CS 357 D

Lecture 3

Proving invariants

http://cs357d.stanford.edu/

April 10, 2007

Computational Model

Behaviors: sequences of states

System description: state transition systems

compact first-order representation of all sequences of states that can be generated by a system

Programming language: SPL (simple programming language)

with well-defined semantics in terms of transition systems

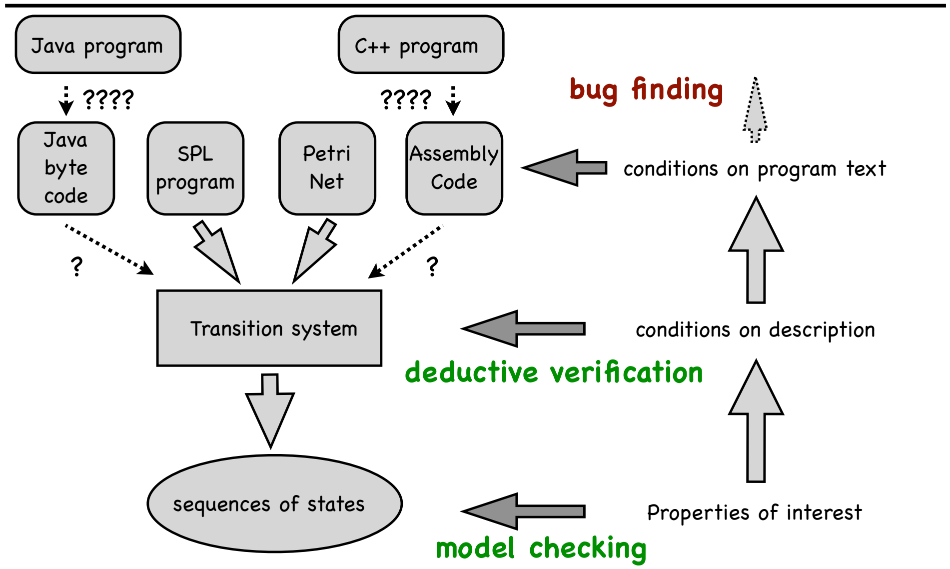
Reference:
Zohar Manna, Amir Pnueli, Temporal Verification of Reactive Systems: Safety, Springer-Verlag, 1995.

Properties of interest

Invariants: overapproximation of the reachable state space

Loop termination: demonstrated by the existence of a ranking function

Semantics



System Description: Transition systems

Set of typed variables

Example: {x:int, y:int}

Φ: < V , Θ , $\mathcal{T}$ >

Initial condition: first-order formula

Example: x=0 $\wedge$ y=0

Set of transitions

Compact first-order representation of all sequences of states that can be generated by a system

Runs

Infinite sequence of states

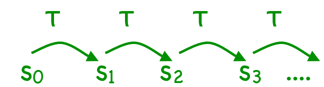
σ: s_0 s_1 s_2 s_3 s_4 ....

is a **run** of Φ if

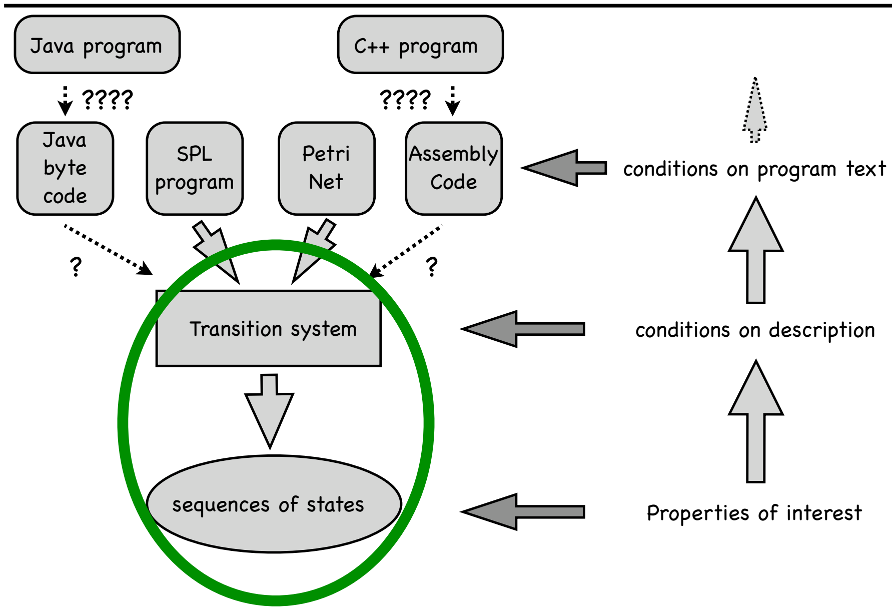
- **Initiality:** $s_0 \models \Theta$ (s_0 is an initial state)
- **Consecution:** for all i > 0

  s_{i+1} is a τ-successor of s_i

  for some $\tau \in \mathcal{T}$



Semantics



SPL: Simple Programming Language

Given an SPL program P we can construct the corresponding transition system Φ: < V , Θ , $\mathcal{T}$ >.

