

COMP424/524-06A Topics in Software Engineering

Part I – CTL Model Checking

10. Safety Properties

Robi Malik

THE UNIVERSITY OF
WAIKATO DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

Reachability

- A reachability property is a property stating that a particular state can be reached.
- For example:
“The process can enter the critical section.”
- Reachability properties can be viewed as the negation of a safety property.

$$EF \phi \equiv \neg AG \neg \phi$$

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 4

Safety Properties

A **safety property** is a property stating that
“something bad does never happen.”

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 2

Syntactic Characterisation

A temporal logic formula is a safety property if it can be written as

$$AG \phi \quad \text{in CTL or CTL}^*$$

or

$$G \phi \quad \text{in PLTL}$$

where ϕ is a **propositional** formula, i.e., a formula that does not contain any temporal combinators.

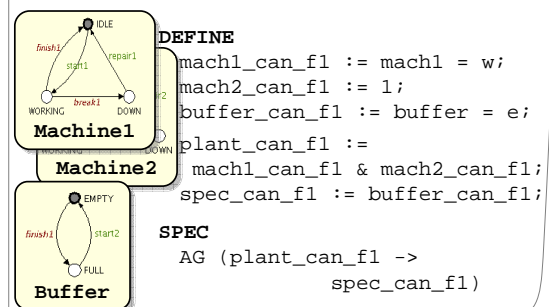
© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 5

Examples

- “The power plant will never blow up.”
- “The reactor temperature will never exceed 100°C.”
- “As long as the key is not in the ignition position, the car won’t start.”

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 3

Controllability is a Safety Property



© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 6

Combining Safety Properties

If ϕ and ψ are safety properties then

- $\phi \wedge \psi$ also is a safety property;

$$\mathbf{AG} \phi \wedge \mathbf{AG} \psi \equiv \mathbf{AG} (\phi \wedge \psi)$$
- $\phi \vee \psi$ is not necessarily a safety property.

$$\mathbf{AG} \phi \vee \mathbf{AG} \psi \neq \mathbf{AG} (\phi \vee \psi)$$

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 7

Example

“As long as the key is not in the ignition position, the car won’t start.”

$$\mathbf{AG} (\neg \text{start} \mathbf{W} \text{key})$$

$$\mathbf{AG} (\text{start} \Rightarrow \mathbf{F}^{-1} \text{key})$$

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 10

Alternative Characterisation

A temporal logic formula is a safety property if it can be written as

$$\mathbf{AG} \phi^-$$

where ϕ^- is a **past-time** temporal formula, i.e., a formula that contains only past-time temporal combinators.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 8

History Variables Method

$$\mathbf{AG} (\text{start} \Rightarrow \mathbf{F}^{-1} \text{key})$$

```

DEFINE
  was_key := key | p_was_key;
ASSIGN
  init(p_was_key) := 0;
  next(p_was_key) := was_key;
SPEC
  AG (start -> was_key)
  
```

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 11

Temporal Logic with Past

Past-time temporal combinators

- $\mathbf{X}^{-1} \phi$ – ϕ was true in the previous state;
- $\mathbf{F}^{-1} \phi$ – ϕ was true at some past state;
- $\mathbf{G}^{-1} \phi$ – ϕ was true in all past states;
- $\phi \mathbf{S} \psi$ – ψ was true at some past state, and after that up to the present state, ϕ was true;
 interpreted *on the tree of possible behaviours*.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 9

The General Case

Introduce one new propositional variable for each subformula having a past-time combinator at its root.

To translate, e.g.,

$$\mathbf{AG} (\text{start} \Rightarrow \mathbf{X}^{-1} (\neg \text{reset} \mathbf{S} \text{key}))$$

introduce symbols

- h_1 for $\neg \text{reset} \mathbf{S} \text{key}$;
- h_2 for $\mathbf{X}^{-1} h_1$.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 12

Observation

- When a safety property is violated, it is immediately possible to notice it.
- If a system fails to satisfy a safety property, then there exists a finite execution that reveals this fact.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 13

Real-Time Properties

```
VAR
count: -1..6;
ASSIGN
init(count) := -1;
next(count) := case
  start: -1;
  count = -1 & key: 0;
  count >= 0 & count < 6 & tick: count + 1;
  1: count;
esac;
SPEC
AG count <= 5
```

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 16

Not a Safety Property

“If the key is in the ignition position, the car will start eventually.”

Note:

This property cannot be refuted on any finite execution.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 14

Bounded Overtaking

- *“If process A requests access to the critical section before process B, then B is not granted access before A.”*
- *“If process A requests access to the critical section before process B, then B is granted access at most once before access is granted to A.”*

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 17

Real-Time Properties

“If the key is in the ignition position, the car will start within five seconds.”

“If the key is in the ignition position, and has been in the ignition position for five seconds, then the car must have started.”

Can be refuted on a finite execution.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 15

Summary

- Safety properties can be written as $AG \varphi$ for some propositional formula φ .
- Safety properties can be checked by exploring all reachable states of a system.
- Usually, they are the easiest properties to be checked by model checkers.

© THE UNIVERSITY OF WAIKATO • TE WHARE WANANGA O WAIKATO COMP424/524-06A I-10 Slide 18