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.

\[\text{EF } \varphi \equiv \neg\text{AG } \neg\varphi\]

Safety Properties

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

Syntactic Characterisation

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

\[
\begin{align*}
\text{AG } \varphi & \quad \text{in CTL or CTL}^* \\
\text{G } \varphi & \quad \text{in PLTL}
\end{align*}
\]

where \(\varphi\) is a propositional formula, i.e., a formula that does not contain any temporal combinators.

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.”

Controllability is a Safety Property

```
DEFINE
mach1_can_f1 := mach1 = w;
mach2_can_f1 := l;
buffer_can_f1 := buffer = e;
plant_can_f1 :=
mach1_can_f1 & mach2_can_f1;
spec_can_f1 := buffer_can_f1;

SPEC
AG (plant_can_f1 ->
    spec_can_f1)
```
Combining Safety Properties

If $\phi$ and $\psi$ are safety properties then

* $\phi \land \psi$ also is a safety property;
  \[ \text{AG } \phi \land \text{AG } \psi \equiv \text{AG } (\phi \land \psi) \]
* $\phi \lor \psi$ is not necessarily a safety property.
  \[ \text{AG } \phi \lor \text{AG } \psi \not\equiv \text{AG } (\phi \lor \psi) \]

Example

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

\[ \text{AG } (\neg \text{start } W \text{ key}) \]
\[ \text{AG } (\text{start } \Rightarrow \text{F}^{-1} \text{ key}) \]

Alternative Characterisation

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

\[ \text{AG } \phi^- \]

where $\phi^-$ is a past-time temporal formula, i.e., a formula that contains only past-time temporal combinators.

History Variables Method

\[ \text{AG } (\text{start } \Rightarrow \text{F}^{-1} \text{ key}) \]

\[ \text{DEFINE} \]
\[ \text{was}_\text{key} := \text{key} | \text{p}_\text{was}_\text{key}; \]
\[ \text{ASSIGN} \]
\[ \text{init} (\text{p}_\text{was}_\text{key}) := 0; \]
\[ \text{next} (\text{p}_\text{was}_\text{key}) := \text{was}_\text{key}; \]
\[ \text{SPEC} \]
\[ \text{AG } (\text{start } \Rightarrow \text{was}_\text{key}) \]

Temporal Logic with Past

Past-time temporal combinators

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

The General Case

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

To translate, e.g.,

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

introduce symbols

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

Real-Time Properties

```plaintext
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:
  esac;
SPEC
  AG count <= 5
```

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.

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.

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."

Summary

- Safety properties can be written as $\text{AG } \phi$ for some propositional formula $\phi$.
- 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.