- each program statement corresponds to a transition

  no sequential structure in transition systems, therefore control is modeled explicitly by a control variable π that ranges over program locations

- V: program variables $\cup$ {π}
- Θ: program initial condition

SPL example

```
local x,y: integer where x=N ∧ y=0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

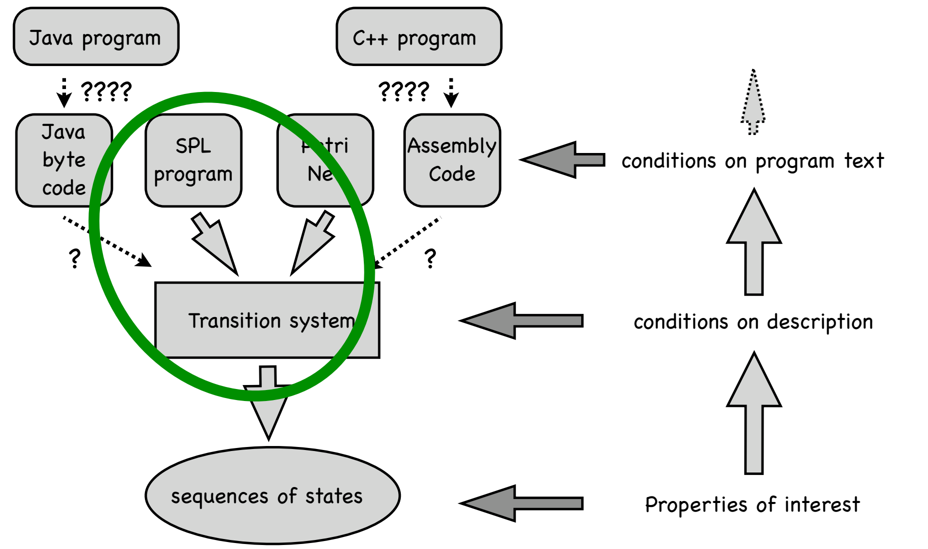
Φ: < V , Θ , $\mathcal{T}$ > with

V : { x:int , y:int , π:{ l_1, l_2 , l_3 , l_4 } } Θ : $x = N \wedge y = 0 \wedge \pi = l_1$

$\mathcal{T}$: { τ_1 , τ_2 , τ_3 , τ_4 } with

$\rho_{\tau 1}$: $\pi = l_1 \wedge ((x > 0 \wedge \pi' = l_2) \vee (x \leq 0 \wedge \pi' = l_4)) \wedge$ pres({ x , y })
$\rho_{\tau 2}$: $\pi = l_2 \wedge \pi' = l_3 \wedge x' = x - 1 \wedge y' = y$
$\rho_{\tau 3}$: $\pi = l_3 \wedge \pi' = l_1 \wedge y' = y + x \wedge x' = x$
$\rho_{\tau 4}$: pres({ x , y , π })

Semantics



Reachable state space

state s is Φ-reachable if it appears in some Φ-run

σ: s_0 s_1 s_2 s_3 s_4 ....

system Φ is finite-state if the set of Φ-reachable states is finite

Notation: Σ : state space
$\Sigma_{\Phi_\triangleright}$: Φ-reachable state space

Reachable state space

```
local x: integer where x > 0
l1: while x ≠ 1 do [
    l2: if odd(x) then
           l3: x := 3x + 1 ;
        else
           l4: x := x / 2 ;
    ]
l5:
```

size of the reachable state space not known in general

Example runs:

3, 10, 5, 16, 8, 4, 2, 1
7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
9, 28, 14, 7, ...
19, 58, 29, 88, 44, 22, ...

Reachable state space vs runs

System Φ may have any combination of

finite state space	$\otimes$	finite # of runs
infinite state space		infinite # of runs

Invariants

An **invariant** q of program P:

- is a superset of the reachable state space of P
- q is an **assertion** (first-order formula)
- also written:

$P \Vdash q$ all reachable states of P satisfy q

$P \vDash \Box q$ all states of all runs of P satisfy q

Invariants: examples

absence of array out-of-bounds accesses:

```
A: array[1..N] of integer
i : integer
.......
ℓ: A[i] := 7
........
```

$(\pi = \ell) \rightarrow 1 \leq i \leq N$

absence of division by zero

```
x,y,z: integer
.......
.......
ℓ: x := y / z
.......
```

$(\pi = \ell) \rightarrow z \neq 0$

Invariants: example

```
local x,y: integer where x=2 ∧ y=0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

reachable state space:

$\{ (2, 0, l_1), (2, 0, l_2), (1, 0, l_3), (1, 1, l_1), (1, 1, l_2), (0, 1, l_3), (0, 1, l_1), (0, 1, l_4) \}$

some invariants:

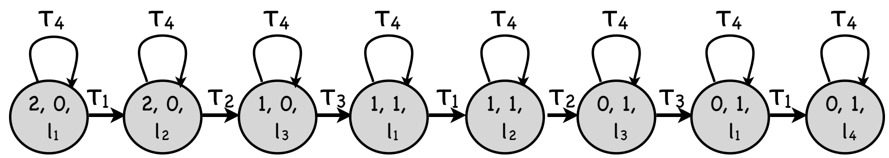
$0 \leq x \leq 2$	$(\pi = l_4) \rightarrow (y = 1)$	$y \leq x + 1$
$0 \leq y \leq 1$	$(\pi = l_4) \rightarrow (x = 0)$	$(\pi = l_3) \rightarrow x + y = 1$

Proving invariants by model checking: example

To prove $y \le x + 2$:

1. Construct the reachable state space

$\Theta : x = 2 \wedge y = 0 \wedge \pi = l_1$



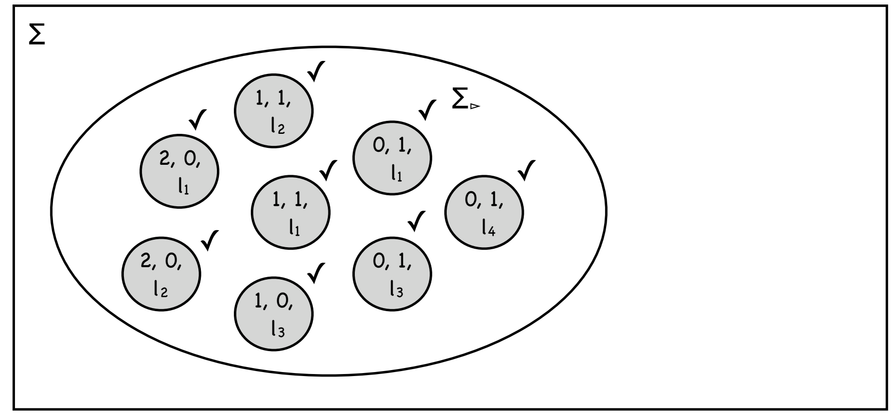
$\mathcal{T}$: { τ_1 , τ_2 , τ_3 , τ_4 } with

