Properties of interest

Invariants: overapproximation of the reachable state space

Loop termination: demonstrated by the existence of a ranking function

System Description: Transition systems

Set of typed variables
Example: \{x:int, y:int\}

Initial condition:
first-order formula
Example: \(x = 0 \land y = 0\)

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

Runs

Infinite sequence of states

\(\sigma: s_0 \ s_1 \ s_2 \ s_3 \ s_4 \ \ldots\ldots\)

is a run of \(\Phi\) if

- **Initiality:** \(s_0 \models \Theta\)
  \((s_0\) is an initial state)\n
- **Consecution:** for all \(i > 0\)
  \(s_{i+1}\) is a \(T\)-successor of \(s_i\)
  for some \(T \in \mathcal{T}\)

SPL: Simple Programming Language

Given an SPL program \(P\) we can construct the corresponding transition system \(\Phi: <V, \Theta, \mathcal{T}>\).

- each program statement corresponds to a transition
- no sequential structure in transition systems, therefore control is modeled explicitly by a control variable \(n\) that ranges over program locations

- \(V\): program variables \(\cup \{n\}\)
- \(\Theta\): program initial condition
Reachable state space

state s is \( \Phi \)-reachable if it appears in some \( \Phi \)-run

\( \sigma : S_0 | S_1 | S_2 | S_3 | S_4 \) ............

system \( \Phi \) is finite-state if the set of \( \Phi \)-reachable states is finite

Notation: \( \Sigma : \) state space
\( \Sigma_{\Phi} : \Phi \)-reachable state space

Reachable state space

local x, y: integer where \( x=N \land y=0 \)
\( l_0: \) while \( x > 0 \) do
\( \quad l_1: x := x - 1 ; \)
\( \quad l_2: y := y + x ; \)
\( \} \)
\( l_3: \)

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  finite # of runs
- infinite state space infinite # of runs

Invariants: examples

absence of array out-of-bounds accesses:

A: array[1..N] of integer
\[ i : \text{integer} \]
\[ \ldots \]
\[ \ell : A[i] := 7 \]
\[ \ldots \]

absence of division by zero

x,y,z: integer
\[ \ldots \]
\[ \ell : x := y / z \]
\[ \ldots \]

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:

\[ P \models q \]  all reachable states of P satisfy q
\[ P \models \square q \]  all states of all runs of P satisfy q

Invariants: example

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

reachable state space:
\{ (2, 0, l₁), (2, 0, l₂), (1, 0, l₁), (1, 1, l₁), (1, 1, l₂), (0, 1, l₁), (0, 1, l₂) \}

some invariants:

\[ 0 \leq x \leq 2 \]  \[ (\pi = l₄) \rightarrow (y = 1) \]  \[ y \leq x + 1 \]
\[ 0 \leq y \leq 1 \]  \[ (\pi = l₄) \rightarrow (x = 0) \]  \[ (\pi = l₃) \rightarrow x + y = 1 \]
Proving invariants by model checking: example

To prove \( y \leq x + 2 \):

1. Construct the reachable state space

\[ \Theta : x = 2 \land y = 0 \land \pi = l_1 \]

\[ \mathcal{T} : \{ \begin{array}{c} T_4 \end{array} \} \]

\[ \mathcal{J} : \{ \begin{array}{c} T_1, T_2, T_3, T_4 \end{array} \} \]

- \( \rho_{11} : \pi = l_1 \land ( ( x > 0 \land \pi' = l_2 ) \lor ( x \leq 0 \land \pi' = l_4 ) ) \land \text{pres}( \{ x, y \} ) \)
- \( \rho_{12} : \pi = l_2 \land \pi' = l_3 \land x = x - 1 \land y' = y \)
- \( \rho_{13} : \pi = l_3 \land \pi' = l_1 \land y' = y + x \land x' = x \)
- \( \rho_{14} : \text{pres}( \{ x, y, \pi \} ) \)

2. Check that all reachable states satisfy \( y \leq x + 2 \)

\[ \Sigma \]

