, \ell \rangle$ consisting of

- a set S of states,
- a set L of labels,
- a set $I \subseteq S$ of initial states,
- a transition relation $\to \subseteq S \times S$, and
- a labelling function $\ell : S \to 2^L$.

Definition 2. The set $succ(s)$ of successors of the state s is defined by

$$succ(s) = \{\, t \in S \mid s \to t \,\}.$$

Definition 3. Linear temporal logic (LTL) is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \,\mathrm{U}\, f$$

where $a \in L$.

We use the following syntactic sugar.

$$\begin{array}{rcll} f \vee g & = & \neg(\neg f \wedge \neg g) & \\ \mathrm{true} & = & a \vee \neg a & \\ \Diamond f & = & \mathrm{true}\,\mathrm{U}\, f & \text{(eventually } f\text{)} \\ \Box f & = & \neg\Diamond\neg f & \text{(always } f\text{)} \end{array}$$

Definition 4. $Paths(s)$ is the set of (execution) paths starting in state s. Let $p \in Paths(s)$ and $n \geq 0$. Then $p[n]$ is the $(n+1)^{\text{th}}$ state of the path p and $p[n..]$ is the suffix of p starting with the $(n+1)^{\text{th}}$ state.

Definition 5. The relation $\models$ is defined by

$$\begin{array}{rcl} p \models a & \text{iff} & a \in \ell(p[0]) \\ p \models f \wedge g & \text{iff} & p \models f \text{ and } p \models g \\ p \models \neg f & \text{iff} & \text{not}(p \models f) \\ p \models \bigcirc f & \text{iff} & p[1..] \models f \\ p \models f \,\mathrm{U}\, g & \text{iff} & \exists i \geq 0 : p[i..] \models g \text{ and } \forall 0 \leq j < i : p[j..] \models f \end{array}$$

and

$$\langle S, L, I, \to, \ell \rangle \models f \text{ iff } \forall s \in I : \forall p \in Paths(s) : p \models f$$

Definition 6. Computation tree logic (CTL) is defined as follows. The state formulas are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists g \mid \forall g$$

where $a \in L$. The path formulas are defined by

$$g ::= \bigcirc f \mid f \ \mathrm{U} \ f$$

Definition 7. The relation $\models$ is defined by

$$\begin{array}{rcl} s \models a & \text{iff} & a \in \ell(s) \\ s \models f \wedge g & \text{iff} & s \models f \text{ and } s \models g \\ s \models \neg f & \text{iff} & \mathrm{not}(s \models f) \\ s \models \exists g & \text{iff} & \exists p \in Paths(s) : p \models g \\ s \models \forall g & \text{iff} & \forall p \in Paths(s) : p \models g \end{array}$$

and

$$\begin{array}{rcl} p \models \bigcirc f & \text{iff} & p[1] \models f \\ p \models f \ \mathrm{U} \ g & \text{iff} & \exists i \geq 0 : p[i] \models g \text{ and } \forall 0 \leq j < i : p[j] \models f \end{array}$$

and

$$\langle S, L, I, \rightarrow, \ell \rangle \models f \text{ iff } \forall s \in I : s \models f$$

Definition 8. The satisfaction set $Sat(f)$ is defined by

$$Sat(f) = \{ \, s \in S \mid s \models f \, \}.$$

Definition 9. LTL/CTL formulas f and g are equivalent, denoted $f \equiv g$, if $\langle S, L, I, \rightarrow, \ell \rangle \models f$ iff $\langle S, L, I, \rightarrow, \ell \rangle \models g$ for all transition systems $\langle S, L, I, \rightarrow, \ell \rangle$.