

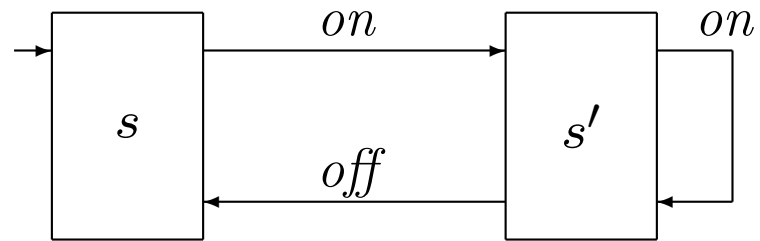
# Labelled Transition Systems and Invariants

## Labelled Transition Systems

A labelled transition system (Lts)  $B$  consists of

- a set  $states(B)$
- a nonempty set  $start(B) \subseteq states(B)$
- a set  $acts(B)$
- a set  $steps(B) \subseteq states(B) \times acts(B) \times states(B)$ ;  
we write  $s \xrightarrow{a}_B s' \triangleq (s, a, s') \in steps(B)$

## Switch Example



LTS  $S$ :

- $states(S) = \{s, s'\}$
- $start(S) = \{s\}$
- $acts(S) = \{on, off\}$ ,
- $steps(S) = \{(s, on, s'), (s', on, s'), (s', off, s)\}$

## IOA Spec of Switch (precondition/effect style)

automaton Switch

signature

input on

output off

states

burning: Bool := false

transitions

input on

eff burning := true

output off

pre burning = true

eff burning := false

## IOA Spec of Switch (predicative style)

Transitions can be specified in predicate logic using **primed** and **un-primed** variables.

Unprimed variables refer to value before transition, primed variables refer to value after transition.

Each IOA spec can be translated into predicative style.

## IOA Spec of Switch (predicative style)

automaton Switch

signature

input on

output off

states

burning: Bool

initially ~burning

transitions

input on

eff burning := choose

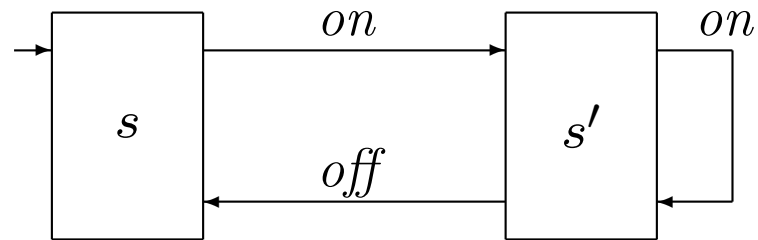
ensuring burning' = true

output off

eff burning := choose

ensuring burning /\ ~burning'

## The Behavior of LTSs



### Execution fragments:

$s$   
 $s' \text{ on } s' \text{ off } s$   
 $s \text{ on } s' \text{ off } s \text{ on } s' \text{ off } \dots$   
 $s' \text{ on } s' \text{ on } s' \text{ on } s' \text{ on } \dots$

### Executions:

$s$   
 $s \text{ on } s' \text{ off } s \text{ on } s' \text{ off } \dots$

**Notation:**  $f\text{fragm}(B)$ ,  $\text{fragm}(B)$ ,  $f\text{execs}(B)$ ,  $\text{execs}(B)$

## The Behavior of LTSs (cnt)

### Concatenation

Partial operation on  $fragm(B)$ ;  $\sigma \cdot \rho$  defined iff  $\sigma$  is finite and last state  $\sigma$  is first state  $\rho$

### Prefix Ordering

$$\sigma \leq \rho \triangleq \sigma = \rho \vee \exists \tau : \sigma \cdot \tau = \rho$$

