On Compositional safety verification with Max-SMT
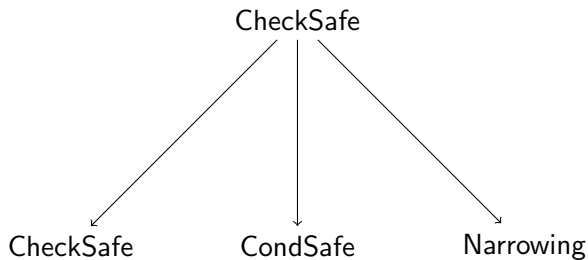
Fabian Böller with David Korzeniewski

RWTH Aachen

fabian.boeller@rwth-aachen.de

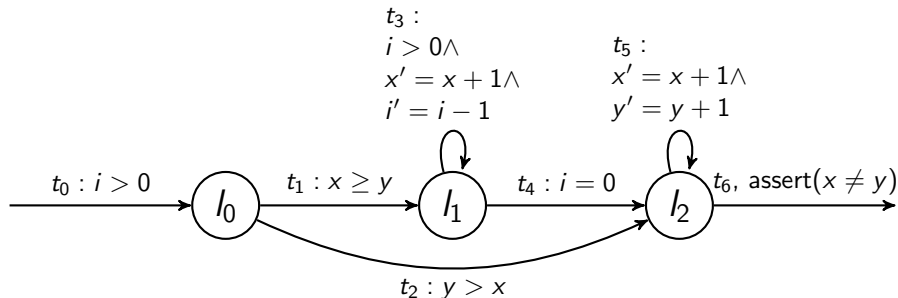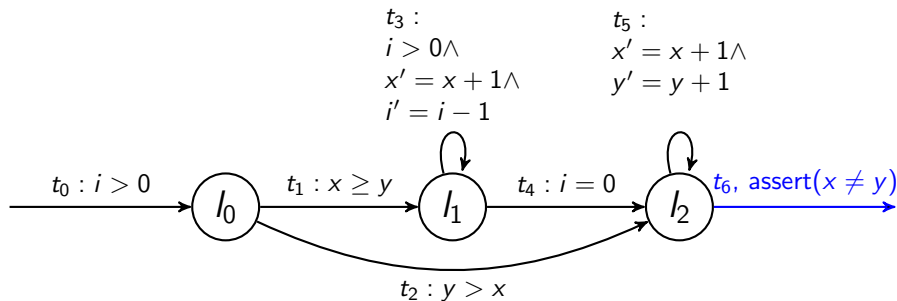SS 2017

Overview

Algorithms

Example program

Program

$\mathcal{V} = \{x, y, i\}$, $\mathcal{L} = \{\ell_0, \ell_1, \ell_2\}$, $\mathcal{T} = \{t_i \mid i \in \{1, \ldots, 6\}\}$

$$t_3 : \\ i > 0 \wedge \\ x' = x + 1 \wedge \\ i' = i - 1$$

$$t_5 : \\ x' = x + 1 \wedge \\ y' = y + 1$$

$t_0 : i > 0 \qquad l_0 \qquad t_1 : x \geq y \qquad l_1 \qquad t_4 : i = 0 \qquad l_2 \qquad t_6, \text{assert}(x \neq y)$