$\rho_{\tau 1} : \pi = l_1 \wedge ((x > 0 \wedge \pi' = l_2) \vee (x \le 0 \wedge \pi' = l_4)) \wedge pres(\{ x , y \})$
$\rho_{\tau 2} : \pi = l_2 \wedge \pi' = l_3 \wedge x' = x - 1 \wedge y' = y$
$\rho_{\tau 3} : \pi = l_3 \wedge \pi' = l_1 \wedge y' = y + x \wedge x' = x$
$\rho_{\tau 4} : pres(\{ x , y , \pi \})$

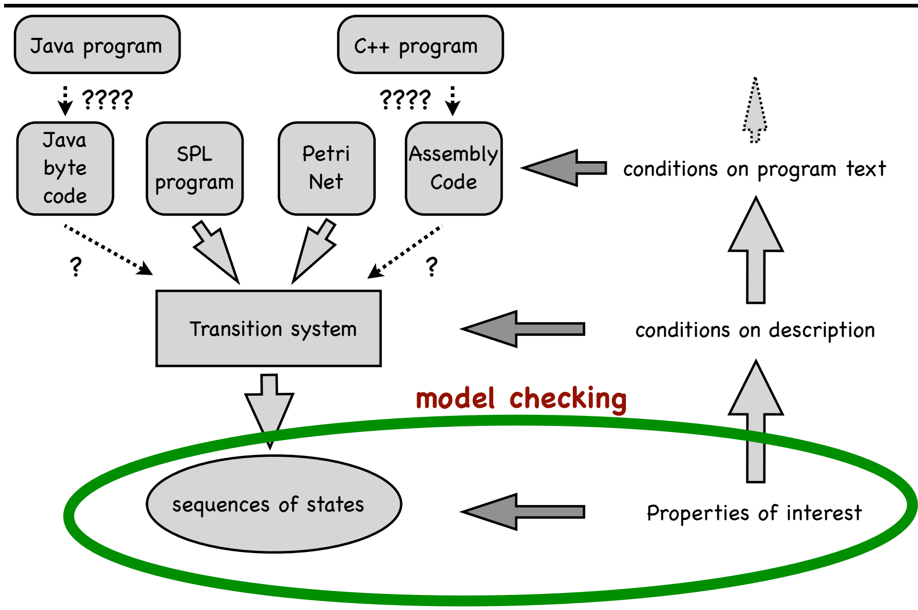
Proving invariants by model checking: example

To prove $y \le x + 2$:

2. Check that all reachable states satisfy $y \le x + 2$



Semantics

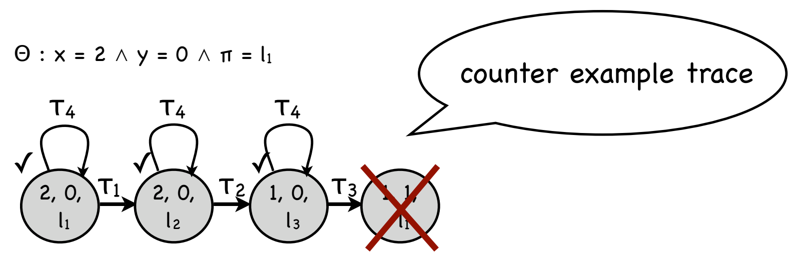


Proving invariants by model checking: example

Or check on the fly

Trying to prove $x \neq y$ is invariant:

$\Theta : x = 2 \wedge y = 0 \wedge \pi = l_1$



$\mathcal{T}$: { τ_1 , τ_2 , τ_3 , τ_4 } with

$\rho_{\tau 1} : \pi = l_1 \wedge ((x > 0 \wedge \pi' = l_2) \vee (x \le 0 \wedge \pi' = l_4)) \wedge pres(\{ x , y \})$
$\rho_{\tau 2} : \pi = l_2 \wedge \pi' = l_3 \wedge x' = x - 1 \wedge y' = y$
$\rho_{\tau 3} : \pi = l_3 \wedge \pi' = l_1 \wedge y' = y + x \wedge x' = x$
$\rho_{\tau 4} : pres(\{ x , y , \pi \})$

Invariants: example

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

invariants ?

$0 \leq x \leq 2$ $\quad$ $(\pi = l_4) \rightarrow (y = 1)$ $\quad$ $y \leq x + 1$

$0 \leq y \leq 1$ $\quad$ $(\pi = l_4) \rightarrow (x = 0)$ $\quad$ $(\pi = l_3) \rightarrow x + y = 1$

replace by N or N-1 ?

Invariants: example

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

invariants ?

