Unit-3: Linear-time properties

B. Srivathsan

Chennai Mathematical Institute

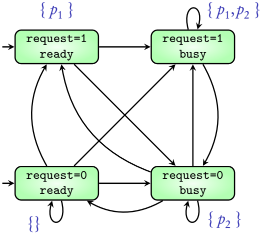
NPTEL-course

July - November 2015

Module 4:
Safety properties

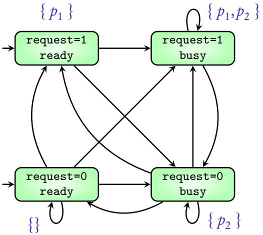
Atomic propositions AP $= \{ p_1, p_2 \}$

p_1: `request=1` p_2: `status=busy`



Atomic propositions AP $= \{ p_1, p_2 \}$

p_1: `request=1` $\qquad$ p_2: `status=busy`



AP-INF = set of **infinite words** over *PowerSet*(**AP**)

Property: Always: if p_1 is true, then in the **next step** p_2 is true

$$\{ A_0 A_1 A_2 \cdots \in \text{AP-INF} \mid \text{ if } A_i \text{ contains } p_1, \text{ then } A_{i+1} \text{ contains } p_2 \}$$

$\{ p_1 \} \{ p_2 \} \{ p_1 \} \{ p_1, p_2 \} \{ p_2 \} \{ p_1 \} \{ p_1, p_2 \} \dots$

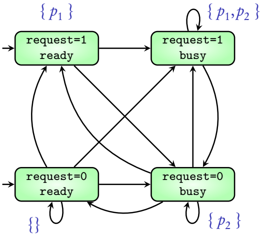
$\{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \dots$

$\{ \} \{ \} \{ \} \{ \} \{ \} \{ \} \dots$

$\vdots$

Atomic propositions AP $= \{ p_1, p_2 \}$

p_1: `request=1` $\quad$ p_2: `status=busy`



AP-INF = set of **infinite words** over *PowerSet*(**AP**)

Property: Always: if p_1 is true, then in the **next step** p_2 is true

$\{ A_0 A_1 A_2 \cdots \in \text{AP-INF} \mid \text{ if } A_i \text{ contains } p_1, \text{ then } A_{i+1} \text{ contains } p_2 \}$

$\{ p_1 \} \{ p_2 \} \{ p_1 \} \{ p_1, p_2 \} \{ p_2 \} \{ p_1 \} \{ p_1, p_2 \} \dots$

$\{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \dots$

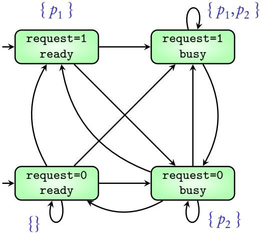
$\{ \} \{ \} \{ \} \{ \} \{ \} \{ \} \dots$

$\vdots$

Property is written as $\mathbf{G} \; (\, p_1 \longrightarrow X p_2 \,)$

Atomic propositions AP $= \{ p_1, p_2 \}$

p_1: `request=1` p_2: `status=busy`



AP-INF = set of **infinite words** over *PowerSet*(**AP**)

Property: Always: if p_1 is true, then in the **next step** p_2 is true

$$\{ A_0 A_1 A_2 \cdots \in \text{AP-INF} \mid \text{ if } A_i \text{ contains } p_1 \text{, then } A_{i+1} \text{ contains } p_2 \}$$

$\{ p_1 \} \{ p_2 \} \{ p_1 \} \{ p_1, p_2 \} \{ p_2 \} \{ p_1 \} \{ p_1, p_2 \} \ldots$

$\{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \{ p_2 \} \ldots$

$\{ \} \{ \} \{ \} \{ \} \{ \} \{ \} \ldots$

$\vdots$

Property is written as $\mathbf{G} \, (p_1 \longrightarrow X p_2)$

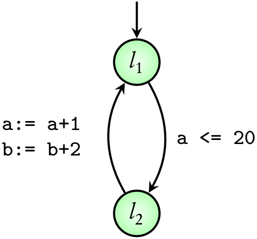
Above TS **satisfies** this property

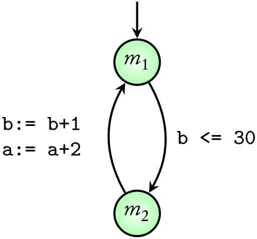
X operator

- $G\,(p_1 \longrightarrow XXp_2)$:
  - Always: if p_1 is true then in the next to next step p_2 is true
- $F\,(p_1 \wedge X\neg p_1)$:
  - Somewhere: p_1 is true and in the next step it becomes false
- $G\,(\,Xp_2 \longrightarrow p_1)$:
  - Always: if p_2 is true then in the previous step p_1 is true

```
while a <= 20
      a := a+1
      b := b+2
```

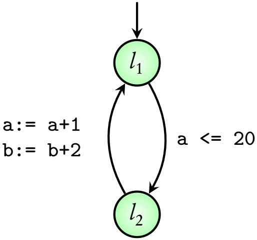
```
while b <= 30
      b:=b+1
      a:=a+2
```

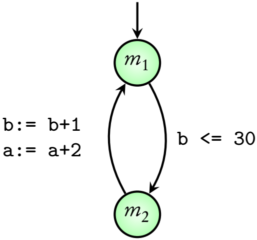




```
while a <= 20
      a := a+1
      b := b+2
```

```
while b <= 30
      b:=b+1
      a:=a+2
```