$t_2 : y > x$

Example execution

Task

Prove that the program is safe for $x \neq y$ at t_6

Example execution

$$t_3 : \\ i > 0 \wedge \\ x' = x + 1 \wedge \\ i' = i - 1$$

$$t_5 : \\ x' = x + 1 \wedge \\ y' = y + 1$$

$t_0 : i > 0$

$t_1 : x \geq y$

$t_4 : i = 0$

$t_6, \text{assert}(x \neq y)$
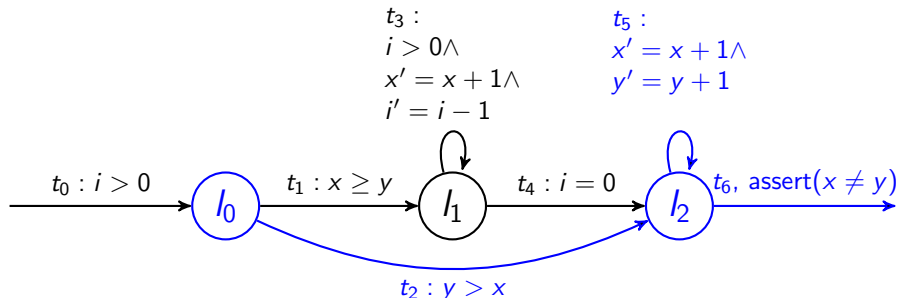
$t_2 : y > x$

l_0 l_1 l_2
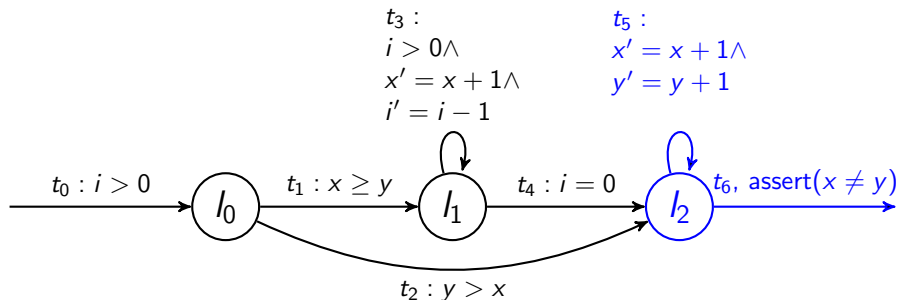
CheckSafe on $\{\ell_2\}$ for $x \neq y$

t_6 does not already imply $x \neq y$

t_6 is not an initial transition
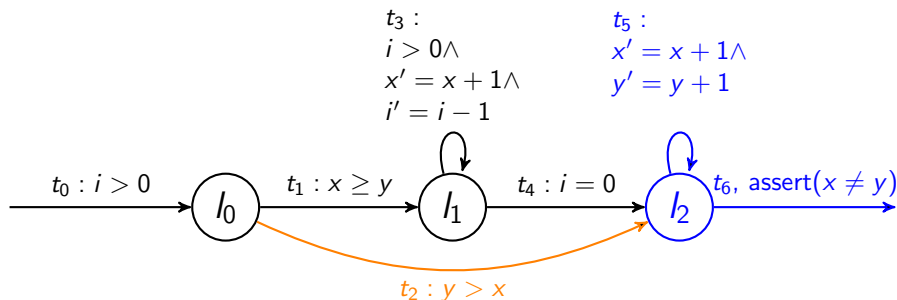
Call CondSafe, get $x > y$ as precondition

CheckSafe on $\{\ell_0\}$ for $x > y$

t_2 does not already imply $x > y$

t_2 is not an initial transition

Call CondSafe

Example execution

$t_3:$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5:$
$x' = x + 1 \wedge$
$y' = y + 1$

