Unit-3: Linear-time properties

B. Srivathsan

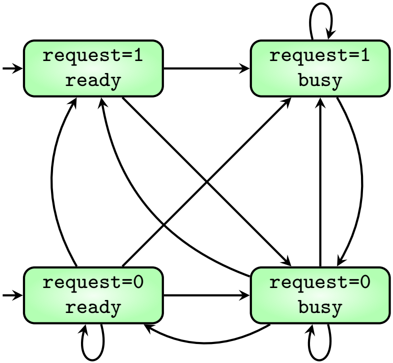
Chennai Mathematical Institute

NPTEL-course

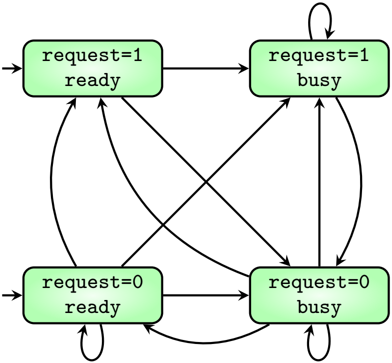
July - November 2015

Module 2:
What is a “property”?

Goal: Attach a **mathematical meaning** to “property”

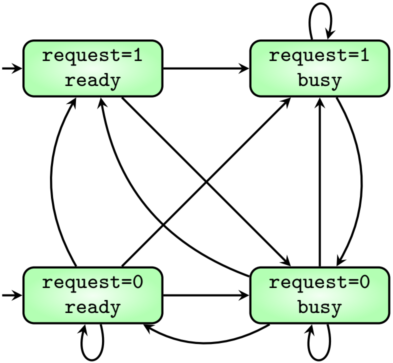



```
MODULE main
VAR
   request:  boolean;
   status:  {ready, busy}
ASSIGN
   init(status) := ready;
   next(status) := case
                request :  busy;
                TRUE : {ready,busy};
                    esac;
```




```
MODULE main
VAR
    request:  boolean;
    status:  {ready, busy}
ASSIGN
    init(status) := ready;
    next(status) := case
                        request :  busy;
                        TRUE : {ready,busy};
                              esac;
```

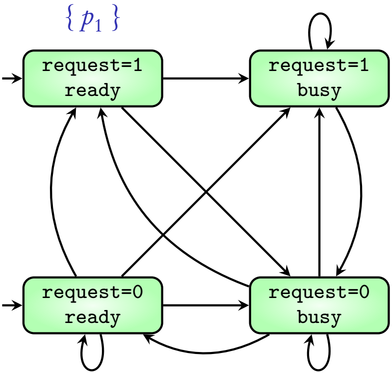
p_1: (request=1) p_2: (status=busy)




```
MODULE main
VAR
    request:  boolean;
    status:  {ready, busy}
ASSIGN
    init(status) := ready;
    next(status) := case
                    request :  busy;
                    TRUE : {ready,busy};
                         esac;
```

Atomic propositions

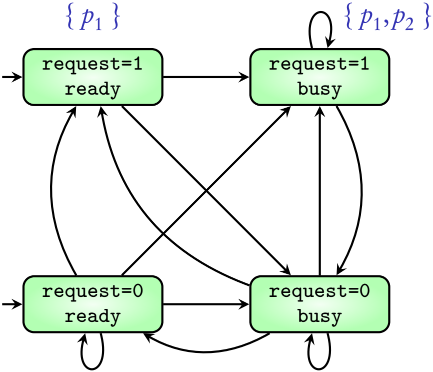
p_1: (request=1) p_2: (status=busy)




```
MODULE main
VAR
    request:  boolean;
    status:  {ready, busy}
ASSIGN
    init(status) := ready;
    next(status) := case
                        request :  busy;
                        TRUE : {ready,busy};
                              esac;
```

Atomic propositions

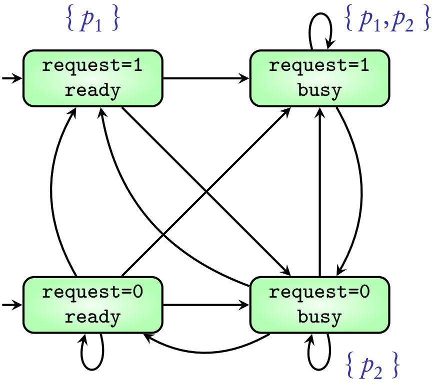
p_1: (request=1) p_2: (status=busy)




```
MODULE main
VAR
    request:  boolean;
    status:  {ready, busy}
ASSIGN
    init(status) := ready;
    next(status) := case
                    request :  busy;
                    TRUE : {ready,busy};
                          esac;
```