Or check on the fly

Trying to prove \( x \neq y \) is invariant:

\[ \Theta : x = 2 \land y = 0 \land \pi = l_1 \]

\[ \mathcal{T} : \{ \begin{array}{c} T_4 \end{array} \} \]

\[ \mathcal{J} : \{ \begin{array}{c} T_1, T_2, T_3, T_4 \end{array} \} \]

- \( \rho_{11} : \pi = l_1 \land ( ( x > 0 \land \pi' = l_2 ) \lor ( x \leq 0 \land \pi' = l_4 ) ) \land \text{pres}( \{ x, y \} ) \)
- \( \rho_{12} : \pi = l_2 \land \pi' = l_3 \land x = x - 1 \land y' = y \)
- \( \rho_{13} : \pi = l_3 \land \pi' = l_1 \land y' = y + x \land x' = x \)
- \( \rho_{14} : \text{pres}( \{ x, y, \pi \} ) \)
Invariants: example

```java
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?

- $0 \leq x \leq 2$ (π = l4) → (y = 1)
- $0 \leq y \leq 1$ (π = l4) → (x = 0)
- $\pi = l3$ → $x + y = 1$

replace by N or N-1?

---

Invariants: example

```java
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?

- $0 \leq x \leq N$ (π = l4) → (y = N - 1)
- $0 \leq y \leq N - 1$ (π = l4) → (x = 0)
- $\pi = l3$ → $x + y = 1$

How do we check?

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

---

Semantics

- Java program
- C++ program
- Java byte code
- SPL program
- Petri Net
- Assembly Code

Transition system

conditions on program text

conditions on description

deductive verification

sequences of states

Properties of interest

model checking

---

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
- p holds at the beginning of every run ( base case )
- p is preserved by every transition T ( 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 $s_0, s_1, s_2, \ldots$ is a run if

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

sufficient condition for $p$ to hold at all initial states:

$$\Theta \rightarrow p$$

( $\Theta$ implies $p$ )

---

Proving invariance properties deductively: example

local $x, y$: integer where $x = N \land y = 0 \land N > 0$

l1: while $x > 0$ do [
    l2: $x := x - 1$
    l3: $y := y + x$
]
l4:

$\Phi$ holds at the beginning of every run:

( base case )

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

valid

$$\Theta \rightarrow p$$

( $\Theta$ implies $p$ )

---

Proving invariance properties deductively: consecution

These conditions can be expressed in first-order logic:

- $p$ is preserved by every transition $\tau$ (inductive step)

From the definition of a $a$ run:

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

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

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

in first-order logic:

$$p \land \rho_\tau \rightarrow p'$$

---

Proving invariance properties deductively: example

local $x, y$: integer where $x = N \land y = 0 \land N > 0$

l1: while $x > 0$ do [
    l2: $x := x - 1$
    l3: $y := y + x$
]
l4:

$\Phi$ holds at the beginning of every run:

( base case )

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

valid

$$\Theta \rightarrow p$$

( $\Theta$ implies $p$ )

---

Proving invariance properties deductively: example

invariant to prove:

$$x \leq N$$

---

Proving invariance properties deductively: example

invariant to prove:

$$x \leq N$$
Proving invariance properties deductively: example

\[ \mathcal{J} : \{ T_1 , T_2 , T_3 , T_4 \} \text{ with } \]
\[ \rho_{T_1} : \pi = l_1 \land \left( ( x > 0 \land n' = l_2 ) \lor ( x \leq 0 \land n' = l_4 ) \right) \land \text{pres}( \{ x , y \} ) \]
\[ \rho_{T_2} : \pi = l_2 \land n' = l_3 \land x' = x - 1 \land y' = y \]
\[ \rho_{T_3} : \pi = l_3 \land n' = l_1 \land y' = y + x \land x' = x \]
\[ \rho_{T_4} : \text{pres}( \{ x , y , \pi \} ) \]

\( p \) is preserved by every transition \( \tau \)

( inductive step )

\[ T_1 : x \leq N \land \ldots \land x' = x \land \ldots \rightarrow x' \leq N \quad \text{valid} \]
\[ T_2 : x \leq N \land \ldots \land x' = x - 1 \land \ldots \rightarrow x' \leq N \quad \text{valid} \]
\[ T_3 : x \leq N \land \ldots \land x' = x \land \ldots \rightarrow x' \leq N \quad \text{valid} \]
\[ T_4 : x \leq N \land \ldots \land x' = x \land \ldots \rightarrow x' \leq N \quad \text{valid} \]