$t_0 : i > 0$

$t_1 : x \geq y$

l_0

l_1

$t_4 : i = 0$

l_2

$t_6, \text{assert}(x \neq y)$

$t_2 : y > x$

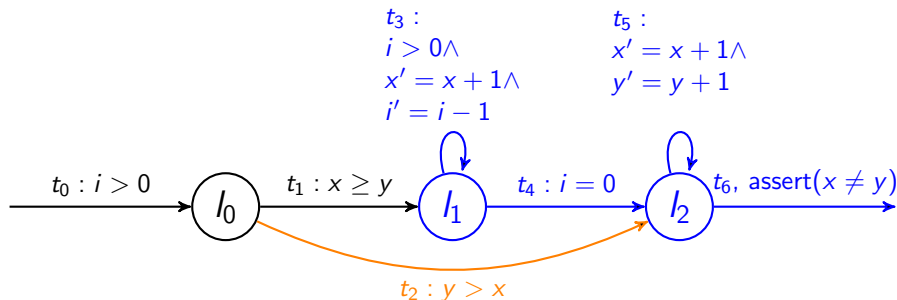## CheckSafe on $\{\ell_0\}$ for $x > y$

No precondition, since $y > x$ contradicts $x > y$

Path is maybe safe, but not for $x > y$

Example execution

$t_3:$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5:$
$x' = x + 1 \wedge$
$y' = y + 1$

$t_0 : i > 0$ l_0 $t_1 : x \geq y$ l_1 $t_4 : i = 0$ l_2 t_6, assert$(x \neq y)$
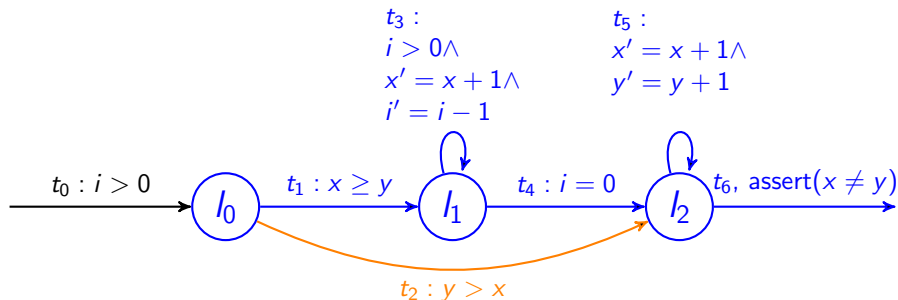
$t_2 : y > x$

CheckSafe on $\{\ell_1\}$ for $x > y$

t_4 does not already imply $x > y$

t_4 is not an initial transition

Call CondSafe, get $i > 0 \wedge x \geq y$ as precondition

Example execution

t_3 :
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

t_5 :
$x' = x + 1 \wedge$
$y' = y + 1$

$t_0 : i > 0$

$t_1 : x \geq y$

$t_4 : i = 0$

t_6, assert$(x \neq y)$
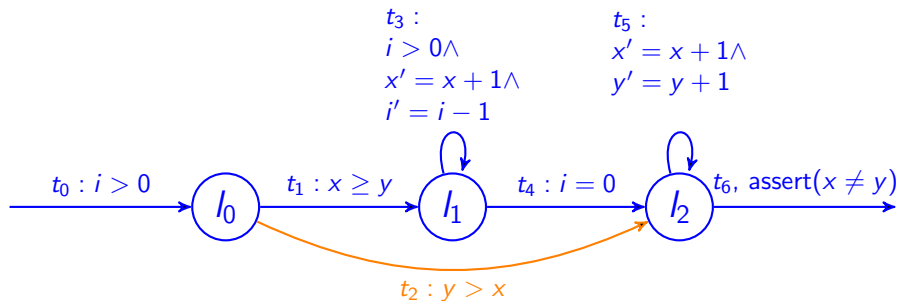
$t_2 : y > x$

CheckSafe on $\{\ell_0\}$ for $i > 0$

t_1 does not already imply $i > 0$

t_1 is not an initial transition

Call CondSafe, get $i > 0$ as precondition

Example execution

t_3 :
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

t_5 :
$x' = x + 1 \wedge$
$y' = y + 1$

$t_0 : i > 0$

$t_1 : x \geq y$

$t_4 : i = 0$

t_6, assert($x \neq y$)

