# Verifikacija liveness svojstava programa

Matko Botinčan[1]

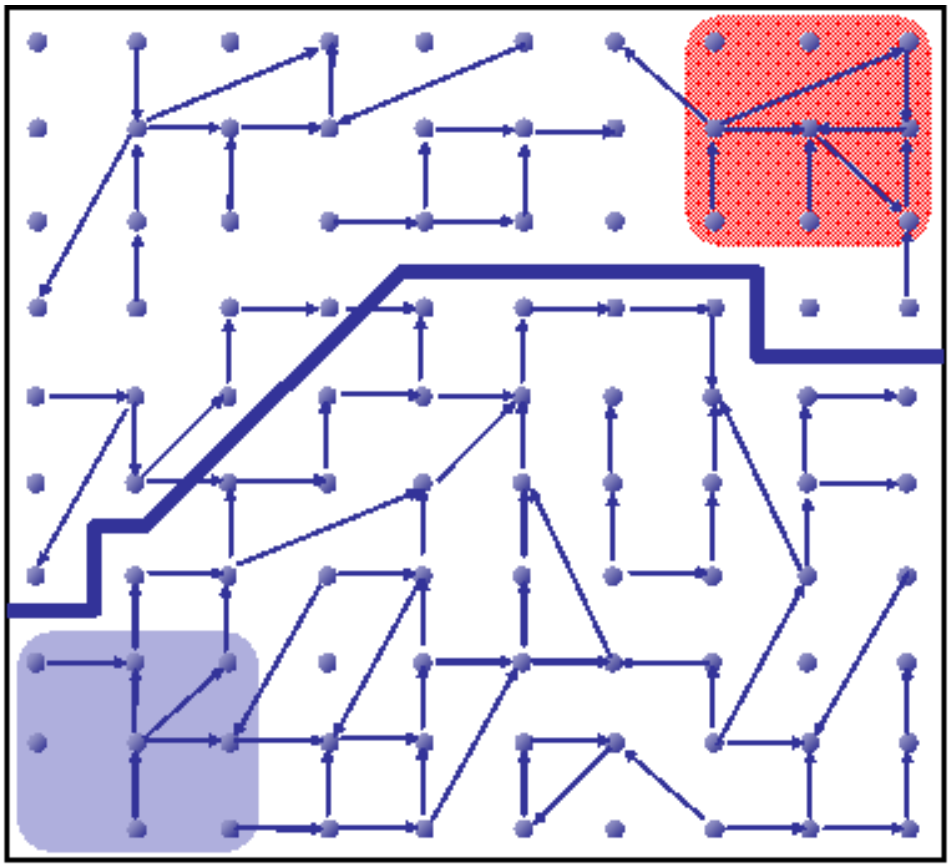
[1]PMF - Matematički odjel

Seminar za teorijsko računarstvo, 07.12.2006.

# Sadržaj

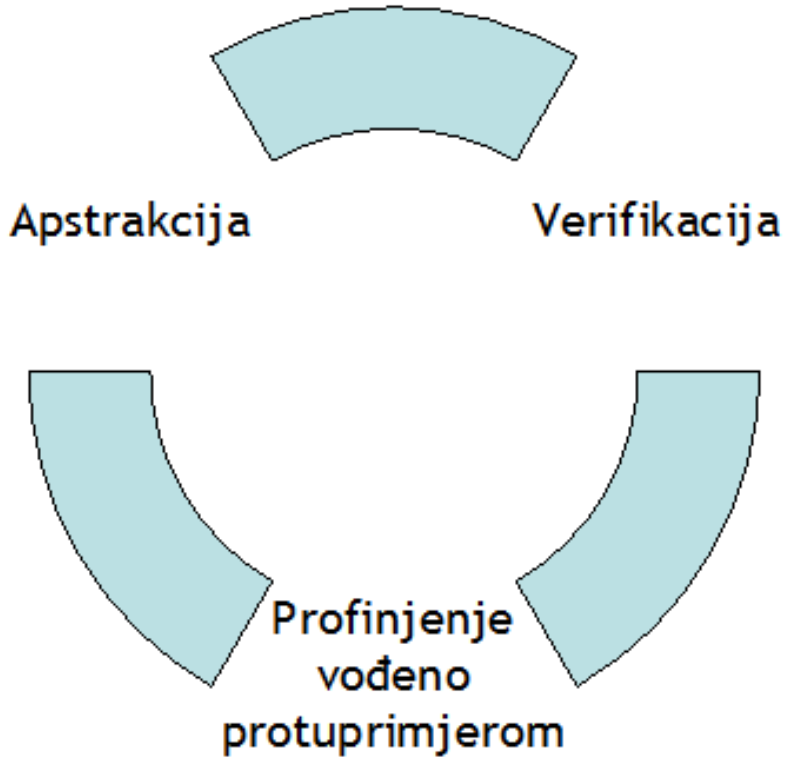
# Pregled

# Model checking safety svojstava

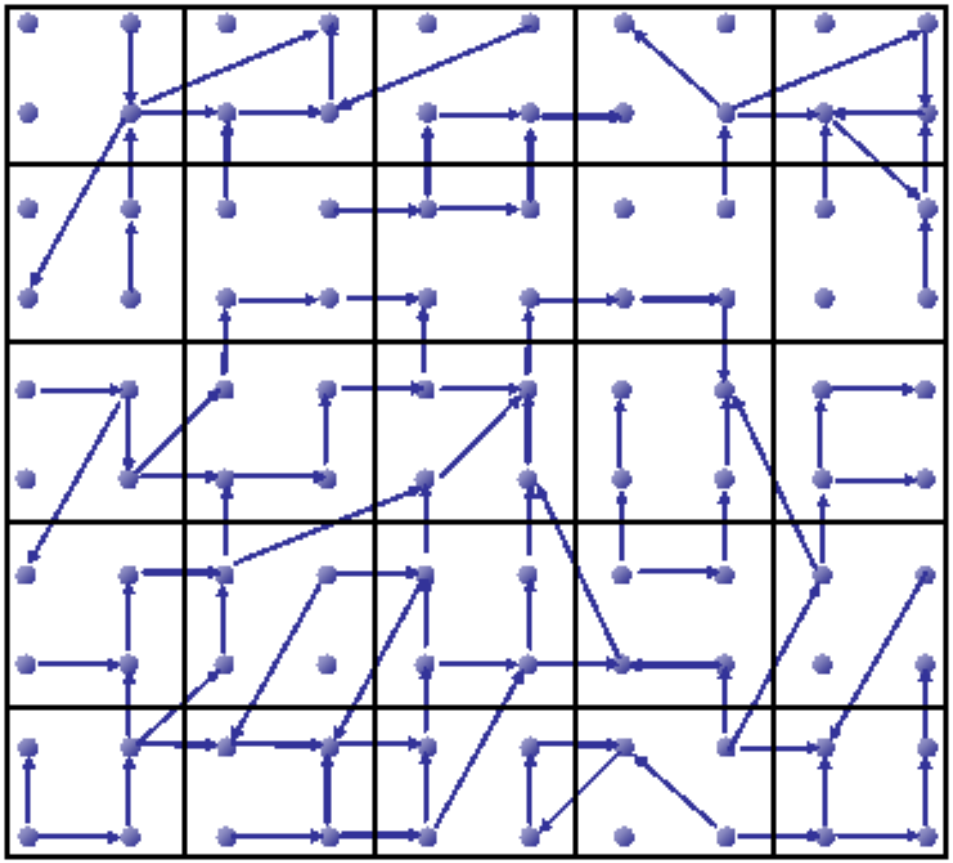


Izvor ilustracija: SPIN 2005 tutorial – Software Verification with BLAST
(T. Henzinger, R. Jhala, R. Majumdar)