Verification rule B-INV (basic invariance)

For assertion \( p \)

\[ \begin{align*}
B1. & \quad \Theta \rightarrow p \\
B2. & \quad p \land \rho_\tau \rightarrow p' \quad \text{for all } \tau \text{ in } \mathcal{J} \\
\hline \\
& \quad p \quad ( p \text{ is an invariant of } \Theta )
\end{align*} \]

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

Proving invariance properties deductively: example

\[ \text{local } x,y : \text{integer where } x = N \land y = 0 \land N > 0 \]
\[ l_1 : \text{while } x > 0 \text{ do } [ \\
\quad l_2: x := x - 1 ; \\
\quad l_3: y := y + x ; \\
\quad ] \\
\quad l_4: \]

Proof: (validity of 5 first-order formulas)

\[ x = N \land y = 0 \land N > 0 \land \pi = l_1 \rightarrow x \leq N \]
\[ T_1 : x \leq N \land \ldots \land x' = x \land \ldots \rightarrow x' \leq N \]
\[ T_2 : x \leq N \land \ldots \land x' = x - 1 \land \ldots \rightarrow x' \leq N \]
\[ T_3 : x \leq N \land \ldots \land x' = x \land \ldots \rightarrow x' \leq N \]
\[ T_4 : x \leq N \land \ldots \land x' = x \land \ldots \rightarrow x' \leq N \]

\[ x \geq 0 \quad \text{valid} \]

\[ \Theta \rightarrow p \]

invariant to prove:

\[ \text{is an invariant for all values of } N > 0 \]
Proving invariance properties deductively: example

\( \mathcal{J} = \{ T_1, T_2, T_3, T_4 \} \) with

- \( \rho_1 : \pi = l_1 \land ( ( x > 0 \land \pi' = l_2 ) \lor ( x \leq 0 \land \pi' = l_4 ) ) \land \text{pres}(\{ x, y \}) \)
- \( \rho_2 : \pi = l_2 \land \pi' = l_3 \land x' = x - 1 \land y' = y \)
- \( \rho_3 : \pi = l_3 \land \pi' = l_1 \land y' = y + x \land x' = x \)
- \( \rho_4 : \text{pres}(\{ x, y, \pi \}) \)

- \( p \) is preserved by every transition \( \tau \)
  (inductive step)

- \( T_1 : x \geq 0 \land \ldots \land x' = x \land \ldots \rightarrow x' \geq 0 \)
  valid

- \( T_2 : x \geq 0 \land \ldots \land x' = x - 1 \land \ldots \rightarrow x' \geq 0 \)
  not valid

- \( T_3 : x \geq 0 \land \ldots \land x' = x \land \ldots \rightarrow x' \geq 0 \)
  valid

- \( T_4 : x \geq 0 \land \ldots \land x' = x \land \ldots \rightarrow x' \geq 0 \)
  valid

what is the problem?

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

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

\[ \Sigma \]

\[ \Sigma^{-} \]

what is the problem?

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

- \( T_2 : x \geq 0 \land \ldots \land x' = x - 1 \land \ldots \rightarrow x' \geq 0 \)

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

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

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

remove them by strengthening the inductive hypothesis

\[x \geq 0 \land ((\pi = l_2) \rightarrow x > 0)\]

what is the problem?

Transition \(\tau_2\) is preserved

\[\tau_2 : x \geq 0 \land ((\pi = l_2) \rightarrow x > 0) \land \pi = l_2 \land x' = x - 1 \land \pi' = l_3 \rightarrow x' \geq 0 \land ((\pi' = l_2) \rightarrow x' > 0)\]

How about the initial condition?