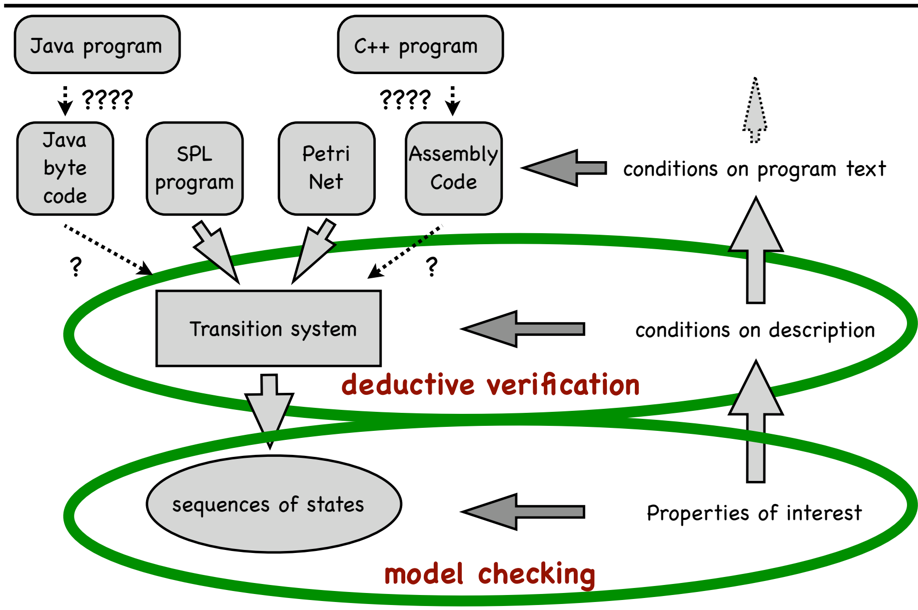
$0 \leq x \leq N$ $\quad$ $(\pi = l_4) \rightarrow (y = N - 1)$ $\quad$ $y \leq x + (N - 1)$

$0 \leq y \leq N - 1$ $\quad$ $(\pi = l_4) \rightarrow (x = 0)$ $\quad$ $(\pi = l_3) \rightarrow x + y = 1$

How do we check?

Model checking for N = 1, N = 2, N = 3, N = 4, N = 5, ...

Semantics



Proving invariance properties deductively

To prove that assertion p is an invariant of system Φ :

(every state of every run of Φ satisfies p)

it is sufficient to prove that (proof by induction on the run)

- p holds at the beginning of every run (base case)
- p is preserved by every transition τ (inductive step)

Proving invariance properties deductively : initiation

These conditions can be expressed in first-order logic:

- p holds at the beginning of every run (base case)

From the definition of a a run:

a sequence of states $s_0\ s_1\ s_2.....$ is a run if

Initiality: $s_0 \vDash \Theta$ (all initial states must satisfy Θ)

sufficient condition for p to hold at all initial states:

$$\Theta \rightarrow p$$

(Θ implies p)

Proving invariance properties deductively : consecution

These conditions can be expressed in first-order logic:

- p is preserved by every transition τ (inductive step)

From the definition of a a run:

a sequence of states $s_0\ s_1\ s_2.....$ is a run if

Consecution: for each $j \geq 0$, s_{j+1} is a τ-successor of s_j, for some $\tau \in \mathcal{T}$

induction step: assume p holds on s_j -- to prove: p holds on s_{j+1} after taking τ

in first-order logic: $p \wedge \rho_\tau \rightarrow p'$

Proving invariance properties deductively: example

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

invariant to prove: $x \leq N$

- p holds at the beginning of every run: (base case) $\Theta \rightarrow p$

$$\underbrace{x = N \wedge y = 0 \wedge N > 0 \wedge \pi = l_1}_{\Theta} \rightarrow \underbrace{x \leq N}_{p} \quad \text{valid}$$

Proving invariance properties deductively: example

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

invariant to prove: $x \leq N$

- p is preserved by every transition τ (inductive step) $p \wedge \rho_\tau \rightarrow p'$

Proving invariance properties deductively: example

$\mathcal{T}$: { τ_1 , τ_2 , τ_3 , τ_4 } with

$\rho_{\tau 1}$: $\pi = l_1 \wedge ((x > 0 \wedge \pi' = l_2) \vee (x \leq 0 \wedge \pi' = l_4)) \wedge pres(\{ x , y \})$

$\rho_{\tau 2}$: $\pi = l_2 \wedge \pi' = l_3 \wedge x' = x - 1 \wedge y' = y$

$\rho_{\tau 3}$: $\pi = l_3 \wedge \pi' = l_1 \wedge y' = y + x \wedge x' = x$

$\rho_{\tau 4}$: $pres(\{ x , y , \pi \})$

- p is preserved by every transition τ
  (inductive step)

$p \wedge \rho_\tau \rightarrow p'$

$\tau_1 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x \wedge \ldots\ldots \rightarrow x' \leq N$ valid

$\tau_2 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x - 1 \wedge \ldots\ldots \rightarrow x' \leq N$ valid

$\tau_3 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x \wedge \ldots\ldots \rightarrow x' \leq N$ valid

$\tau_4 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x \wedge \ldots\ldots \rightarrow x' \leq N$ valid

Proving invariance properties deductively: example

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

$x \leq N$

is an invariant for all values of N > 0

Proof: (validity of 5 first-order formulas)

$x = N \wedge y = 0 \wedge N > 0 \wedge \pi = l_1 \rightarrow x \leq N$

$\tau_1 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x \wedge \ldots\ldots \rightarrow x' \leq N$

$\tau_2 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x - 1 \wedge \ldots\ldots \rightarrow x' \leq N$

$\tau_3 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x \wedge \ldots\ldots \rightarrow x' \leq N$

$\tau_4 : x \leq N \wedge \ldots\ldots\ldots \wedge x' = x \wedge \ldots\ldots \rightarrow x' \leq N$

Verification rule B-INV (basic invariance)

For assertion p

B1. $\Theta \rightarrow p$

B2. $p \wedge \rho_\tau \rightarrow p'$ for all τ in $\mathcal{T}$

$\Box p$ (p is an invariant of Φ)

B-INV reduces the proof of an invariant to checking the validity of $|\mathcal{T}| + 1$ first-order formulas (verification conditions in the underlying assertion language).