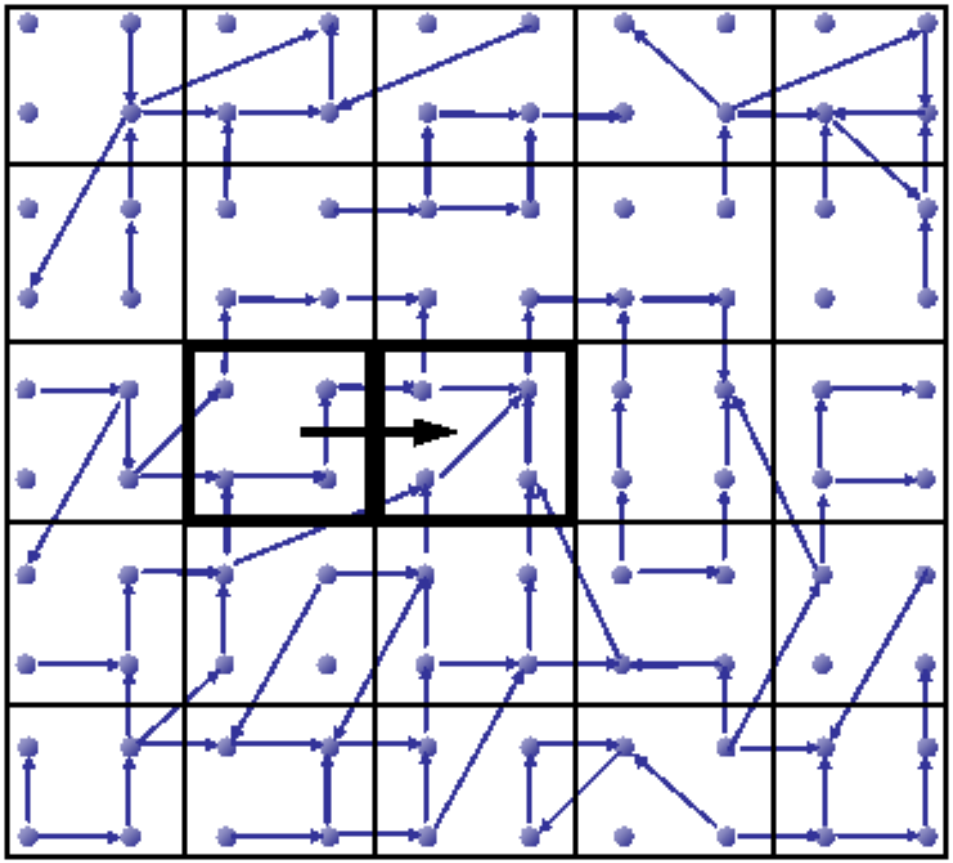
# "Software Model Checking via Counterexample-driven Abstraction Refinement"