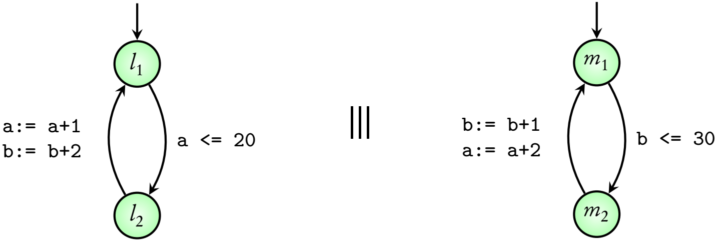




Check: Whenever $a \geq 10$, in the next to next step $b \geq 12$

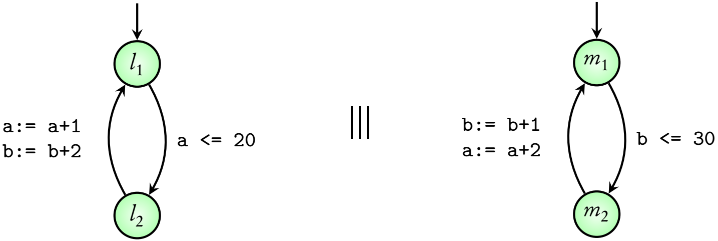
Atomic propositions AP $= \{ p_1, p_2 \}$

p_1 : `a >= 10` p_2 : `b >= 12`



Atomic propositions AP $= \{\, p_1, p_2 \,\}$

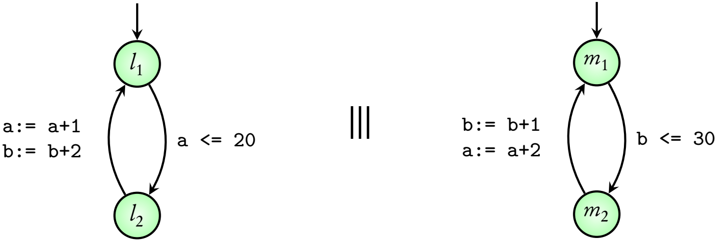
p_1 : `a >= 10` p_2 : `b >= 12`



Check: G $(p_1 \longrightarrow XXp_2\ )$

Atomic propositions AP $= \{ p_1, p_2 \}$

p_1 : `a >= 10` p_2 : `b >= 12`

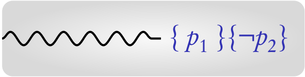


Check: $G\,(p_1 \longrightarrow XXp_2\ )$

NuSMV demo

Coming next: idea of safety properties

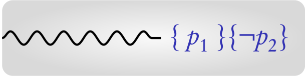
Property 1: if p_1 is true, then p_2 should be true in the next step



$\{p_1\}\{\neg p_2\}$

"something bad"

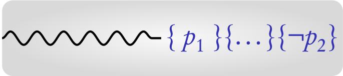
Property 1: if p_1 is true, then p_2 should be true in the next step



$\{p_1\}\{\neg p_2\}$

"something bad"

Property 2: if p_1 is true, then p_2 should be true in the next to next step



$\{p_1\}\{\dots\}\{\neg p_2\}$

"something bad"

Property 1: if p_1 is true, then p_2 should be true in the next step

$\rightsquigarrow \{ p_1 \}\{\neg p_2\}$ “something bad”

Property contains all words where **something bad** is absent

Property 2: if p_1 is true, then p_2 should be true in the next to next step

$\rightsquigarrow \{ p_1 \}\{\dots\}\{\neg p_2\}$ “something bad”

Safety properties

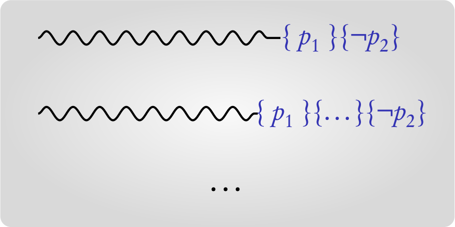
AP-INF = set of **infinite words** over *PowerSet*(**AP**)

P: a property over AP

Safety properties

AP-INF = set of **infinite words** over *PowerSet*(**AP**)

P: a property over AP



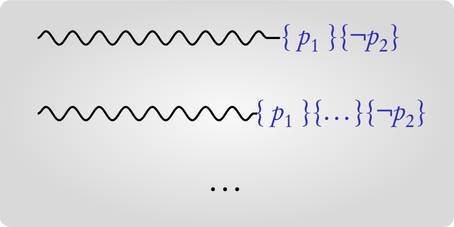
Bad-Prefixes

P is a safety property if there **exists** a set Bad-Prefixes such that

Safety properties

AP-INF = set of **infinite words** over *PowerSet*(**AP**)

P: a property over AP



$\{p_1\}\{\neg p_2\}$

$\{p_1\}\{\ldots\}\{\neg p_2\}$

$\ldots$

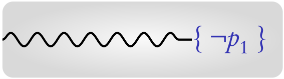
Bad-Prefixes

P is a safety property if there **exists** a set Bad-Prefixes such that

P is the set of **all words** that **do not start** with a Bad-Prefix

Invariants are **special cases** of safety properties

Property: Always p_1 is true



$\{ \neg p_1 \}$

“Bad-Prefixes”

Safety properties

Avoiding bad prefixes

X operator