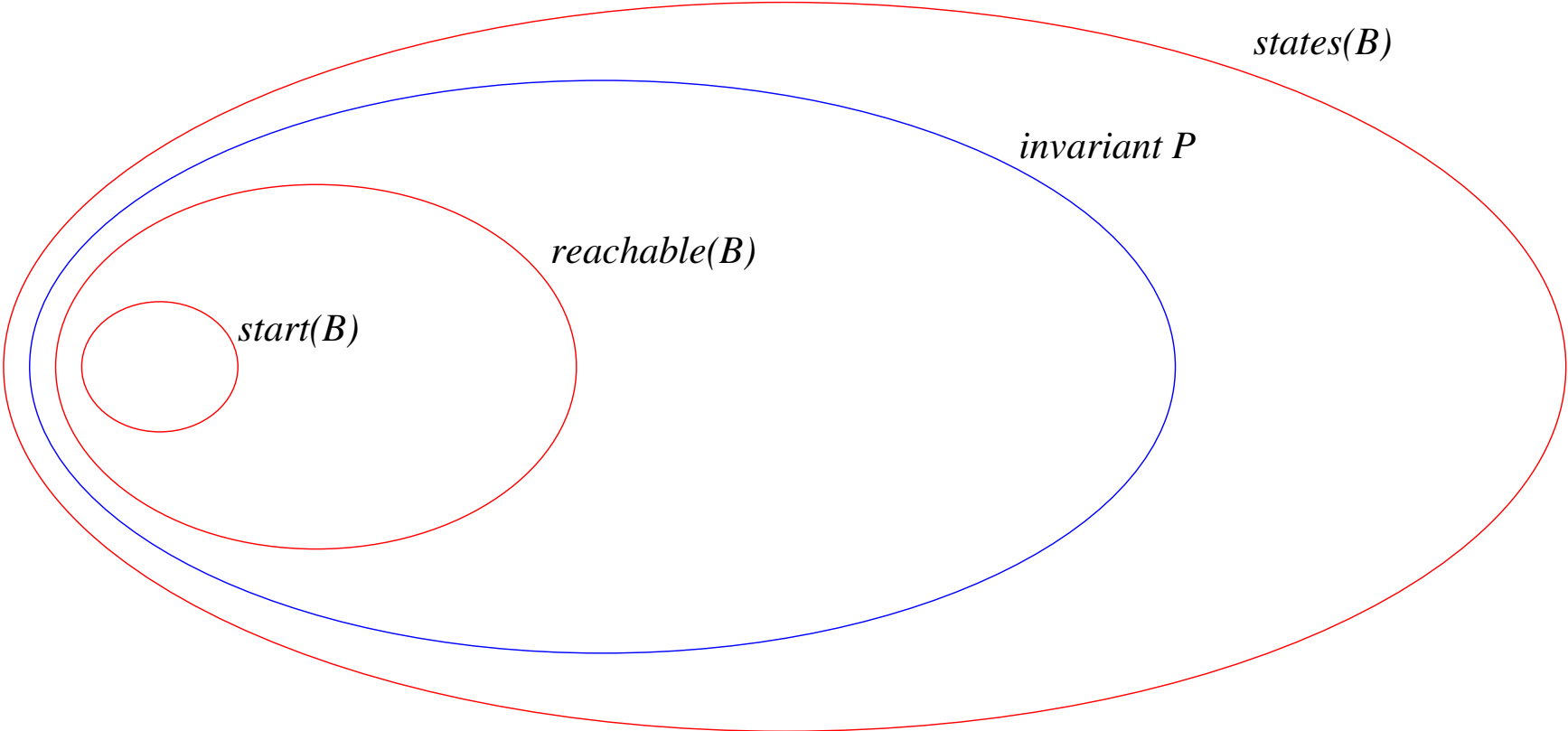
## Invariants

A state is **reachable** if it is the last state of a finite execution

Let  $P$  and  $Q$  be sets of states of  $B$ . Then:

- $P$  is **invariant** in  $B$  if  $P$  contains all reachable states
- $P$  is **stable** in  $B$  if  $s \in P \wedge s \xrightarrow{a}_B s' \Rightarrow s' \in P$
- $P$  is **inductive** in  $B$  if  $start(B) \subseteq P$  and  $P$  is stable

# Invariants (cnt)



## Invariants (cnt)

### Fact 1

If  $P$  is inductive then  $P$  is invariant

### Fact 2

If  $P$  is invariant and  $P \subseteq Q$  then  $Q$  is invariant

### Fact 3

If  $P$  and  $Q$  are invariant (stable, inductive) then  $P \cap Q$  is invariant (stable, inductive)

## Invariants (cnt)

$P$  is stable relative to  $Q$  if  $s \in P \cap Q \wedge s \xrightarrow{a}_B s' \wedge s' \in Q \Rightarrow s' \in P$

$P$  is inductive relative to  $Q$  if  $start(B) \subseteq P \wedge P$  is stable relative to  $Q$

### Fact 4

If  $Q$  is invariant and  $P$  is inductive relative to  $Q$  then  $P$  is invariant

### Fact 5

If  $Q$  is stable (inductive) and  $P$  is stable (inductive) relative to  $Q$ , then  $P \cap Q$  is stable (inductive)