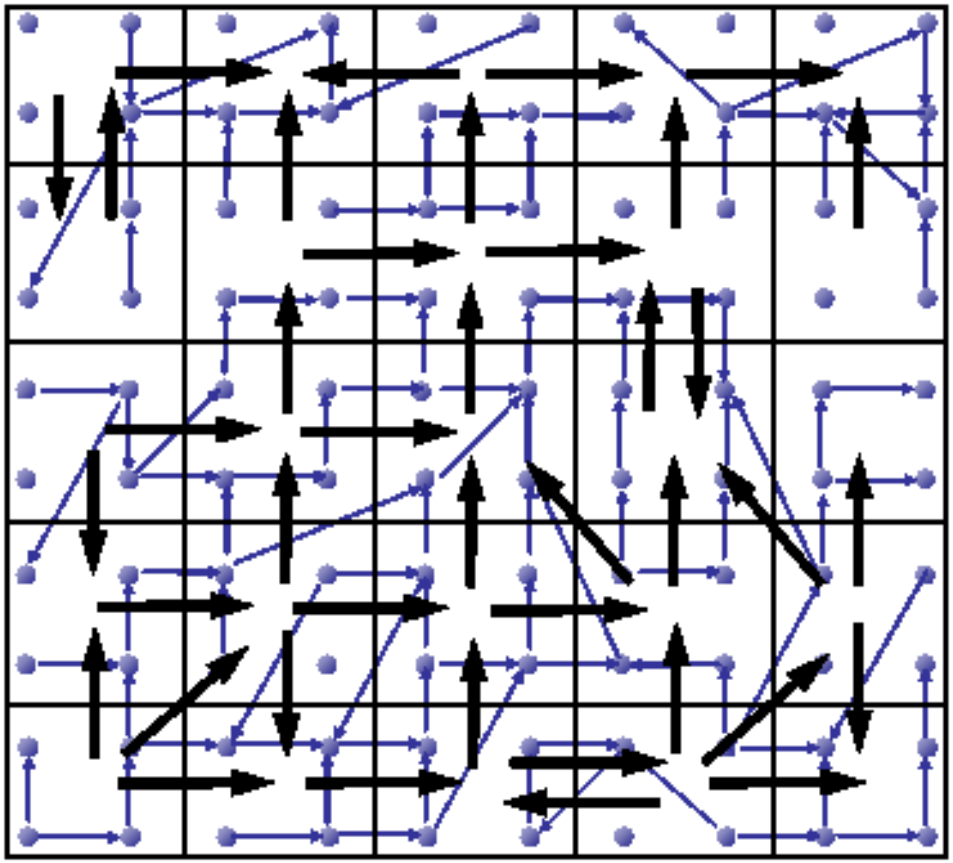
# Apstrakcija



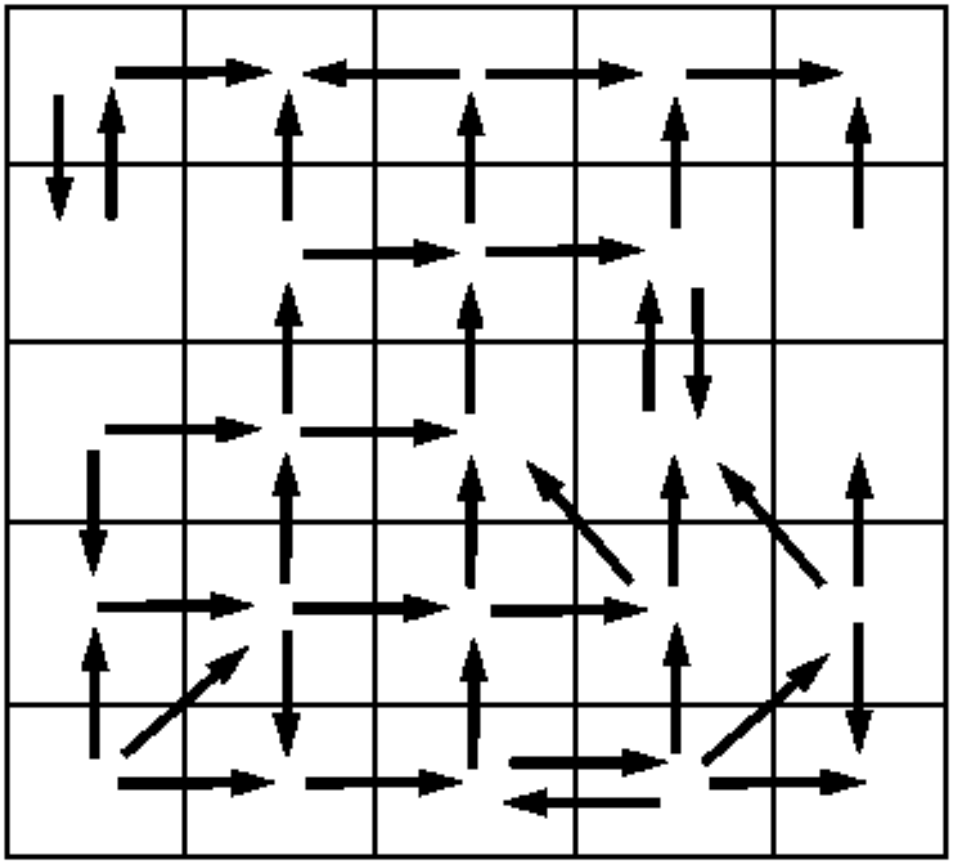
# Apstrakcija



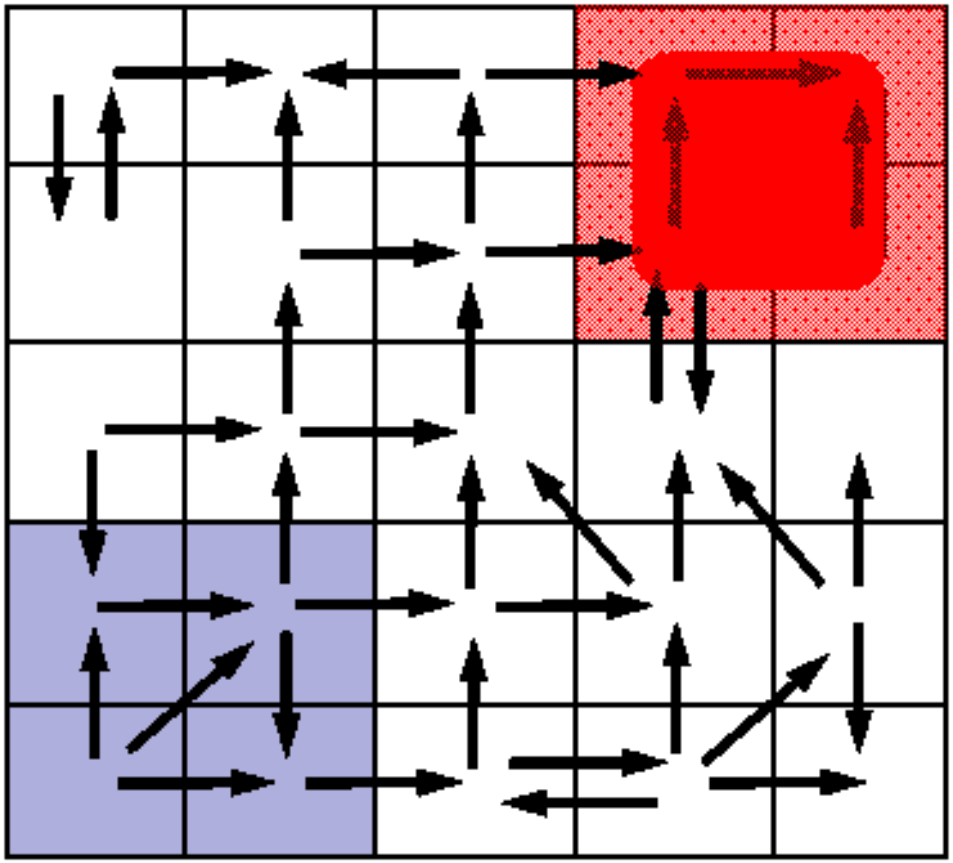
# Apstrakcija



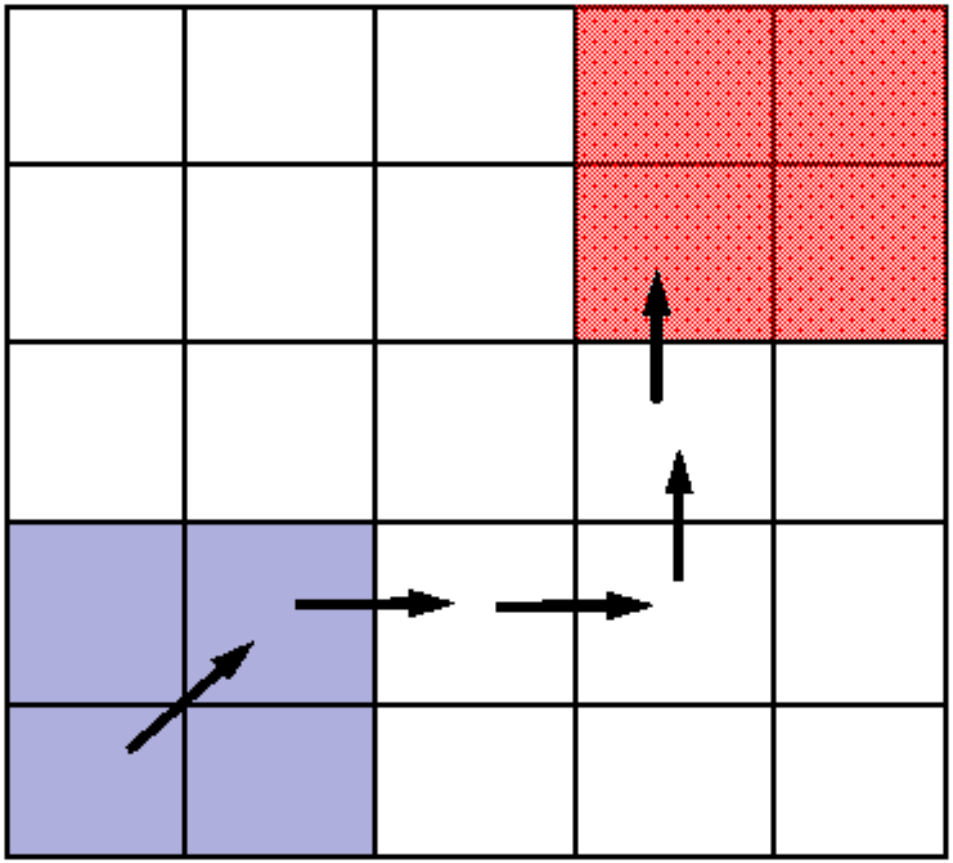
# Apstrakcija



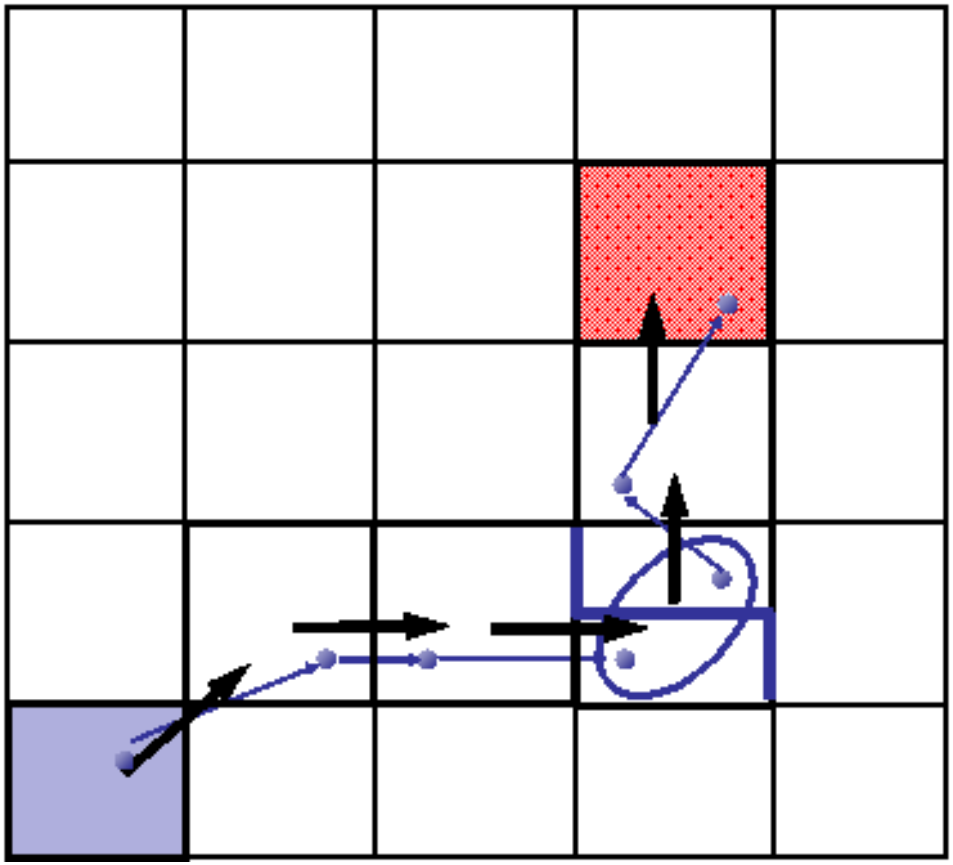
# Problem: lažni kontraprimjeri



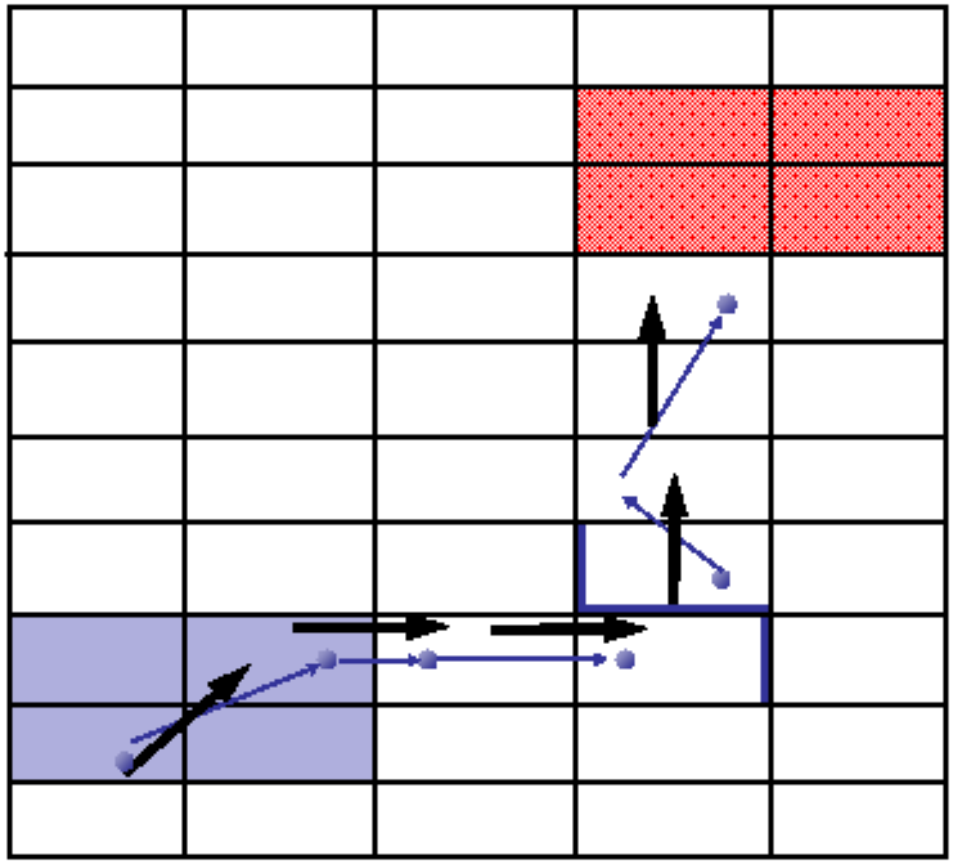
# Profinjenje vođeno protuprimjerom



# Profinjenje vođeno protuprimjerom



# Profinjenje vođeno protuprimjerom



# Apstrakcija ne čuva liveness svojstva

Zašto apstrakciju ne možemo (barem ne na isti način) iskoristiti za verifikaciju liveness svojstava?

# Apstrakcija ne čuva liveness svojstva

Zašto apstrakciju ne možemo (barem ne na isti način) iskoristiti za verifikaciju liveness svojstava?

Primjer: Svojstvo terminacije

# Apstrakcija ne čuva liveness svojstva

Zašto apstrakciju ne možemo (barem ne na isti način) iskoristiti za verifikaciju liveness svojstava?

Primjer: Svojstvo terminacije

Tranzicijski sustav $S$ s konačno mnogo stanja ima svojstvo terminacije $\Leftrightarrow$ ne postoji trag u $S$ koji sadrži petlju.

# Apstrakcija ne čuva liveness svojstva

Zašto apstrakciju ne možemo (barem ne na isti način) iskoristiti za verifikaciju liveness svojstava?

Primjer: Svojstvo terminacije

Tranzicijski sustav $S$ s konačno mnogo stanja ima svojstvo terminacije $\Leftrightarrow$ ne postoji trag u $S$ koji sadrži petlju.

