Systems and Software Verification

Chapter 7. Safety Properties

Lecturer: JUNBEOM YOO
jbyoo@konkuk.ac.kr
http://dslab.konkuk.ac.kr

Ver. 2.0

7. Safety Properties

- Safety property
  - Under certain conditions, an (undesirable) event never occur.
  - Examples:
    - (S1) “ Both processes will never be in their critical sections simultaneously (mutual exclusion) ”
    - (S2) “ Memory overflow will never occur ”
    - (S3) “ The situation … is impossible “
    - (S4) “ As long as the key is not in the ignition position, the car won’t start “ ← with conditions

    - ¬ safety property = reachability property
    - ¬ reachability property = safety property

- Organization of Chapter 7
  - Safety Properties in Temporal Logic
  - A Formal Definition
  - Safety Properties in Practice
  - The history Variables Method

7.1 Safety Properties in Temporal Logic

- AG Φ
  - " Φ never occurs. "

  - (S1) " Both processes will never be in their critical sections simultaneously "
    - AG $\neg(\mathrm{crit\_sec}_1 \wedge \mathrm{crit\_sec}_2)$
  - (S2) " Memory overflow will never occur "
    - AG $\neg$overflow
  - (S3) " The situation … is impossible "
    - AG $\neg$situation
  - (S4) " As long as the key is not in the ignition position, the car won't start "
    - A ($\neg$start W key) (using weak until)
    - A ($\neg$start U key) ← Not a safety property !

7.2 A Formal Definition

- Syntactic characterization
  - Safety properties can be written in the form AG Φ^{-}
    - Φ^{-} is a past temporal formula
  - When a safety property is violated, it should be possible to instantly notice it.
  - We can only notice it, in the current state, relying on events which occurred earlier.

- Temporal logic with past
  - CTL* does not provide past combinators
  - But, we can use a mirror image of future combinators (F^{-1}, X^{-1})

- AG Φ^{-} in practice
  - (S1) AG $\neg(\text{crit\_sec}_1 \wedge \text{crit\_sec}_2)$
    - $\neg(\text{crit\_sec}_1 \wedge \text{crit\_sec}_2)$ is a ϕ^{-}
  - (S4) A $\neg\text{start}$ W key
    - Can be rewritten in the form: AG $(\text{start} \Rightarrow F^{-1}\ \text{key})$
    - " It is always true (AG) that if the car starts, then (⇒) the key was inserted beforehand (F^{-1}). "
  - If Ψ_1 and ψ_2 are safety properties, then $\Psi_1 \wedge \psi_2$ again a safety property.
    - But, $\Psi_1 \vee \psi_2$ is in general not

- Safety properties and diagnostic
  - If AG Φ^{-} is not satisfied, then there necessarily exists a finite path leading from *init* to it.
  - Since Φ^{-} is a past formula.

7.3 Safety Properties in Practice

- Safety properties are verified simply by submitting it to a model checker.
- But, in real life, hurdles spring up.

- A simple case: non-reachability
  - The most safety properties
  - $\neg \mathrm{EF}\ (\mathrm{crit\_in}_1 \wedge \mathrm{crit\_in}_2) = \mathrm{AG}\ \Phi^-$
    - $\neg(\mathrm{crit\_in}_1 \wedge \mathrm{crit\_in}_2)$ is a present formula

- Safety without past
  - $\mathrm{A}\ (\neg \mathrm{start}\ \mathrm{W}\ \mathrm{key})$ is used more often than $\mathrm{AG}\ (\mathrm{start} \Rightarrow \mathrm{F}^{-1}\ \mathrm{key})$
  - But, no model checker is able to deal with past formulas. So, mixed logics are used.
  - The problem is their identification.
    → If they are identified, then it can be dealt with similarly
    → Otherwise, we have to use the method of history variables (in section 7.4)

- Safety with explicit past
  - No model checker is able to handle temporal formula with past.
  - Two approaches:
    1. Eliminate the past (in principle, it is possible to translate mixed formulas to pure-future ones)
       - $\mathrm{AG}\ (\phi \Rightarrow \mathrm{F}^{-1}\ \psi) \equiv \mathrm{A}\ (\neg\phi\ \mathrm{W}\ \psi)$, but not easy.
    2. History variable method (section 7.4)

7.4 The History Variables Method

- Skipped !!!