## Invariants for Adder

automaton Adder

signature

input add(i, j: Int)

output result(k: Int)

states

value: Int := 0,

ready: Bool := false

transitions

input add(i, j)

eff value := i + j; ready := true

output result(k)

pre k = value /\ ready

eff value := 0; ready := false

invariant of Adder:  $\sim\text{ready} \Rightarrow \text{value} = 0$

## Invariants for Square Root Module

```
automaton SquareRoot
  signature
    internal compute    output halt
  states
    x: Nat, u: Nat:=1, w: Nat:=1, z: Nat:=0, done : Bool:=false
  transitions
    internal compute
      pre ~done /\ w <= x
      eff z := z+1; u := u+2; w := w+u
    output halt
      pre ~done /\ w > x
      eff done := true
invariant of SquareRoot:
  done => (z*z) <= x /\ x < ((z+1)*(z+1))
```

**Proof** First translate spec into predicative style:

$$\begin{aligned}\varphi_{init} &\triangleq u = 1 \wedge w = 1 \wedge z = 0 \wedge \neg done \\ \varphi_{compute} &\triangleq \neg done \wedge w \leq x \wedge x' = x \wedge \neg done' \\ &\quad \wedge z' = z + 1 \wedge w' = w + u' \wedge u' = u + 2 \\ \varphi_{halt} &\triangleq \neg done \wedge w > x \wedge done' \\ &\quad \wedge z' = z \wedge w' = w \wedge u' = u \wedge x' = x\end{aligned}$$

Write  $\varphi'$  for formula obtained by replacing in  $\varphi$  each variable  $x$  by  $x'$ .

## Proof (cnt)

Claim 1.  $\varphi_1 \triangleq u = 2 * z + 1$  is inductive

Proof We establish

1.  $\varphi_{init} \Rightarrow \varphi_1$
2.  $\varphi_1 \wedge \varphi_{compute} \Rightarrow \varphi'_1$
3.  $\varphi_1 \wedge \varphi_{halt} \Rightarrow \varphi'_1$

Ad 1. Assume  $\varphi_{init}$ . Then

$$u = 1 = 2 * 0 + 1 = 2 * z + 1$$

Ad 2. Assume  $\varphi_1 \wedge \varphi_{compute}$ . Then

$$u' = u + 2 = (2 * z + 1) + 2 = 2 * (z + 1) + 1 = 2 * z' + 1$$

Ad 3. Assume  $\varphi_1 \wedge \varphi_{halt}$ . Then

$$u' = u = 2 * z + 1 = 2 * z' + 1$$

QED

## Proof (cnt)

Claim 2.  $\varphi_2 \triangleq w = (z + 1)^2$  is inductive relative to  $\varphi_1$

Proof We establish

1.  $\varphi_{init} \Rightarrow \varphi_2$
2.  $\varphi_1 \wedge \varphi_2 \wedge \varphi_{compute} \Rightarrow \varphi'_2$
3.  $\varphi_1 \wedge \varphi_2 \wedge \varphi_{halt} \Rightarrow \varphi'_2$

Ad 1. Assume  $\varphi_{init}$ . Then  
 $w = 1 = (0 + 1)^2 = (z + 1)^2$

Ad 2. Assume  $\varphi_1 \wedge \varphi_2 \wedge \varphi_{compute}$ . Then  
 $w' = w + u + 2 = (z + 1)^2 + 2 * z + 1 + 2 = (z + 2)^2 = (z' + 1)^2$

Ad 3. Assume  $\varphi_1 \wedge \varphi_2 \wedge \varphi_{halt}$ . Then  
 $w' = w = (z + 1)^2 = (z' + 1)^2$

QED

## Proof (cnt)

Claim 3.  $\psi_1 \triangleq z^2 \leq x$  inductive relative to  $\varphi_2$

Proof We establish

1.  $\varphi_{init} \Rightarrow \psi_1$
2.  $\psi_1 \wedge \varphi_2 \wedge \varphi_{compute} \Rightarrow \psi'_1$
3.  $\psi_1 \wedge \varphi_2 \wedge \varphi_{halt} \Rightarrow \psi'_1$

Ad 1. Assume  $\varphi_{init}$ . Then  
 $z^2 = 0 \leq x$

Ad 2. Assume  $\psi_1 \wedge \varphi_2 \wedge \varphi_{compute}$ . Then  
 $z'^2 = (z + 1)^2 = w \leq x = x'$