Atomic propositions

p_1: (request=1) p_2: (status=busy)

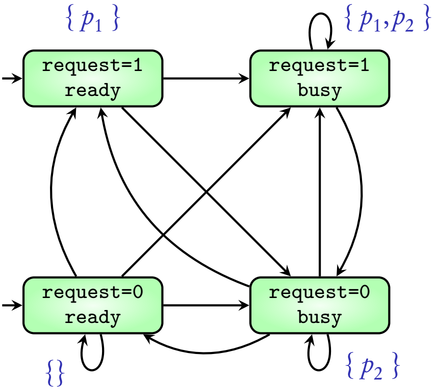



```
MODULE main
VAR
    request:  boolean;
    status:  {ready, busy}
ASSIGN
    init(status) := ready;
    next(status) := case
                       request :  busy;
                       TRUE : {ready,busy};
                             esac;
```

Atomic propositions

p_1: (request=1)

p_2: (status=busy)

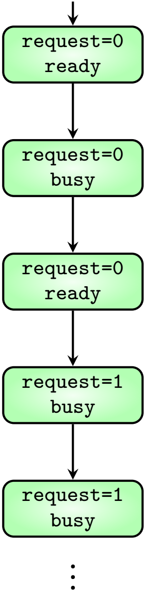



```
MODULE main
VAR
    request:  boolean;
    status:  {ready, busy}
ASSIGN
    init(status) := ready;
    next(status) := case
                    request :  busy;
                    TRUE : {ready,busy};
                         esac;
```

Atomic propositions

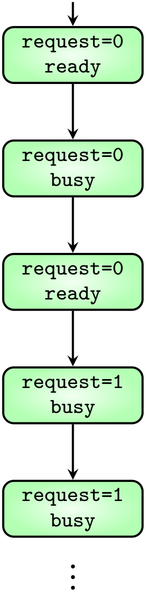
p_1: (request=1) p_2: (status=busy)

Execution



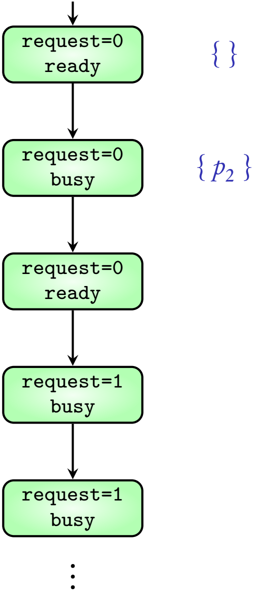
request=0
ready
request=0
busy
request=0
ready
request=1
busy
request=1
busy

Execution

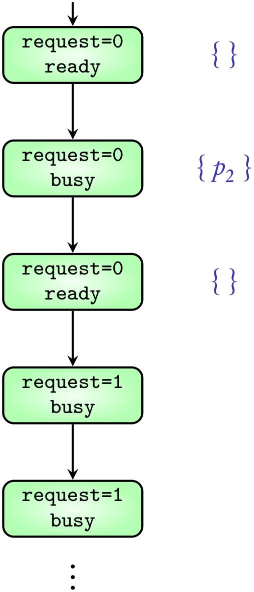

request=0
ready
request=0
busy
request=0
ready
request=1
busy
request=1
busy

{ }

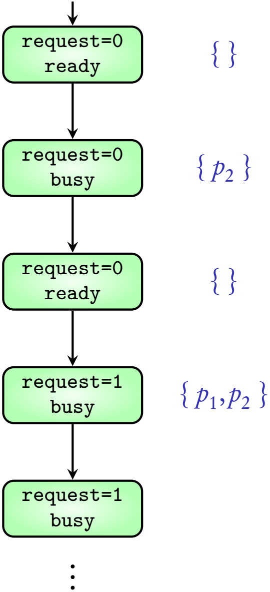
Execution


request=0
ready
{ }
request=0
busy
{ p_2 }
request=0
ready
request=1
busy
request=1
busy