No, kako onda apstrahirati npr:

```
while (i > 0)
{
    i = i - 1;
}
```

# Software model checking – state of the art

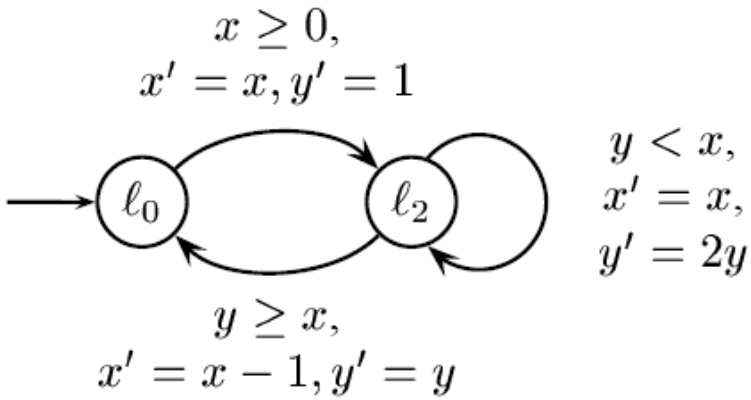
| Alati | Model checking | Apstrakcija | Svojstva |
|---|---|---|---|
| SLAM, BLAST | simbolički | automatska | safety |
| Bandera, (stari) JPF | explicit-state | manualna | safety & LTL |
| Terminator | simbolički | automatska | liveness / LTL |

# Pregled

# Primjer 1

$$\begin{array}{ll} \textbf{in } n & : \textbf{integer where } n > 0 \\ \textbf{local } x, y & : \textbf{integer where } x = n \end{array}$$

$$\left[\begin{array}{l} \ell_0 : \textbf{while } x \geq 0 \textbf{ do} \\ \quad \left[\begin{array}{l} \ell_1 : y := 1 \\ \ell_2 : \textbf{while } y < x \textbf{ do} \\ \quad\quad \ell_3 : y := 2y \\ \ell_4 : x := x - 1 \end{array}\right] \end{array}\right]$$

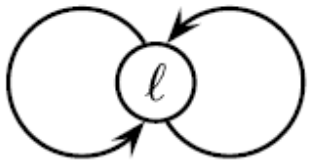


# Primjer 2

$$\textbf{local } x, y \qquad : \textbf{integer}$$

$$\left[\begin{array}{l} \textbf{while } x > 0 \wedge y > 0 \textbf{ do} \\ \quad \left[\begin{array}{l} \ell^a : (x, y) := (x - 1, x) \\ \quad \textbf{or} \\ \ell^b : (x, y) := (y - 2, x + 1) \end{array}\right] \end{array}\right]$$

$$x > 0, y > 0, \quad x' = x - 1, \quad y' = x$$




$$x > 0, y > 0, \quad x' = y - 2, \quad y' = x + 1$$

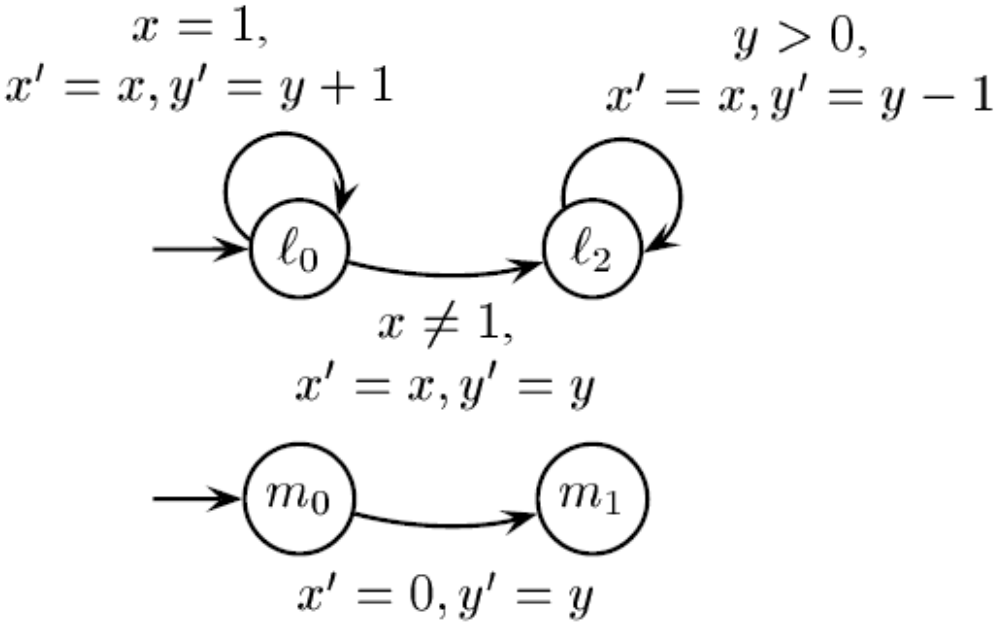
Primjeri preuzeti iz:

A. Podelski, A. Rybalchenko: Transition invariants. LICS'04.

# Primjer 3

$$\textbf{local } x, y \qquad : \textbf{integer where } x = 1, y = 0$$

$$P_1 :: \left[\begin{array}{c} \ell_0 : \textbf{while } x = 1 \textbf{ do} \\ \quad \ell_1 : y := y + 1 \\ \ell_2 : \textbf{while } y > 0 \textbf{ do} \\ \quad \ell_3 : y := y - 1 \end{array}\right] \parallel P_2 :: \left[\begin{array}{l} m_0 : x := 0 \\ m_1 : \end{array}\right]$$

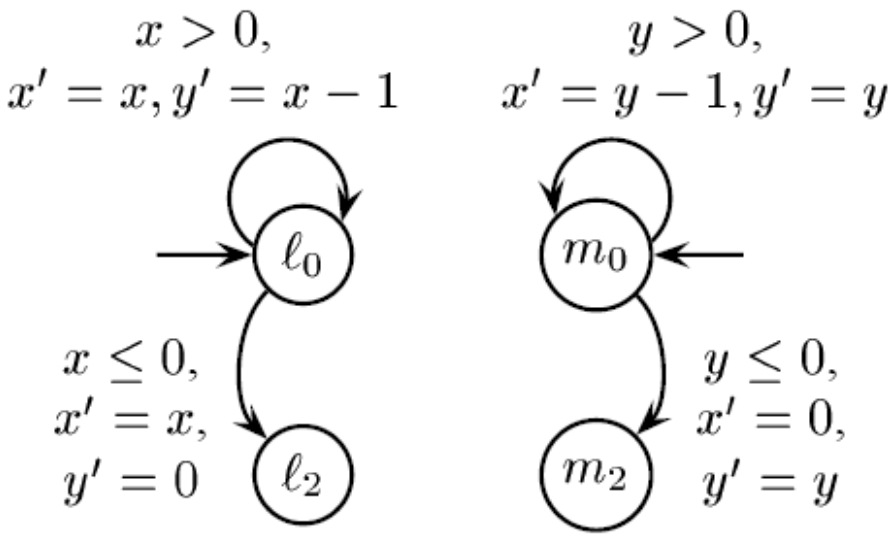


# Primjer 4

$$\textbf{local } x, y \qquad : \textbf{integer where } x > 0, y > 0$$

$$P_1 :: \left[ \begin{array}{l} \ell_0 : \textbf{while } x > 0 \textbf{ do} \\ \qquad \ell_1 : y := x - 1 \\ \ell_2 : y := 0 \end{array} \right]$$

$$\| \quad P_2 :: \left[ \begin{array}{l} m_0 : \textbf{while } y > 0 \textbf{ do} \\ \qquad m_1 : x := y - 1 \\ m_2 : x := 0 \end{array} \right]$$



# Tranzicijska invarijanta

Program $P = (W, I, R)$ sastoji se od:

- $W$ — skup stanja
- $I \subseteq W$ — skup početnih stanja
- $R \subseteq W \times W$ — tranzicijska relacija

# Tranzicijska invarijanta

Program $P = (W, I, R)$ sastoji se od:

- $W$ — skup stanja
- $I \subseteq W$ — skup početnih stanja
- $R \subseteq W \times W$ — tranzicijska relacija

Označimo

$$\begin{aligned} R_I^+ &:= R^+ \cap (Reach \times Reach) \\ &= \{(s_1, s_2) \mid \exists s_0 \in I \,.\, (s_0, s_1) \in R^* \wedge (s_1, s_2) \in R^+\} \end{aligned}$$

# Tranzicijska invarijanta

Program $P = (W, I, R)$ sastoji se od:

- $W$ — skup stanja
- $I \subseteq W$ — skup početnih stanja
- $R \subseteq W \times W$ — tranzicijska relacija

Označimo

$$\begin{aligned} R_I^+ &:= R^+ \cap (Reach \times Reach) \\ &= \{(s_1, s_2) \mid \exists s_0 \in I \,.\, (s_0, s_1) \in R^* \wedge (s_1, s_2) \in R^+\} \end{aligned}$$

Tranzicijska invarijanta je relacija $T$ za koju vrijedi:

$$R_I^+ \subseteq T$$

$\rightarrow$ invarijanta stanja: $I \cup \{s' \mid s \in I \wedge (s, s') \in T\}$

# Terminacija

Kažemo da je relacija $T$ disjunktivno dobro utemeljena ako je $T = T_1 \cup \ldots \cup T_n$, gdje su sve $T_i$ dobro utemeljene relacije.

# Terminacija

Kažemo da je relacija $T$ disjunktivno dobro utemeljena ako je $T = T_1 \cup \ldots \cup T_n$, gdje su sve $T_i$ dobro utemeljene relacije.

Kažemo da program $P$ terminira ukoliko ne postoji beskonačno izračunavanje, tj. ne postoji niz $(s_i)_{i \in \mathbb{N}}$ t.d. $s_0 \in I$ i $\forall i > 0 \,.\, (s_i, s_{i+1}) \in R$.

# Terminacija

Kažemo da je relacija $T$ disjunktivno dobro utemeljena ako je $T = T_1 \cup \ldots \cup T_n$, gdje su sve $T_i$ dobro utemeljene relacije.

Kažemo da program $P$ terminira ukoliko ne postoji beskonačno izračunavanje, tj. ne postoji niz $(s_i)_{i \in \mathbb{N}}$ t.d. $s_0 \in I$ i $\forall i > 0 \,.\, (s_i, s_{i+1}) \in R$.

**Teorem**

*$P$ terminira ako i samo ako postoji disjunktivno dobro utemeljena tranzicijska invarijanta za $P$.*

# Terminacija

## Dokaz.

# Terminacija

## Dokaz.

$\boxed{\Leftarrow}$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.

# Terminacija

## Dokaz.

$\Leftarrow$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$

# Terminacija

## Dokaz.

$\Leftarrow$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k, l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.

# Terminacija

## Dokaz.

$\Leftarrow$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k,l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.
Def. rel. ekvivalencije na $\mathbb{N} \times \mathbb{N}$: $(k,l) \sim (k',l') \Leftrightarrow f(k,l) = f(k',l')$.
$\rightarrow \sim$ ima konačan indeks.

# Terminacija

## Dokaz.

$\Leftarrow$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k,l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.
Def. rel. ekvivalencije na $\mathbb{N} \times \mathbb{N}$: $(k,l) \sim (k',l') \Leftrightarrow f(k,l) = f(k',l')$.
$\rightarrow \sim$ ima konačan indeks.
Ramseyev tm $\Rightarrow$ postoji $\infty$ $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$ t.d. $\forall k < l$ vrijedi $(k,l) \in [(m,n)]_\sim$ (za neke fiksne $m$ i $n$).