Proving invariance properties deductively: example

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

invariant to prove:

$x \geq 0$

- p holds at the beginning of every run:
  (base case)

$\Theta \rightarrow p$

$\underbrace{x = N \wedge y = 0 \wedge N > 0 \wedge \pi = l_1}_{\Theta} \rightarrow \underbrace{x \geq 0}_{p}$ valid

Proving invariance properties deductively: example

$\mathcal{T}$: { τ_1 , τ_2 , τ_3 , τ_4 } with

$\rho_{\tau 1}$: $\pi = l_1 \wedge ((x > 0 \wedge \pi' = l_2) \vee (x \leq 0 \wedge \pi' = l_4)) \wedge pres(\{ x , y \})$

$\rho_{\tau 2}$: $\pi = l_2 \wedge \pi' = l_3 \wedge x' = x - 1 \wedge y' = y$

$\rho_{\tau 3}$: $\pi = l_3 \wedge \pi' = l_1 \wedge y' = y + x \wedge x' = x$

$\rho_{\tau 4}$: $pres(\{ x , y , \pi \})$

☛ p is preserved by every transition τ

(inductive step)

$p \wedge \rho_\tau \rightarrow p'$

τ_1 : $x \geq 0 \wedge$ .... $\wedge\ x' = x \wedge$ .... $\rightarrow\ x' \geq 0$ valid

τ_2 : $x \geq 0 \wedge$ .... $\wedge\ x' = x - 1 \wedge$ .... $\rightarrow\ x' \geq 0$ not valid

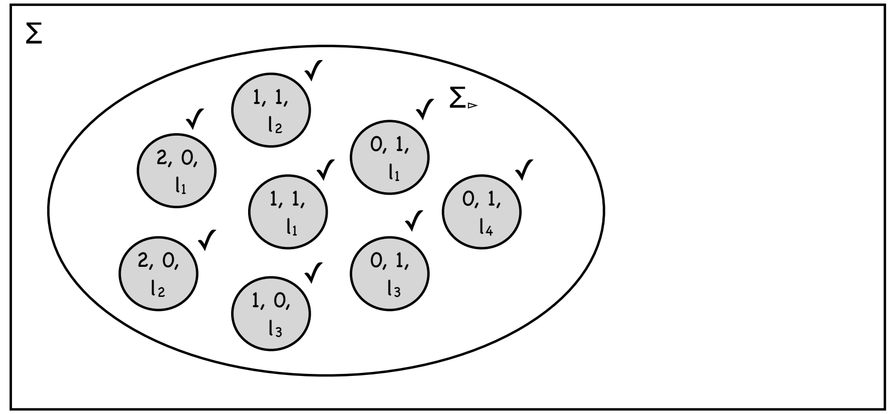
τ_3 : $x \geq 0 \wedge$ .... $\wedge\ x' = x \wedge$ .... $\rightarrow\ x' \geq 0$ valid

τ_4 : $x \geq 0 \wedge$ .... $\wedge\ x' = x \wedge$ .... $\rightarrow\ x' \geq 0$ valid

what is the problem?

To prove **$x \geq 0$** : (for N = 2)

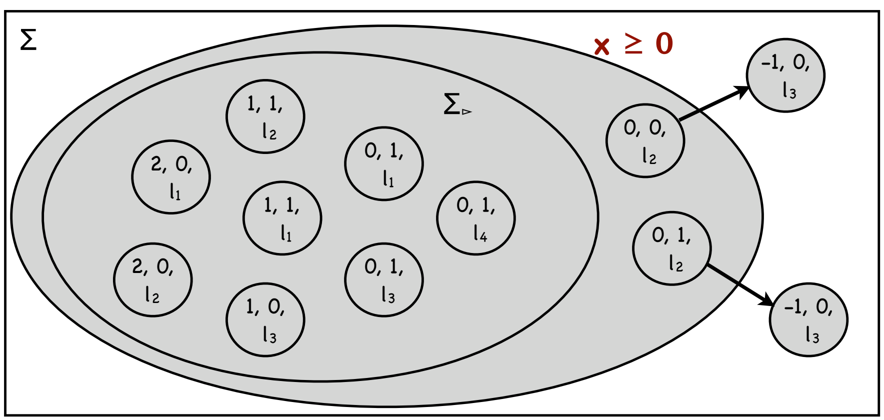
(Model checking) check that all reachable states satisfy **$x \geq 0$**



what is the problem?

To prove **$x \geq 0$** : (deductively for N = 2 ?)

τ_2 : $x \geq 0 \wedge$ .... $\wedge\ x' = x - 1 \wedge$ .... $\rightarrow\ x' \geq 0$



what is the problem?

τ_2 : $x \geq 0$ $\wedge$ .... $\wedge\ x' = x - 1 \wedge$ .... $\rightarrow\ x' \geq 0$

inductive hypothesis is too weak

it is not preserved by all transitions

$x \geq 0$ is an invariant, but it is not **inductive**

it cannot be proven deductively directly

Solution: strengthen the inductive hypothesis

identify the problem states

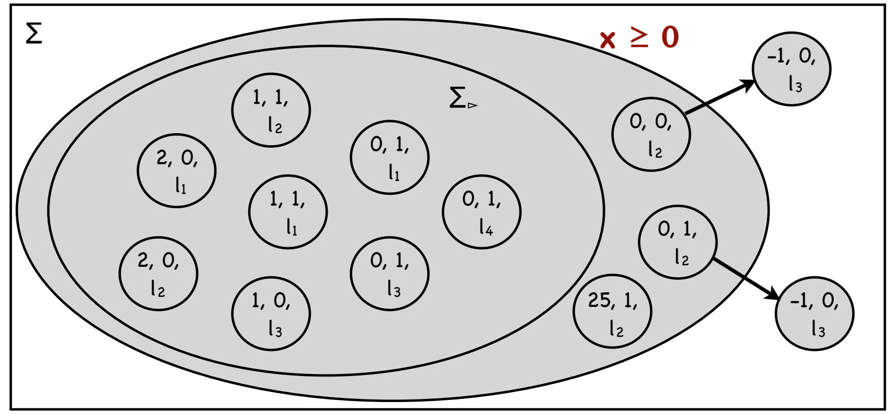
$(0, 0, l_2), (0, 1, l_2), \ldots\ldots\ldots\ldots$