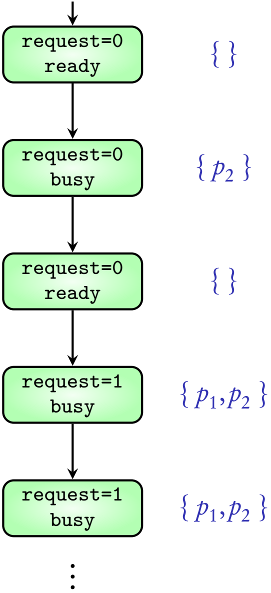
Execution


request=0
ready
{ }
request=0
busy
{ p_2 }
request=0
ready
{ }
request=1
busy
request=1
busy

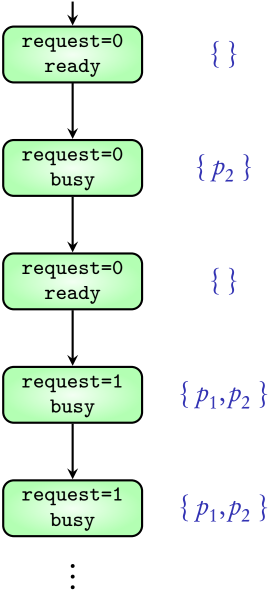
Execution

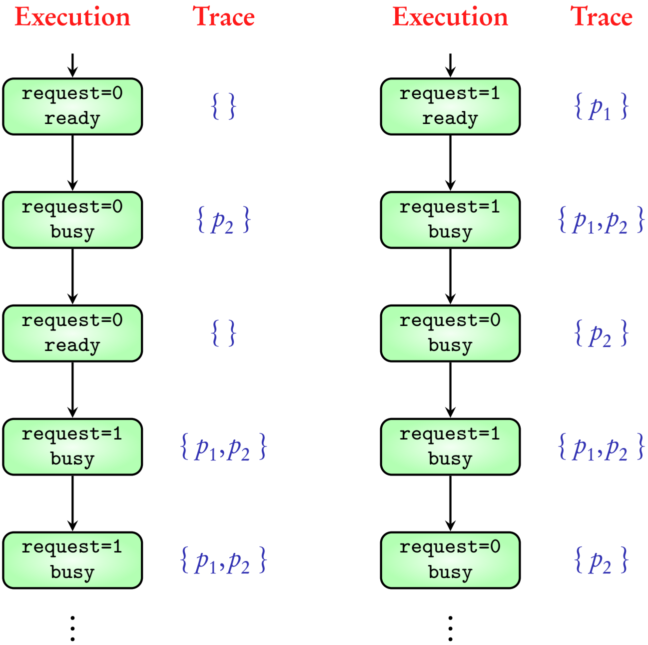

request=0
ready
{ }
request=0
busy
{ p_2 }
request=0
ready
{ }
request=1
busy
{ p_1, p_2 }
request=1
busy

Execution


request=0
ready
$\{\ \}$
request=0
busy
$\{\ p_2\ \}$
request=0
ready
$\{\ \}$
request=1
busy
$\{\ p_1, p_2\ \}$
request=1
busy
$\{\ p_1, p_2\ \}$

Execution **Trace**




Execution
Trace
request=0 ready
{ }
request=0 busy
{ p_2 }
request=0 ready
{ }
request=1 busy
{ p_1,p_2 }
request=1 busy
{ p_1,p_2 }
Execution
Trace
request=1 ready
{ p_1 }
request=1 busy
{ p_1,p_2 }
request=0 busy
{ p_2 }
request=1 busy
{ p_1,p_2 }
request=0 busy
{ p_2 }

$$\mathbf{AP} = \{ p_1, p_2, \dots, p_k \}$$

$$\begin{aligned}
\mathbf{AP} &= \{\ p_1, p_2,\ \cdots, p_k\ \} \\
\mathit{PowerSet}(\mathbf{AP}) &= \{\ \{\ \},\ \{p_1\},\ \ldots,\ \{p_k\}, \\
&\qquad \{\ p_1, p_2\ \},\ \{\ p_1, p_3\ \},\ \ldots,\ \{\ p_{k-1}, p_k\ \}, \\
&\qquad \cdots \\
&\qquad \{\ p_1, p_2,\ \ldots,\ p_k\ \}\ \}
\end{aligned}$$

$$
\begin{aligned}
\mathbf{AP} &= \{\ p_1, p_2,\ \dots\ , p_k\ \} \\
\mathit{PowerSet}(\mathbf{AP}) &= \{\ \{\ \},\ \{p_1\},\ \dots\ ,\ \{p_k\}, \\
&\qquad \{\ p_1, p_2\ \},\ \{\ p_1, p_3\ \},\ \dots\ ,\ \{\ p_{k-1}, p_k\ \}, \\
&\qquad\qquad \dots \\
&\qquad \{\ p_1, p_2,\ \dots,\ p_k\ \}\ \}
\end{aligned}
$$