# Terminacija

## Dokaz.

$\Leftarrow$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k,l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.
Def. rel. ekvivalencije na $\mathbb{N} \times \mathbb{N}$: $(k,l) \sim (k',l') \Leftrightarrow f(k,l) = f(k',l')$.
$\rightarrow \sim$ ima konačan indeks.
Ramseyev tm $\Rightarrow$ postoji $\infty$ $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$ t.d. $\forall k < l$ vrijedi $(k,l) \in [(m,n)]_\sim$ (za neke fiksne $m$ i $n$).
$\rightarrow \forall i \,.\, (s_{k_i}, s_{k_{i+1}}) \in f(m,n) \rightarrow f(m,n)$ nije dobro utemeljena. $\Rightarrow\Leftarrow$

# Terminacija

## Dokaz.

$\boxed{\Leftarrow}$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_I^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k,l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.
Def. rel. ekvivalencije na $\mathbb{N} \times \mathbb{N}$: $(k,l) \sim (k',l') \Leftrightarrow f(k,l) = f(k',l')$.
$\rightarrow \sim$ ima konačan indeks.
Ramseyev tm $\Rightarrow$ postoji $\infty$ $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$ t.d. $\forall k < l$ vrijedi $(k,l) \in [(m,n)]_\sim$ (za neke fiksne $m$ i $n$).
$\rightarrow \forall i \,.\, (s_{k_i}, s_{k_{i+1}}) \in f(m,n) \rightarrow f(m,n)$ nije dobro utemeljena. $\Rightarrow\Leftarrow$

$\boxed{\Rightarrow}$ Neka $P$ terminira. Definiramo $T := R_I^+$.

# Terminacija

## Dokaz.

$\Leftarrow$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_l^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k,l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.
Def. rel. ekvivalencije na $\mathbb{N} \times \mathbb{N}$: $(k,l) \sim (k',l') \Leftrightarrow f(k,l) = f(k',l')$.
$\rightarrow \sim$ ima konačan indeks.
Ramseyev tm $\Rightarrow$ postoji $\infty$ $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$ t.d. $\forall k < l$ vrijedi $(k,l) \in [(m,n)]_\sim$ (za neke fiksne $m$ i $n$).
$\rightarrow \forall i \,.\, (s_{k_i}, s_{k_{i+1}}) \in f(m,n) \rightarrow f(m,n)$ nije dobro utemeljena. $\Rightarrow\Leftarrow$

$\Rightarrow$ Neka $P$ terminira. Definiramo $T := R_l^+$.
Kad $T$ ne bi bila dobro utemeljena postojao bi $\sigma = s_0 s_1 \ldots$ t.d. $\forall i \,.\, (s_i, s_{i+1}) \in T$.

# Terminacija

## Dokaz.

$\boxed{\Leftarrow}$ Neka je $T = T_1 \cup \ldots \cup T_n$ t.d. $R_l^+ \subseteq T$.
Pretp. da $P$ ne terminira, tj. postoji $\infty$ izračunavanje $\sigma = s_0 s_1 \ldots$
Za $k < l$ def. $f(k,l) := T_i$, gdje je $T_i$ t.d. $(s_k, s_l) \in T_i$.
Def. rel. ekvivalencije na $\mathbb{N} \times \mathbb{N}$: $(k,l) \sim (k',l') \Leftrightarrow f(k,l) = f(k',l')$.
$\rightarrow \sim$ ima konačan indeks.
Ramseyev tm $\Rightarrow$ postoji $\infty$ $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$ t.d. $\forall k < l$ vrijedi $(k,l) \in [(m,n)]_\sim$ (za neke fiksne $m$ i $n$).
$\rightarrow \forall i \,.\, (s_{k_i}, s_{k_{i+1}}) \in f(m,n) \rightarrow f(m,n)$ nije dobro utemeljena. $\Rightarrow\Leftarrow$

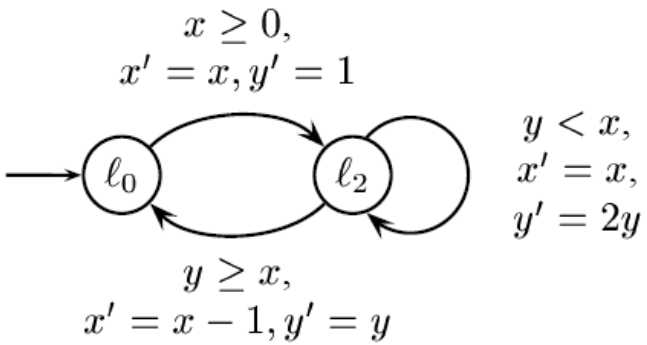
$\boxed{\Rightarrow}$ Neka $P$ terminira. Definiramo $T := R_l^+$.
Kad $T$ ne bi bila dobro utemeljena postojao bi $\sigma = s_0 s_1 \ldots$ t.d. $\forall i \,.\, (s_i, s_{i+1}) \in T$.
$\rightarrow \ldots s_0 \ldots s_1 \ldots$ je $\infty$ izračunavanje u $P$. $\Rightarrow\Leftarrow$

$\square$

# Primjer 1

$$\begin{array}{ll} \textbf{in } n & : \textbf{integer where } n > 0 \\ \textbf{local } x, y & : \textbf{integer where } x = n \end{array}$$

$$\left[\begin{array}{l} \ell_0 : \textbf{while } x \geq 0 \textbf{ do} \\ \quad \left[\begin{array}{l} \ell_1 : y := 1 \\ \ell_2 : \textbf{while } y < x \textbf{ do} \\ \qquad \ell_3 : y := 2y \\ \ell_4 : x := x - 1 \end{array}\right] \end{array}\right]$$



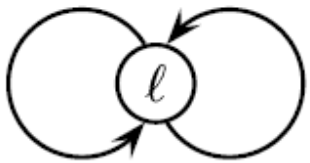

$$\begin{array}{ll} T_1 & x \geq 0 \wedge x' < x \\ T_2 & x - y > 0 \wedge x' - y' < x - y \\ T_{ij} & pc = \ell_i \wedge pc' = \ell_j \quad \text{where } i \neq j \in \{0, \ldots, 4\} \end{array}$$

# Primjer 2

$$\textbf{local } x, y \qquad : \textbf{integer}$$

$$\left[\begin{array}{l} \textbf{while } x > 0 \wedge y > 0 \textbf{ do} \\ \quad \left[\begin{array}{l} \ell^a : (x, y) := (x - 1, x) \\ \quad \textbf{or} \\ \ell^b : (x, y) := (y - 2, x + 1) \end{array}\right] \end{array}\right]$$

$x > 0, y > 0,$
$x' = x - 1,$
$y' = x$




$x > 0, y > 0,$
$x' = y - 2,$
$y' = x + 1$

$$\begin{array}{ll} T_1 & x > 0 \wedge x' < x \\ T_2 & x + y > 0 \wedge x' + y' < x + y \\ T_3 & y > 0 \wedge y' < y \end{array}$$

# Pregled

# Verifikacija liveness svojstava

Kažemo da je program $P$ korektan obzirom na fairness uvjet $\varphi$ i liveness svojstvo $\psi$ ako za svako izračunavanje $\pi$ od $P$ vrijedi:

ako $\pi \vDash \varphi$ onda $\pi \vDash \psi$.

# Verifikacija liveness svojstava

Kažemo da je program $P$ korektan obzirom na fairness uvjet $\varphi$ i liveness svojstvo $\psi$ ako za svako izračunavanje $\pi$ od $P$ vrijedi:

ako $\pi \models \varphi$ onda $\pi \models \psi$.

Kako dokazati da je $P$ korektan obzirom na $(\varphi, \psi)$?

# Verifikacija liveness svojstava

Kažemo da je program $P$ korektan obzirom na fairness uvjet $\varphi$ i liveness svojstvo $\psi$ ako za svako izračunavanje $\pi$ od $P$ vrijedi:

ako $\pi \vDash \varphi$ onda $\pi \vDash \psi$.

Kako dokazati da je $P$ korektan obzirom na $(\varphi, \psi)$?

$\rightarrow$ Dokazujemo da $P$ nije inkorektan, tj. da ne postoji izračunavanje $\pi$ od $P$ t.d. $\pi \models \varphi \wedge \neg\psi$:

# Verifikacija liveness svojstava

Kažemo da je program $P$ korektan obzirom na fairness uvjet $\varphi$ i liveness svojstvo $\psi$ ako za svako izračunavanje $\pi$ od $P$ vrijedi:

ako $\pi \models \varphi$ onda $\pi \models \psi$.