in general: $(0, y, l_2)$ for any value of y

remove them by strengthening the inductive hypothesis

$x \geq 0 \;\wedge\; ((\pi = l_2) \rightarrow x > 0)$

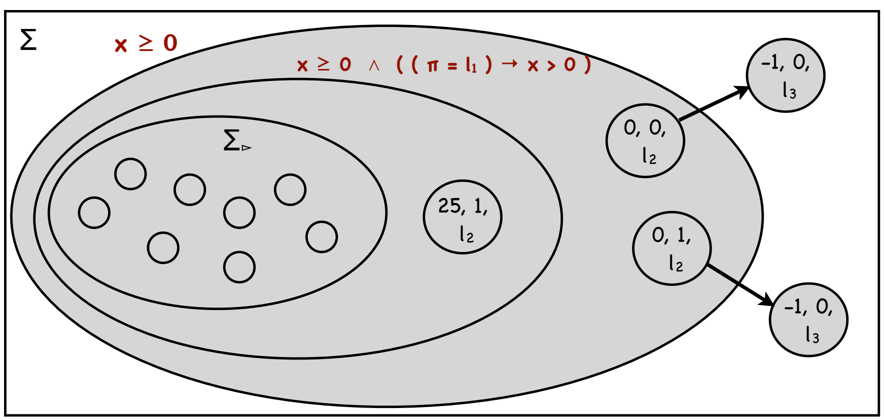
what is the problem?



Transition τ_2 is preserved

$\tau_2 : x \geq 0 \wedge ((\pi = l_2) \rightarrow x > 0) \wedge \pi = l_2 \wedge x' = x - 1 \wedge \pi' = l_3$

$\rightarrow \; x' \geq 0 \wedge ((\pi' = l_2) \rightarrow x' > 0)$



How about the initial condition?

- p holds at the beginning of every run: $\Theta \rightarrow p$

  (base case)

$\underbrace{x = N \wedge y = 0 \wedge N > 0 \wedge \pi = l_1}_{\Theta} \rightarrow \underbrace{x \geq 0 \wedge ((\pi = l_2) \rightarrow x > 0)}_{p}$

still valid

How about the other transitions?

$\mathcal{T}$: $\{ \tau_1 , \tau_2 , \tau_3 , \tau_4 \}$ with

$\rho_{\tau_1} : \pi = l_1 \wedge ((x > 0 \wedge \pi' = l_2) \vee (x \leq 0 \wedge \pi' = l_4)) \wedge pres(\{ x , y \})$

$\rho_{\tau_2} : \pi = l_2 \wedge \pi' = l_3 \wedge x' = x - 1 \wedge y' = y$

$\rho_{\tau_3} : \pi = l_3 \wedge \pi' = l_1 \wedge y' = y + x \wedge x' = x$

$\rho_{\tau_4} : pres(\{ x , y , \pi \})$

$p \wedge \rho_\tau \rightarrow p'$

- p is preserved by every transition τ
  (inductive step)

$\tau_1 : x \geq 0 \wedge ((\pi = l_2) \rightarrow x > 0) \wedge$
$(x > 0 \wedge \pi' = l_2) \wedge \ldots\ldots\ldots \wedge \; x' = x \wedge \ldots\ldots\ldots$
$\rightarrow$
$x' \geq 0 \wedge ((\pi' = l_2) \rightarrow x > 0)$

valid

Summary of proof of $x \geq 0$

To prove $x \geq 0$:

Application of B-INV did not work : $x \geq 0$ was too weak

Strengthen into

$x \geq 0 \wedge ((\pi = l_2) \rightarrow x > 0)$ (implies the invariant we want to prove)

Application of B-INV on stronger invariant works:
all verification conditions are valid

Verification rule B-INV (basic invariance)

For assertion p

B1. $\Theta \rightarrow p$

B2. $p \wedge \rho_\tau \rightarrow p'$ for all τ in $\mathcal{T}$

$\Box p$ (p is an invariant of Φ)

B-INV reduces the proof of an invariant to checking the validity of $|\mathcal{T}| + 1$ first-order formulas (verification conditions in the underlying assertion language).

Verification rule G-INV (general invariance)