l_0 l_1 l_2

$t_2 : y > x$

CheckSafe on initial SCC for $i > 0$

t_0 does already imply $i > 0$

Example execution
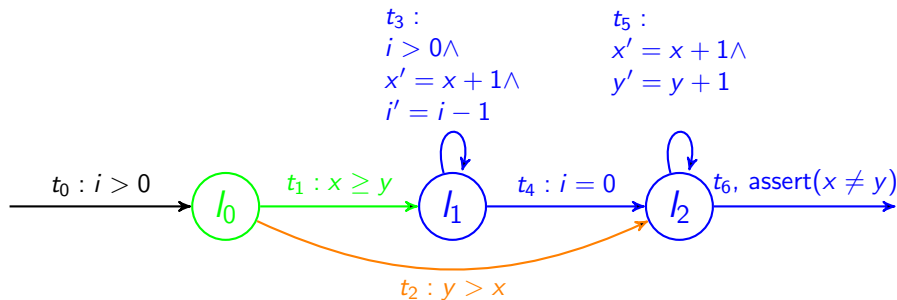
Example execution

CheckSafe on $\{\ell_0\}$ for $i > 0$
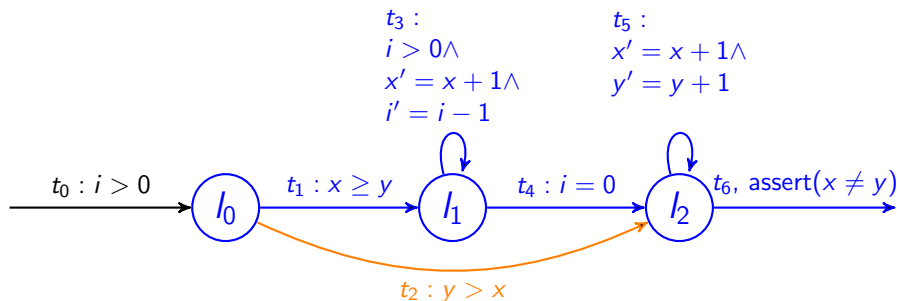
Path is safe for $i > 0$

CheckSafe on $\{\ell_0\}$ for $x \geq y$

t_1 does already imply $x \geq y$

$t_3:$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5:$
$x' = x + 1 \wedge$
$y' = y + 1$

$t_0 : i > 0$

l_0

$t_1 : x \geq y$

l_1

$t_4 : i = 0$

l_2

$t_6, \text{assert}(x \neq y)$

$t_2 : y > x$

CheckSafe on $\{\ell_0\}$ for $x \geq y$

Path is safe for $x \geq y$

Example execution

$t_3:$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5:$
$x' = x + 1 \wedge$
$y' = y + 1$

$t_0 : i > 0$ $t_1 : x \geq y$ l_0 l_1 $t_4 : i = 0$ l_2 t_6, assert$(x \neq y)$

$t_2 : y > x$

CheckSafe on $\{\ell_1\}$ for $x > y$

Path is safe for $x > y$

Example execution

$t_3 :$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5 :$
$x' = x + 1 \wedge$
$y' = y + 1$
$\wedge \mathbf{x} \leq \mathbf{y}$