Kako dokazati da je $P$ korektan obzirom na $(\varphi, \psi)$?

$\rightarrow$ Dokazujemo da $P$ nije inkorektan, tj. da ne postoji izračunavanje $\pi$ od $P$ t.d. $\pi \models \varphi \wedge \neg\psi$:

1. Iz $\varphi$ i $\psi$ konstruiramo Büchijev automat $\mathcal{A}_{\varphi,\psi}$ t.d. $L(\mathcal{A}_{\varphi,\psi}) = \{\omega \mid \omega \models \varphi \wedge \neg\psi\}$

# Verifikacija liveness svojstava

Kažemo da je program $P$ korektan obzirom na fairness uvjet $\varphi$ i liveness svojstvo $\psi$ ako za svako izračunavanje $\pi$ od $P$ vrijedi:

ako $\pi \models \varphi$ onda $\pi \models \psi$.

Kako dokazati da je $P$ korektan obzirom na $(\varphi, \psi)$?

$\rightarrow$ Dokazujemo da $P$ nije inkorektan, tj. da ne postoji izračunavanje $\pi$ od $P$ t.d. $\pi \models \varphi \wedge \neg\psi$:

1. Iz $\varphi$ i $\psi$ konstruiramo Büchijev automat $\mathcal{A}_{\varphi,\psi}$ t.d. $L(\mathcal{A}_{\varphi,\psi}) = \{\omega \mid \omega \models \varphi \wedge \neg\psi\}$
2. Pokažemo da se niti jedno izračunavanje od $P$ ne nalazi u $L(\mathcal{A}_{\varphi,\psi})$

# Podsjetnik: Büchijev automat

Büchijev automat $\mathcal{A} = (Q, Q_0, \Delta, F)$ sastoji se od:

- $Q$ — skup stanja
- $Q_0 \subseteq Q$ — skup početnih stanja
- $\Delta \subseteq Q \times W \times Q$ — tranzicijska relacija
- $F$ — skup završnih stanja

$\mathcal{A}$ prihvaća $\omega$-riječ $w = s_0 s_1 \ldots \in W^{\mathbb{N}}$ ako postoji izračunavanje $q_0 q_1 \ldots$ na $w$ t.d. $|\{k \mid q_k \in F\}| = \infty$.

# Verifikacija liveness svojstava

Kako pokazati da niti jedno izračunavanje $\pi$ od $P$ nije u $L(\mathcal{A}_{\varphi,\psi})$?

# Verifikacija liveness svojstava

Kako pokazati da niti jedno izračunavanje $\pi$ od $P$ nije u $L(\mathcal{A}_{\varphi,\psi})$?

1. Konstruiramo $P_{\varphi,\psi} := P \otimes \mathcal{A}_{\varphi,\psi}$:
   - skup stanja: $W \times Q$
   - skup početnih stanja: $I \times Q_0$
   - tranzicijska relacija: $\{((s,q),(s',q')) \mid (s,s') \in R \wedge (q,s,q') \in \Delta\}$
   - skup završnih stanja: $W \times F$

# Verifikacija liveness svojstava

Kako pokazati da niti jedno izračunavanje $\pi$ od $P$ nije u $L(\mathcal{A}_{\varphi,\psi})$?

1. Konstruiramo $P_{\varphi,\psi} := P \otimes \mathcal{A}_{\varphi,\psi}$:
   - skup stanja: $W \times Q$
   - skup početnih stanja: $I \times Q_0$
   - tranzicijska relacija: $\{((s,q),(s',q')) \mid (s,s') \in R \wedge (q,s,q') \in \Delta\}$
   - skup završnih stanja: $W \times F$
2. Pokažemo da $P_{\varphi,\psi}$ pravedno terminira,
   tj. da sva (beskonačna) izračunavanja $\pi$ od $P_{\varphi,\psi}$ nisu pravedna,
   tj. da ne postoji $\pi = (s_0,q_0)(s_1,q_1)\ldots \in (W \times Q)^{\mathbb{N}}$ t.d.
   $|\{k \mid (s_k,q_k) \in W \times F\}| = \infty$.

# Verifikacija liveness svojstava

Kako pokazati da niti jedno izračunavanje $\pi$ od $P$ nije u $L(\mathcal{A}_{\varphi,\psi})$?

1. Konstruiramo $P_{\varphi,\psi} := P \otimes \mathcal{A}_{\varphi,\psi}$:
   - skup stanja: $W \times Q$
   - skup početnih stanja: $I \times Q_0$
   - tranzicijska relacija: $\{((s,q),(s',q')) \mid (s,s') \in R \wedge (q,s,q') \in \Delta\}$
   - skup završnih stanja: $W \times F$
2. Pokažemo da $P_{\varphi,\psi}$ pravedno terminira,
   tj. da sva (beskonačna) izračunavanja $\pi$ od $P_{\varphi,\psi}$ nisu pravedna,
   tj. da ne postoji $\pi = (s_0,q_0)(s_1,q_1)\ldots \in (W \times Q)^{\mathbb{N}}$ t.d.
   $|\{k \mid (s_k,q_k) \in W \times F\}| = \infty$.

**Teorem (Vardi)**

*$P$ je korektan obzirom na $(\varphi,\psi)$ akko $P_{\varphi,\psi}$ pravedno terminira.*

# Kako dokazati pravednu terminaciju?

## Teorem (Vardi)

*$P_{\varphi,\psi}$ pravedno terminira ako i samo ako postoji ordinal $\lambda$ i predikat $\rho\colon 2^{W\times Q\times\lambda} \to \{true, false\}$ t.d. vrijede slijedeća svojstva:*

- *za sve $s \in I$ i $p \in Q_0$ vrijedi $\rho(s, p, \lambda)$;*
- *za sve $s, t \in W$ i $p, q \in Q$, ako vrijedi $\rho(s, p, \mu)$, te je $(s, t) \in R$ i $(p, s, q) \in \Delta$, onda vrijedi i $\rho(t, q, \nu)$, za neki $\nu \leq \mu$;*
- *za sve $s, t \in W$ i $p, q \in Q$, ako vrijedi $\rho(s, p, \mu)$, te je $(s, t) \in R$, $(p, s, q) \in \Delta$ i $p \in F$, onda vrijedi i $\rho(t, q, \nu)$, za neki $\nu < \mu$.*

# Pregled

# Osnovni algoritam za utvrđivanje terminacije

Ulaz: Program $P = (W, I, R)$

$T := \emptyset$

**repeat**

  **if** $R_I^+ \subseteq T$ **then**

    **return** "Program terminira"

  **else**

    $\rho :=$ binarna relacija t.d. $\rho \subseteq R_I^+$ ali $\rho \nsubseteq T$

  **end if**

  **if** $\rho$ nije dobro utemeljena **then**

    **return** "Program ne terminira"

  **else**

    Konstruiraj dobro utemeljenu relaciju $W \supseteq \rho$

    $T := T \cup W$

  **end if**

**until** 0

# Analiza dostiživosti

Kako odrediti vrijedi li $R_I^+ \subseteq T$?

# Analiza dostiživosti

Kako odrediti vrijedi li $R_I^+ \subseteq T$?

1. Karakterizacija $R_I^+$ kao najmanje fiksne točke od $F$.

# Analiza dostiživosti

Kako odrediti vrijedi li $R_I^+ \subseteq T$?

1. Karakterizacija $R_I^+$ kao najmanje fiksne točke od $F$.
2. Transformacija $P \mapsto \hat{P}$, gdje je $\hat{P}$ program koji implementira $F$. (**lfp**$(F)$ je ekvivalentna skupu dostiživih stanja u $\hat{P}$.)

# Analiza dostiživosti

Kako odrediti vrijedi li $R_I^+ \subseteq T$?