Ad 3. Assume  $\psi_1 \wedge \varphi_2 \wedge \varphi_{halt}$ . Then  
 $z'^2 = z^2 \leq x = x'$

QED

## Proof (cnt)

Claim 4.  $\psi_2 \stackrel{\Delta}{=} done \Rightarrow x < (z + 1)^2$  inductive relative to  $\varphi_2$

Proof We establish

1.  $\varphi_{init} \Rightarrow \psi_2$
2.  $\psi_2 \wedge \varphi_2 \wedge \varphi_{compute} \Rightarrow \psi'_2$
3.  $\psi_2 \wedge \varphi_2 \wedge \varphi_{halt} \Rightarrow \psi'_2$

Ad 1. Assume  $\varphi_{init}$ . Then  $\neg done$ . Hence  $\psi_2$ .

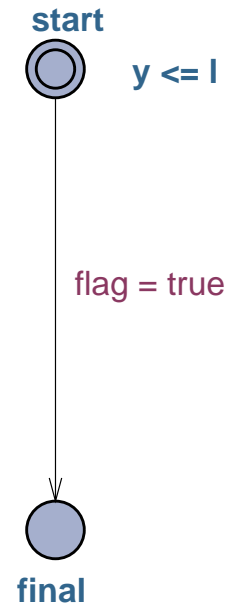
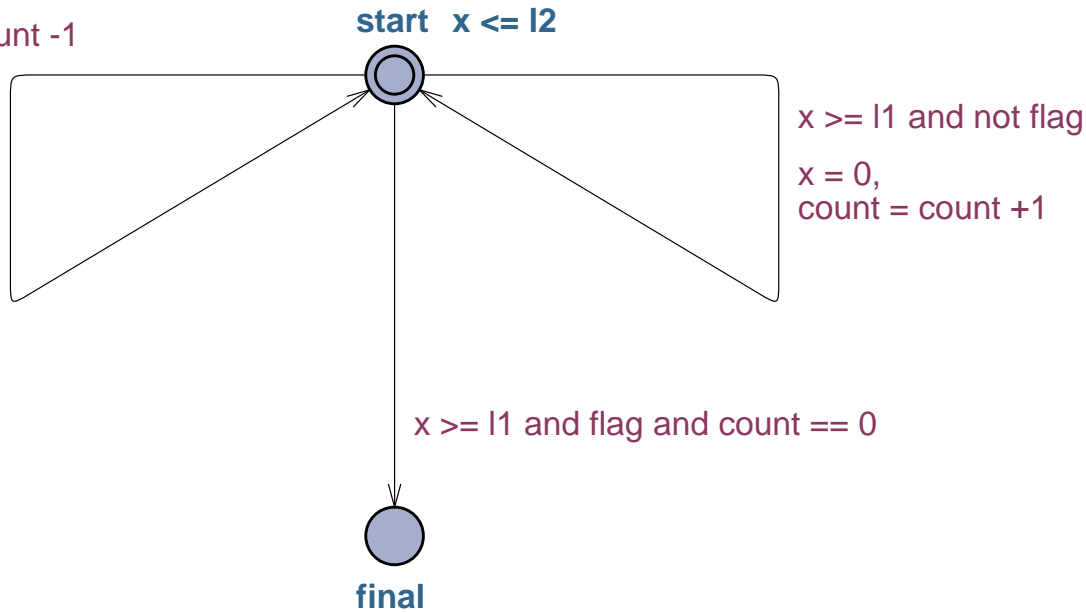
Ad 2. Assume  $\psi_2 \wedge \varphi_2 \wedge \varphi_{compute}$ .  
Then  $\neg done'$ . Hence  $\psi'_2$ .

Ad 3. Assume  $\psi_2 \wedge \varphi_2 \wedge \varphi_{halt}$ .  
Then  $x' = x < w = (z + 1)^2 = (z' + 1)^2$ .  
Hence  $\psi'_2$ .

QED

# Invariant Proofs for Timed Automata

$x \geq l_1$  and flag and count > 0  
 $x = 0,$   
 count = count - 1



How to prove invariant  $\psi \triangleq Main.start \Rightarrow time \leq l + \lfloor \frac{l}{l_1} \rfloor \cdot l_2 + l_2$ ?