$t_0 : i > 0$

$t_1 : x \geq y$

$t_4 : i = 0$

$t_6, \text{assert}(x \neq y)$

l_0 l_1 l_2

$t_2 : y > x$
$\wedge \mathbf{x} \leq \mathbf{y}$
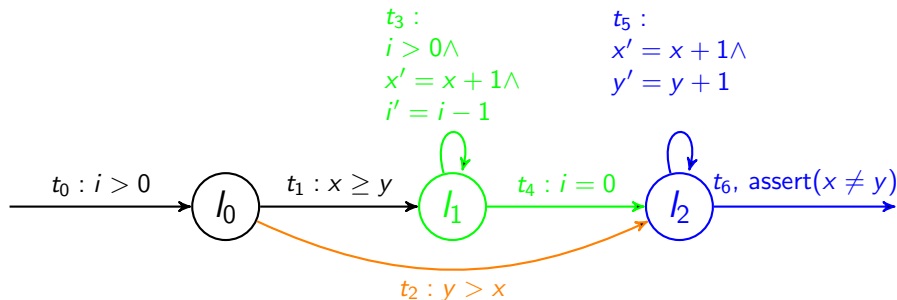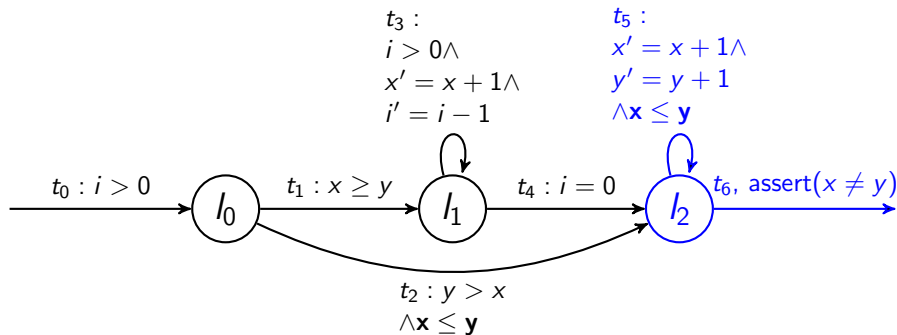
Narrow on $\{\ell_2\}$

Add $x \leq y$ to t_2

Add $x \leq y$ to t_5

Example execution

$t_3 :$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5 :$
$x' = x + 1 \wedge$
$y' = y + 1$
$\wedge \mathbf{x} \leq \mathbf{y}$

$t_0 : i > 0$ $t_1 : x \geq y$ $t_4 : i = 0$ l_2 $t_6, \text{assert}(x \neq y)$

l_0 l_1

$t_2 : y > x$
$\wedge \mathbf{x} \leq \mathbf{y}$
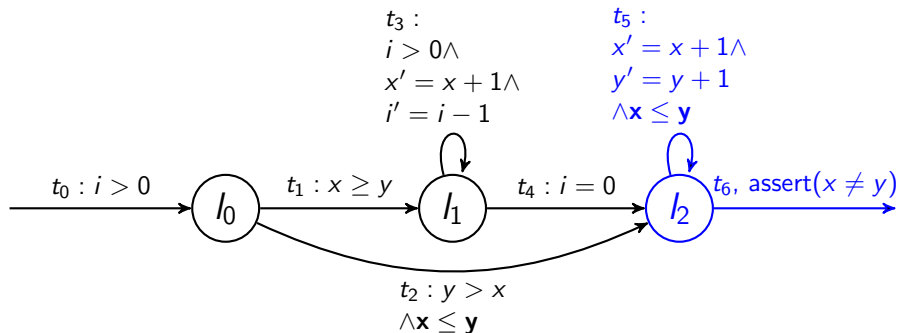
CheckSafe on $\{\ell_2\}$ for $x \neq y$

Call CondSafe, get $y > x$ instead of $x > y$ as precondition

t_3 :
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

t_5 :
$x' = x + 1 \wedge$
$y' = y + 1$
$\wedge \mathbf{x} \leq \mathbf{y}$

$t_0 : i > 0$

$t_1 : x \geq y$

$t_4 : i = 0$

t_6, assert($x \neq y$)

l_0

l_1

l_2

$t_2 : y > x$
$\wedge \mathbf{x} \leq \mathbf{y}$
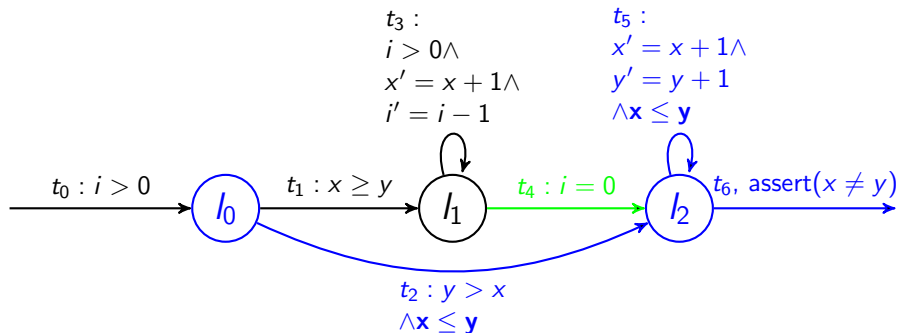
CheckSafe on $\{\ell_0\}$ for $y > x$