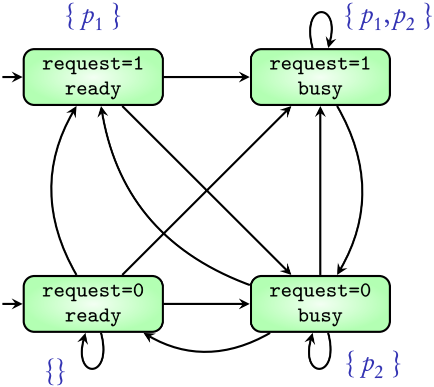
Trace(Execution) is an **infinite word** over *PowerSet*(**AP**)

$$\mathbf{AP} = \{\, p_1, p_2, \ldots, p_k \,\}$$

$$\begin{aligned} \mathit{PowerSet}(\mathbf{AP}) = \{ & \{\,\}, \{p_1\}, \ldots, \{p_k\}, \\ & \{\, p_1, p_2 \,\}, \{\, p_1, p_3 \,\}, \ldots, \{\, p_{k-1}, p_k \,\}, \\ & \ldots \\ & \{\, p_1, p_2, \ldots, p_k \,\} \,\} \end{aligned}$$

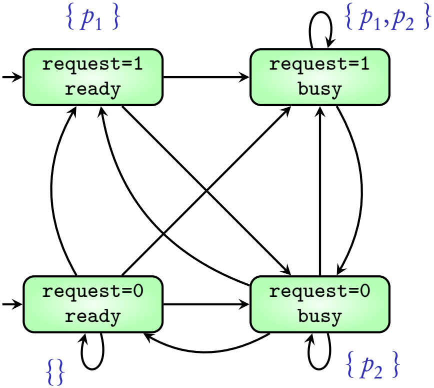
Trace(Execution) is an **infinite word** over *PowerSet*(**AP**)

Traces(TS) is the $\{$ Trace(σ) $\mid$ σ is an execution of the TS $\}$



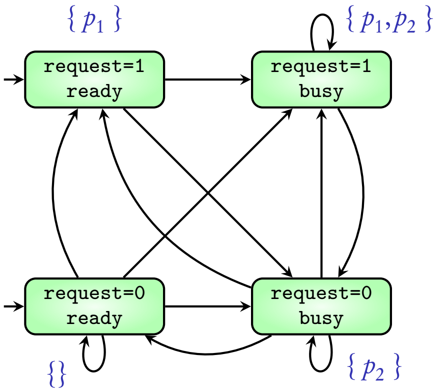
{ p_1 }
{ p_1,p_2 }
request=1
ready
request=1
busy
request=0
ready
request=0
busy
{}
{ p_2 }

Traces:



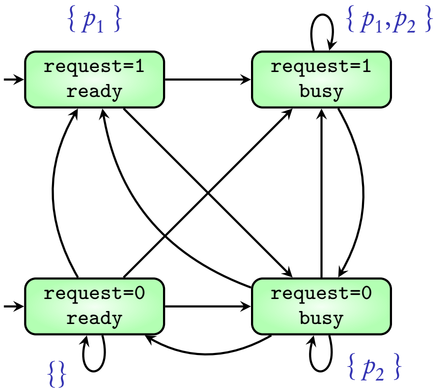
{ p_1 }
{ p_1,p_2 }
request=1
ready
request=1
busy
request=0
ready
request=0
busy
{}
{ p_2 }

Traces: { } { } { } { } { } { } { } { } …



Traces: { } { } { } { } { } { } { } { } ...

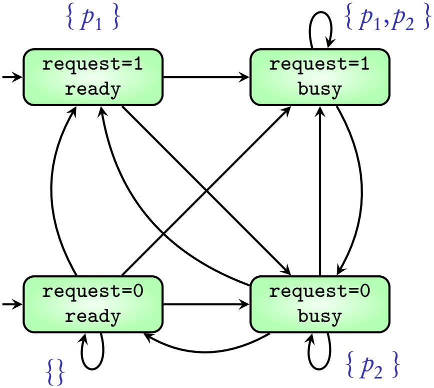
{ } { p_2 } { p_2 } { p_2 } { p_2 } { p_2 } { p_2 } ...