For assertions φ, p

I1. $\varphi \rightarrow p$

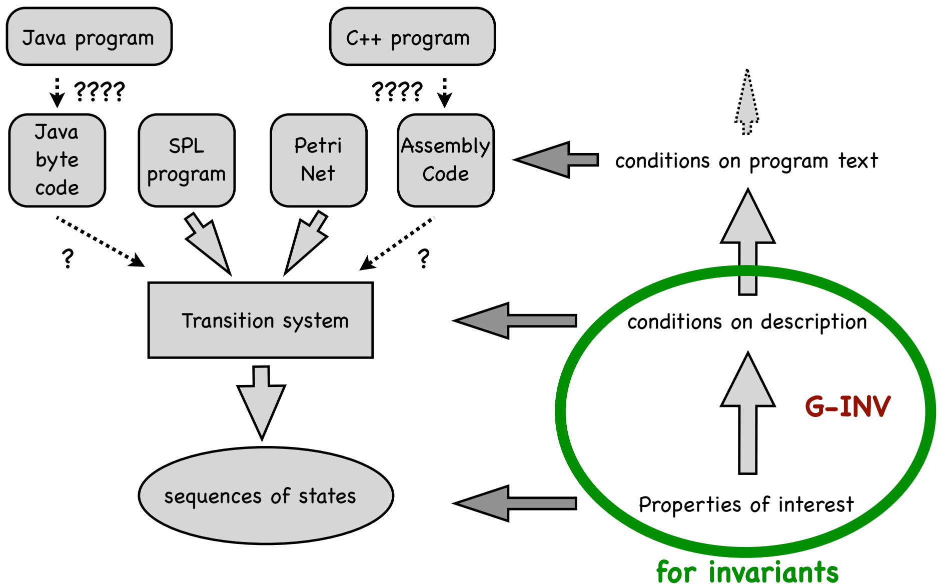
I2. $\Theta \rightarrow \varphi$

I3. $\varphi \wedge \rho_\tau \rightarrow \varphi'$ for all τ in $\mathcal{T}$

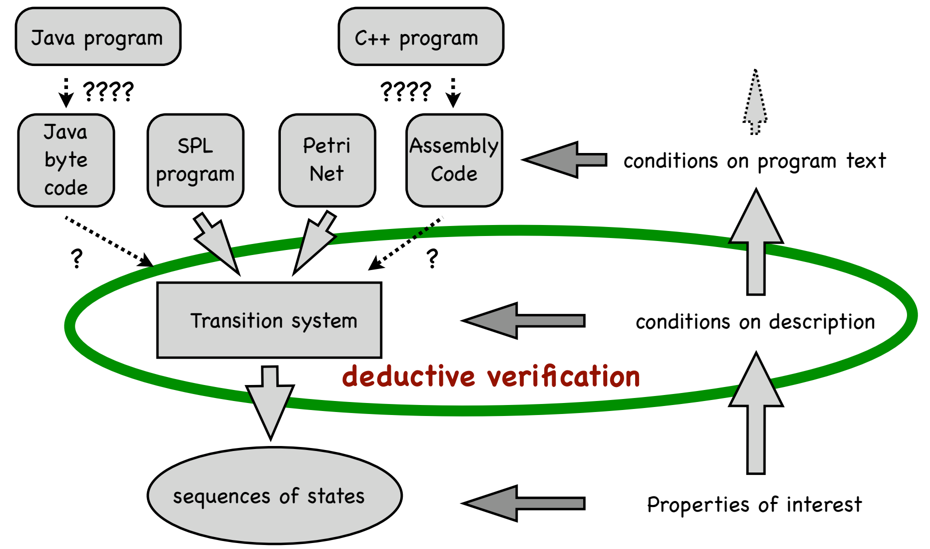
$\Box p$ (p is an invariant of Φ)

G-INV reduces the proof of an invariant p to finding an inductive assertion φ that strengthens p and to checking the validity of $|\mathcal{T}| + 2$ first-order formulas (verification conditions in the underlying assertion language).

Semantics



Semantics



Verification rule G-INV (general invariance)

For assertions φ, p

I1. $\varphi \rightarrow p$

I2. $\Theta \rightarrow \varphi$

I3. $\varphi \wedge \rho_\tau \rightarrow \varphi'$ for all τ in $\mathcal{T}$

$\square p$ (p is an invariant of Φ)

G-INV reduces the proof of an invariant p to finding an inductive assertion φ that strengthens p and to checking the validity of $|\mathcal{T}| + 2$ first-order formulas (verification conditions in the underlying assertion language).

The Big Question

How do we find φ ?

40 years of research has not answered this question

Verification rule G-INV (general invariance)

For assertions φ, p

I1. $\varphi \rightarrow p$

I2. $\Theta \rightarrow \varphi$

I3. $\varphi \wedge \rho_\tau \rightarrow \varphi'$ for all τ in $\mathcal{T}$

$\Box p$ (p is an invariant of Φ)

G-INV is complete:

if p is an invariant of Φ then an assertion φ always exists such that I1 - I3 hold

Reference:
Zohar Manna, Amir Pnueli, Temporal Verification of Reactive Systems: Safety, Springer-Verlag, 1995. Chapter 4.

Verification rule INC-INV (incremental invariance)

For assertion p, q_1 .... q_n

B0. $\Box q_1$ ...... $\Box q_n$

B1. $\Theta \rightarrow p$

B2. $p \wedge q_1 \wedge \ldots\ldots \wedge q_n \wedge \rho_\tau \rightarrow p'$ for all τ in $\mathcal{T}$

$\Box p$ (p is an invariant of Φ)

Static analysis

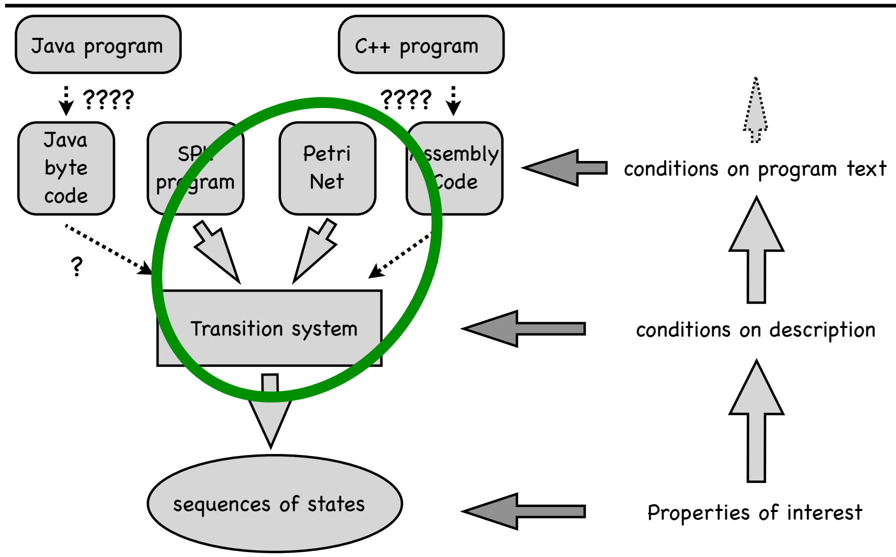
Incremental analysis:

generate many simple q_i's that are inductive

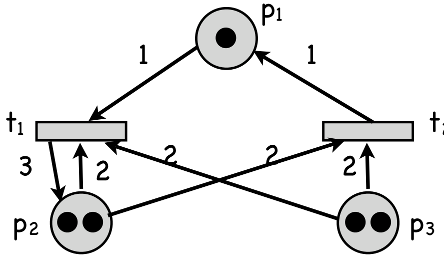
Deep analysis:

generate interesting invariants

Semantics



Petri net semantics: example



described by Φ: < V , Θ , $\mathcal{T}$ > with

V : { p_1 , p_2 , p_3 }

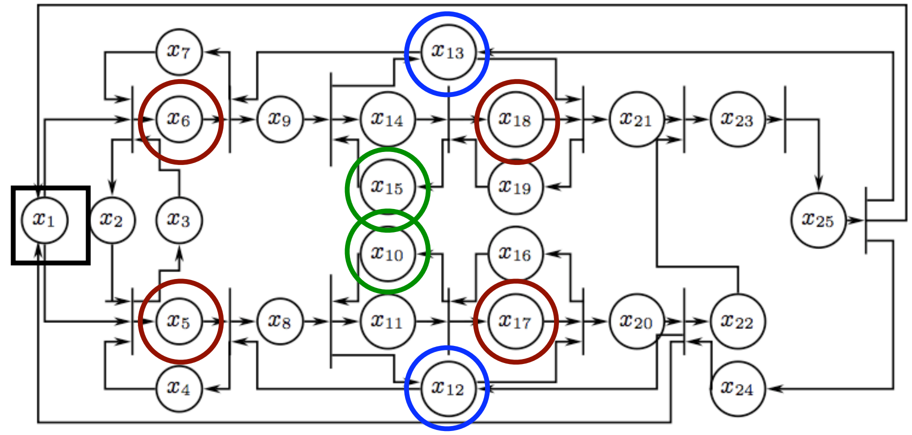
Θ : $p_1 = 1 \wedge p_2 = 2 \wedge p_3 = 2$

$\mathcal{T}$: { t_1 , t_2 } with

$\rho_1 : p_1 \geq 1 \wedge p_2 \geq 2 \wedge p_3 \geq 2 \wedge p_1' = p_1 - 1 \wedge p_2' = p_2 + 1 \wedge p_3' = p_3 - 2$

$\rho_2 : \quad p_2 \geq 2 \wedge p_3 \geq 2 \wedge p_1' = p_1 + 1 \wedge p_2' = p_2 - 2 \wedge p_3' = p_3 - 2$

Petri net: manufacturing system example



Model of a manufacturing system with 4 machines, 2 robots, 2 buffers

Manufacturing system example: description

- Automated manufacturing system with
  - 4 machines M_1 – M_4, whose availability is modeled by x_5, x_6, x_{17}, x_{18}
  - 2 robots R_1 and R_2, whose availability is modeled by x_{12} and x_{13}
  - 2 buffers, modeled by x_{10} and x_{15}
  - delivery area, modeled by x_{25}
- Raw material is introduced in x_1, whose initial marking is parametric (it may start with any number of tokens)
- Raw material passes through two assembly lines, where it is processed by the machines and transported by the robots, and ends up in the delivery area
- Initial marking:

  $x_1 = p$

  $x_2 = x_4 = x_7 = x_{12} = x_{13} = x_{16} = x_{19} = x_{24} = 1$

  $x_{10} = x_{15} = 3$

  all other places: $x_i = 0$

Manufacturing system example: background

Original description:

MengChu Zhou, Frank DiCesare, Alan A. Desrochers, A hybrid methodology for synthesis of petri net models for manufacturing systems. IEEE Transactions on Robotics and Automation, 8(3):350-361, June 1992.

Subsequently analyzed for possibility of deadlocks:

Feng Chu, Xiao-Lan Xie, Deadlock analysis of petri nets using siphons and mathematics p programming. IEEE Transactions on Robotics and Automation, 13(6):793-804, December 1997.

Laurent Fribourg, Hans Olsen, Proving safety properties of infinite-state systems by compilation into Presburger Arithmetic. In Concur'97, LNCS 1243, Springer-Verlag, pp 213-227, 1997.

B. Berard, L. Fribourg, Reachability analysis of (timed) petri nets using real arithmetic. In Concur'99, LNCS 1664, Springer-Verlag, 1999.

Manufacturing system example: our analysis

Described in:

S. Sankaranarayanan, H.B. Sipma, Z. Manna, Petri net analysis using invariant generation. In Verification: Theory and Practice. LNCS 2772. Springer-Verlag, 2004.

Some results:

- generated 1900 invariants
- invariants imply absence of deadlock for initial values $1 \leq x_1 \leq 8$
- invariants imply that the system is bounded
- invariants provide insight in the system structure, for example:

$$x_8 + x_{12} + x_{20} = 1$$
$$x_9 + x_{13} + x_{21} + x_{23} + x_{24} = 1$$

show that robots R_1 and R_2 are not symmetric:

- R_1 is used to transport material from M_1 to M_3 and from M_3 to the packaging area
- R_2 has the same tasks in the other assembly line, but is also responsible to deliver the combined product from the two assembly lines to the output area (x_{25}).

Invariants: exercise

```
local x,y: integer where x = N ∧ y=0 ∧ N > 0
l1: while x > 0 do [
    l2: x := x - 1 ;
    l3: y := y + x ;
    ]
l4:
```

invariants ?

$0 \leq x \leq N$ $\quad (\pi = l_4) \rightarrow (y = N - 1)$ $\quad y \leq x + (N - 1)$

$0 \leq y \leq N - 1$ $\quad (\pi = l_4) \rightarrow (x = 0)$ $\quad (\pi = l_3) \rightarrow x + y = 1$