**Proof** First translate Uppaal model into logic:

$$\varphi_{init} \triangleq \text{Main.start} \wedge \text{Int.start} \wedge \neg \text{flag} \wedge \text{count} = x = y = \text{time} = 0$$

$$\varphi_p \triangleq \text{Main.start} \wedge x \geq l_1 \wedge \neg \text{flag} \wedge x' = 0 \wedge \text{count}' = \text{count} + 1 \wedge \\ \text{Unchanged}(\text{loc}_{\text{Main}}, \text{loc}_{\text{Int}}, \text{flag}, y, \text{time})$$

$$\varphi_n \triangleq \text{Main.start} \wedge x \geq l_1 \wedge \text{flag} \wedge \text{count} > 0 \wedge x' = 0 \wedge \text{count}' = \text{count} - 1 \wedge \\ \text{Unchanged}(\text{loc}_{\text{Main}}, \text{loc}_{\text{Int}}, \text{flag}, y, \text{time})$$

$$\varphi_z \triangleq \text{Main.start} \wedge x \geq l_1 \wedge \text{flag} \wedge \text{count} = 0 \wedge \text{Main.final}' \wedge \\ \text{Unchanged}(\text{loc}_{\text{Int}}, \text{flag}, \text{count}, x, y, \text{time})$$

$$\varphi_f \triangleq \text{Int.start} \wedge \text{Int.final}' \wedge \text{flag}' \wedge \\ \text{Unchanged}(\text{loc}_{\text{Main}}, \text{count}, x, y, \text{time})$$

$$\varphi_t(d) \triangleq x' = x + d \wedge y' = y + d \wedge \text{time}' = \text{time} + d \wedge \\ (\text{Main.start} \Rightarrow x' \leq l_2) \wedge (\text{Int.start} \Rightarrow y' \leq l) \wedge \\ \text{Unchanged}(\text{loc}_{\text{Main}}, \text{loc}_{\text{Int}}, \text{flag}, \text{count})$$

Here

- $loc_A$  is a state variable that records the location of automaton  $A$
- $A.l$  abbreviates  $loc_A = l$  (as in Uppaal)
- $Unchanged(v_1, \dots, v_n)$  abbreviates  $v'_1 = v_1 \wedge \dots \wedge v'_n = v_n$

## Proof: Simple Inductive Properties

$$\begin{aligned}\psi_1 &\triangleq x \geq 0 \\ \psi_2 &\triangleq (\text{Main.start} \Rightarrow x \leq l_2) \\ \psi_3 &\triangleq y = \text{time} \geq 0 \\ \psi_4 &\triangleq (\text{Int.start} \Rightarrow y \leq l) \wedge \\ \psi_5 &\triangleq \text{count} \geq 0 \\ \psi_6 &\triangleq \text{Int.start} \Leftrightarrow \neg \text{flag}\end{aligned}$$

## Proof (cnt)

Claim.  $\psi_7 \triangleq l_1 \cdot count + x \leq y$  inductive.

Proof Interesting case:

$$\psi_7 \wedge \varphi_p \Rightarrow \psi'_7$$

Assume  $\psi_7 \wedge \varphi_p$ . Then

$$l_1 \cdot count' + x' = l_1 \cdot (count + 1) + 0 \leq l_1 \cdot count + x \leq y = y'$$

Hence  $\psi'_7$ .

## Proof (cnt)

Claim.  $\psi_8 \stackrel{\Delta}{=} \text{count} \leq \lfloor \frac{l}{l_1} \rfloor$  inductive relative to  $\psi_i, i < 8$ .

Proof Interesting case:

$$\bigwedge_{i \leq 8} \psi_i \wedge \varphi_p \wedge \bigwedge_{i < 8} \psi'_i \Rightarrow \psi'_8$$

Assume  $\bigwedge_{i \leq 8} \psi_i \wedge \varphi_p \wedge \bigwedge_{i < 8} \psi'_i$ . Then

$$\text{count}' \leq \frac{y' - x'}{l_1} \leq \frac{y'}{l_1} \leq \frac{l}{l_1}$$

Since  $\text{count}$  is an integer

$$\text{count}' \leq \lfloor \frac{l}{l_1} \rfloor$$

That is  $\psi'_8$ .

## Proof (cnt)

Claim.  $\psi_9 \triangleq \text{Main.start} \wedge \text{flag} \Rightarrow \text{time} \leq l + (\lfloor \frac{l}{l_1} \rfloor - \text{count}) \cdot l_2 + x$  inductive relative to  $\psi_i, i < 9$ .

Claim.  $\psi \triangleq \text{Main.start} \Rightarrow \text{time} \leq l + \lfloor \frac{l}{l_1} \rfloor \cdot l_2 + l_2$  is invariant.