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

\[\Theta \rightarrow p\]

\[x = N \land y = 0 \land N > 0 \land \pi = l_1 \rightarrow x \geq 0 \land ((\pi = l_2) \rightarrow x > 0)\]

still valid
How about the other transitions?

\[ \mathcal{J} = \{ \tau_1, \tau_2, \tau_3, \tau_4 \} \]

\[ \rho_{\tau_1} : \pi = l_1 \land ( ( x > 0 \land \pi' = l_2 ) \land \text{pres}( \{ x, y \}) ) \]

\[ \rho_{\tau_2} : \pi = l_2 \land \pi' = l_3 \land x \land y' = y \]

\[ \rho_{\tau_3} : \pi = l_3 \land \pi' = l_1 \land y' = y + x \land x' = x \]

\[ \rho_{\tau_4} : \text{pres}( \{ x, y, \pi \}) \]

\[ p \text{ is preserved by every transition } \tau \]

( inductive step )

\[ \tau_1 : x \geq 0 \land (( \pi = l_2 ) \rightarrow x > 0) \land \]

\[ ( x > 0 \land \pi' = l_2 ) \land \ldots \land x' = x \land \ldots \]

\[ \rightarrow x' \geq 0 \land (( \pi' = l_2 ) \rightarrow x > 0) \]

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

\[ x \geq 0 \land (( \pi = l_2 ) \rightarrow x > 0) \]

(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 \( p \)

\[ \begin{align*}
\text{B1.} & \quad \emptyset \rightarrow p \\
\text{B2.} & \quad p \land \rho_\tau \rightarrow p' \quad \text{for all } \tau \text{ in } \mathcal{J} \\
\end{align*} \]

\[ \square p \quad ( p \text{ is an invariant of } \Phi ) \]

B-INV reduces the proof of an invariant to checking the validity of

\( | \mathcal{J} | + 1 \) first-order formulas ( verification conditions in the underlying
assertion language ).

Verification rule G-INV (general invariance)

For assertions \( \varphi, p \)

\[ \begin{align*}
\text{I1.} & \quad \varphi \rightarrow p \\
\text{I2.} & \quad \emptyset \rightarrow \varphi \\
\text{I3.} & \quad \varphi \land \rho_\tau \rightarrow \varphi' \quad \text{for all } \tau \text{ in } \mathcal{J} \\
\end{align*} \]

\[ \square p \quad ( p \text{ is an invariant of } \Phi ) \]

G-INV reduces the proof of an invariant \( p \) to finding an inductive
assertion \( \varphi \) that strengthens \( p \) and to checking the validity of

\( | \mathcal{J} | + 2 \) first-order formulas ( verification conditions in the underlying
assertion language ).
Verifications rule G-INV (general invariance)

For assertions \( \varphi, p \)

1. \( \varphi \rightarrow p \)
2. \( \emptyset \rightarrow \varphi \)
3. \( \varphi \land p^\tau \rightarrow \varphi' \) for all \( \tau \) in \( \mathcal{J} \)

\[ \square p \quad (p \text{ is an invariant of } \Phi) \]

G-INV reduces the proof of an invariant \( p \) to finding an inductive assertion \( \varphi \) that strengthens \( p \) and to checking the validity of

\[ | \mathcal{J} | + 2 \] first-order formulas (verification conditions in the underlying assertion language).

The Big Question

How do we find \( \varphi \) ?

40 years of research has not answered this question.
Static analysis

Incremental analysis:
- generate many simple q's that are inductive

Deep analysis:
- generate interesting invariants

Verification rule G-INV (general invariance)

For assertions ϕ, p
1. ϕ → p
2. Θ → ϕ
3. ϕ ∧ ρτ → ϕ' for all τ in J

□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:

Verification rule INC-INV (incremental invariance)

For assertion p, q1 .... qn
B0. □ q1 ...... □ qn
B1. Θ → p
B2. p ∧ q1 ∧ ...... ∧ qn ∧ ρτ → p' for all τ in J

□p (p is an invariant of Φ)

Semantics
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 $x_i$, 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: $x_i = 0$

Original description:


Subsequently analyzed for possibility of deadlocks:


Manufacturing system example: our analysis

Described in:

Some results:
- generated 1900 invariants
- invariants imply absence of deadlock for initial values $1 \leq x_i \leq 8$
- 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 $M_4$ to $M_6$ 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

```plaintext
local x,y: integer where x = N \land y=0 \land 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$