t_2 does already imply $y > x$

$t_3:$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5:$
$x' = x + 1 \wedge$
$y' = y + 1$
$\wedge \mathbf{x} \leq \mathbf{y}$

$t_0 : i > 0$ l_0 $t_1 : x \geq y$ l_1 $t_4 : i = 0$ l_2 $t_6, \text{assert}(x \neq y)$

$t_2 : y > x$
$\wedge \mathbf{x} \leq \mathbf{y}$
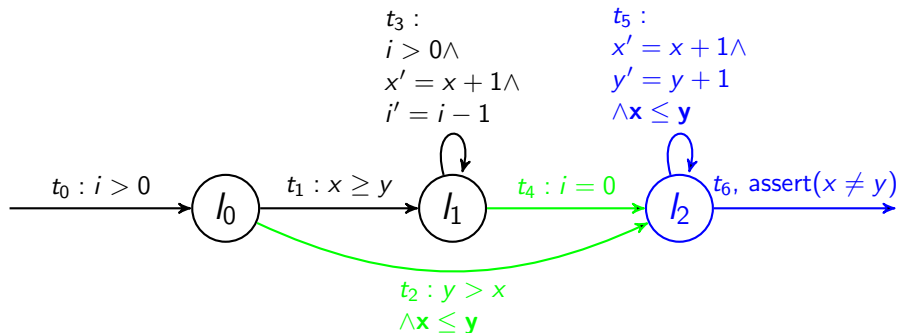
CheckSafe on $\{\ell_0\}$ for $y > x$

Path is safe for $y > x$

$t_3:$
$i > 0 \wedge$
$x' = x + 1 \wedge$
$i' = i - 1$

$t_5:$
$x' = x + 1 \wedge$
$y' = y + 1$
$\wedge \mathbf{x} \leq \mathbf{y}$

$t_0 : i > 0$

$t_1 : x \geq y$

$t_4 : i = 0$

$t_6, \text{assert}(x \neq y)$

l_0

l_1

l_2

$t_2 : y > x$
$\wedge \mathbf{x} \leq \mathbf{y}$
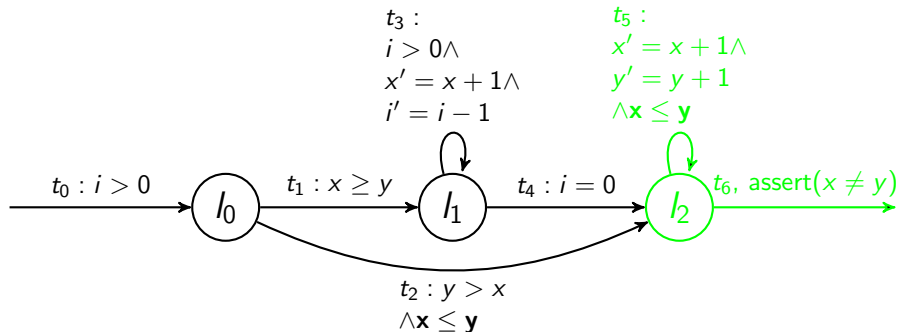
CheckSafe on $\{\ell_0\}$ for $y > x$

Program is safe for $x \neq y$

References

📄 Brockschmidt, Marc and Larraz, Daniel and Oliveras, Albert and Rodriguez-Carbonell, Enric and Rubio, Albert (2015)
Compositional Safety Verification with Max-SMT
Proceedings of FMCAD'15

The End