Traces: $\{\,\}\,\{\,\}\,\{\,\}\,\{\,\}\,\{\,\}\,\{\,\}\,\{\,\}\,\{\,\}\ldots$

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

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

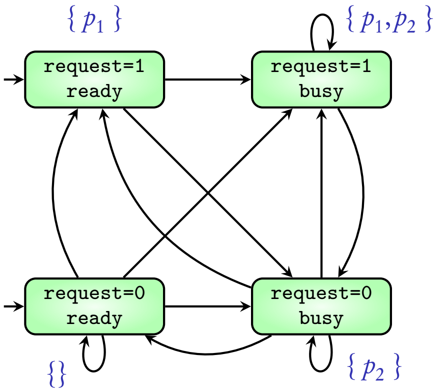


Traces: $\{\,\}\{\,\}\{\,\}\{\,\}\{\,\}\{\,\}\{\,\}\{\,\}\ldots$

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

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

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



Traces:

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

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

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

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

$\vdots$

Traces of a TS describe its **behaviour** with respect to the atomic propositions

Behaviour of TS

Atomic propositions

Set of its **traces**

Coming next: What is a **property**?

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

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

Property 1: p_1 is always true

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

Property 1: p_1 is always true

$$\{ A_0 A_1 A_2 \cdots \in \text{AP-INF} \mid \textbf{each } A_i \text{ contains } p_1 \}$$

$\{p_1\}\{p_1\}\{p_1\}\{p_1\}\{p_1\}\{p_1\}\{p_1\}\ldots$

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

$\vdots$

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

Property 1: p_1 is always true

$$\{\, A_0 A_1 A_2 \cdots \in \text{AP-INF} \mid \textbf{ each } A_i \text{ contains } p_1 \,\}$$

$\{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \dots$

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

$\vdots$

Property 2: p_1 is true at least once and p_2 is always true

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

Property 1: p_1 is always true

$$\{ A_0 A_1 A_2 \dots \in \text{AP-INF} \mid \textbf{each } A_i \text{ contains } p_1 \}$$

$\{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \{ p_1 \} \dots$

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

$\vdots$

Property 2: p_1 is true at least once and p_2 is always true

$$\{ A_0 A_1 A_2 \dots \in \text{AP-INF} \mid \textbf{exists } A_i \text{ containing } p_1 \text{ and } \textbf{every } A_j \text{ contains } p_2 \}$$

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

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

$\vdots$

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

A property over AP is a **subset** of AP-INF

Behaviour of TS

Atomic propositions

Set of its **traces**

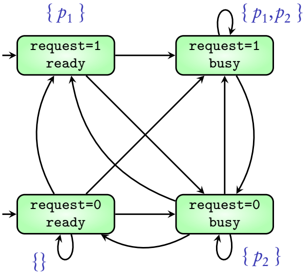
Property over AP

Subset of AP-INF

When does a transition system **satisfy** a property?

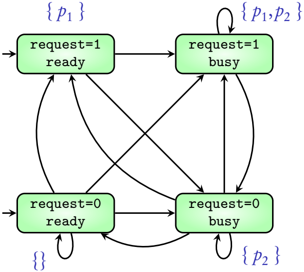
$$\mathrm{AP} = \{\ p_1,\ p_2\ \ \}$$

Transition System



$$\mathrm{AP} = \{\ p_1,\ p_2\ \}$$

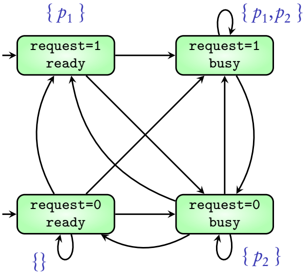
Transition System



Property

$$\mathrm{AP} = \{\ p_1,\ p_2\ \ \}$$

Transition System

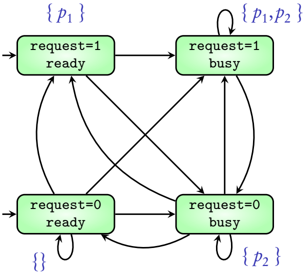


Property

$$\mathbf{G}\ p_1$$

$\mathrm{AP} = \{\ p_1,\ p_2\ \}$

Transition System



Property

$\mathbf{G}\ p_1$

Transition system *TS* satisfies property *P* if

Traces$(TS) \subseteq P$

A property over AP is a subset of AP-INF

A property over AP is a subset of AP-INF

→ hence also called **Linear-time property**

Behaviour of TS

Atomic propositions

Set of its **traces**

Property over AP

Subset of AP-INF

When does system satisfy property?