1. Karakterizacija $R_I^+$ kao najmanje fiksne točke od $F$.
2. Transformacija $P \mapsto \hat{P}$, gdje je $\hat{P}$ program koji implementira $F$. (**lfp**$(F)$ je ekvivalentna skupu dostiživih stanja u $\hat{P}$.)
3. Transformacija $\hat{P} \mapsto \hat{P}_T$.
   $\rightarrow R_I^+ \subseteq T$ akko u $\hat{P}_T$ nije dostiživo stanje ERROR.

B. Cook, A. Podelski, A. Rybalchenko: Termination proofs for systems code. PLDI'06.

# 1. korak: $\mathbf{lfp}(F, \bot)$

Želimo definirati f-ju $F$ čija je najmanja fiksna točka $R_I^+$.

# 1. korak: $\mathbf{lfp}(F, \bot)$

Želimo definirati f-ju $F$ čija je najmanja fiksna točka $R_I^+$.

Koristimo oznake:

- $\bot := \{(s_1, s_2) \mid s_1 \in I \wedge (s_1, s_2) \in R\}$
- $id_2(X) := \{(s_2, s_2) \mid \exists s_1 \,.\, (s_1, s_2) \in X\}$
- $X \circ Y := \{(s_1, s_3) \mid \exists s_2 \,.\, (s_1, s_2) \in X \wedge (s_2, s_3) \in Y\}$.

# 1. korak: $\mathbf{lfp}(F, \bot)$

Želimo definirati f-ju $F$ čija je najmanja fiksna točka $R_I^+$.

Koristimo oznake:

- $\bot := \{(s_1, s_2) \mid s_1 \in I \wedge (s_1, s_2) \in R\}$
- $id_2(X) := \{(s_2, s_2) \mid \exists s_1 \,.\, (s_1, s_2) \in X\}$
- $X \circ Y := \{(s_1, s_3) \mid \exists s_2 \,.\, (s_1, s_2) \in X \wedge (s_2, s_3) \in Y\}$.

Definirajmo $F : S \times S \to S \times S$ s

$$F(X) := (X \cup id_2(X)) \circ R$$

# 1. korak: $\mathbf{lfp}(F, \bot)$

Želimo definirati f-ju $F$ čija je najmanja fiksna točka $R_I^+$.

Koristimo oznake:

- $\bot := \{(s_1, s_2) \mid s_1 \in I \wedge (s_1, s_2) \in R\}$
- $id_2(X) := \{(s_2, s_2) \mid \exists s_1 \,.\, (s_1, s_2) \in X\}$
- $X \circ Y := \{(s_1, s_3) \mid \exists s_2 \,.\, (s_1, s_2) \in X \wedge (s_2, s_3) \in Y\}$.

Definirajmo $F : S \times S \to S \times S$ s

$$F(X) := (X \cup id_2(X)) \circ R$$

**Lema**

$$R_I^+ = \mathbf{lfp}(F, \bot)$$

# 2. korak: $P \mapsto \hat{P}$

```
L: stmt    ⟹    L:  if (nondet()) {
                       'v1 = v1;
                       ...
                       'vn = vn;
                       'pc = L;
                   } else {
                       skip;
                   }
                   stmt
```

# 2. korak: $P \mapsto \hat{P}$

```
L: stmt      ⟹      L:  if (nondet()) {
                             'v1 = v1;
                             ...
                             'vn = vn;
                             'pc = L;
                         } else {
                             skip;
                         }
                         stmt
```

- varijable u $\hat{P}$: $\{v_1, \ldots, v_n\} \cup \{'v_1, \ldots, 'v_n\}$
- $\hat{P} \ni \hat{s} \simeq (s_1, s_2) \in P$ akko $[\!['v_i]\!]_{\hat{s}} = [\![v_i]\!]_{s_1}$ i $[\![v_i]\!]_{\hat{s}} = [\![v_i]\!]_{s_2}$
- $\hat{I} = \{\hat{s} \mid \exists s_0 \in I \,.\, \hat{s} \simeq (s_0, s_0)\}$

# 2. korak: $P \mapsto \hat{P}$

```
L: stmt    ⟹    L:  if (nondet()) {
                     'v1 = v1;
                     ...
                     'vn = vn;
                     'pc = L;
                   } else {
                     skip;
                   }
                   stmt
```

- varijable u $\hat{P}$: $\{v_1, \ldots, v_n\} \cup \{'v_1, \ldots, 'v_n\}$
- $\hat{P} \ni \hat{s} \simeq (s_1, s_2) \in P$ akko $[\!['v_i]\!]_{\hat{s}} = [\![v_i]\!]_{s_1}$ i $[\![v_i]\!]_{\hat{s}} = [\![v_i]\!]_{s_2}$
- $\hat{I} = \{\hat{s} \mid \exists s_0 \in I \,.\, \hat{s} \simeq (s_0, s_0)\}$

**Lema**

$$\mathbf{lfp}(F, \bot) = \mathbf{post}^{+}_{\hat{P}}(\hat{I})$$

# 3. korak: $\hat{P} \mapsto \hat{P}_T$

```
L: if (nondet()) {...} stmt
```

$\Downarrow$

```
L:   if (!(T_L)) { ERROR: skip; }
     if (nondet()) {...} stmt
```

# 3. korak: $\hat{P} \mapsto \hat{P}_T$

```
L: if (nondet()) {...} stmt
```

$\Downarrow$

```
L:   if (!(T_L)) { ERROR: skip; }
     if (nondet()) {...} stmt
```

**Teorem**

*$R_I^+ \subseteq T$ akko u $\hat{P}_T$ nije dostiživo stanje ERROR.*

# Nalaženje rangirajuće funkcije / relacije

- Općenito: teško!
- 3rd-party alati: PolyRank, RankFinder, ...
- Postoji potpuna metoda za nalaženje linearnih rangirajućih funkcija – za većinu slučajeva dovoljno dobro

  A. Podelski, A. Rybalchenko: A complete method for the synthesis of linear ranking functions. VMCAI'04.

# Tretiranje fairnessa

"Zapis fairness uvjeta putem logičke formule ili Büchijevog automata nije user-friendly."

# Tretiranje fairnessa

"Zapis fairness uvjeta putem logičke formule ili Büchijevog automata nije user-friendly."

$\longrightarrow$ Koristi se Streettov automat
$\mathcal{A} = (Q, Q_0, \Delta, \mathcal{C} = \{(P_1, Q_1), \ldots, (P_n, Q_n)\})$

- $\omega$-riječ $w \in W^{\mathbb{N}}$ je prihvaćena ako postoji izračunavanje $\varrho$ na $w$ t.d. za sve $i = 1, \ldots, n$ vrijedi $\mathrm{Inf}(\varrho) \cap P_i = \emptyset$ ili $\mathrm{Inf}(\varrho) \cap Q_i \neq \emptyset$

# Tretiranje fairnessa

"Zapis fairness uvjeta putem logičke formule ili Büchijevog automata nije user-friendly."

$\rightarrow$ Koristi se Streettov automat
$\mathcal{A} = (Q, Q_0, \Delta, \mathcal{C} = \{(P_1, Q_1), \ldots, (P_n, Q_n)\})$

- $\omega$-riječ $w \in W^{\mathbb{N}}$ je prihvaćena ako postoji izračunavanje $\varrho$ na $w$ t.d. za sve $i = 1, \ldots, n$ vrijedi $\mathrm{Inf}(\varrho) \cap P_i = \emptyset$ ili $\mathrm{Inf}(\varrho) \cap Q_i \neq \emptyset$

Terminator kao specifikacijski jezik koristi SLIC nadograđen konstruktima za opis Streetovog automata.

# Tretiranje fairnessa

$$\text{None}_{\mathcal{C}}(S) := \{(P, Q) \in \mathcal{C} \mid S \cap P = \emptyset\}$$
$$\text{Some}_{\mathcal{C}}(S) := \{(P, Q) \in \mathcal{C} \mid S \cap Q \neq \emptyset\}$$

# Tretiranje fairnessa

$\text{None}_{\mathcal{C}}(S) := \{(P, Q) \in \mathcal{C} \mid S \cap P = \emptyset\}$
$\text{Some}_{\mathcal{C}}(S) := \{(P, Q) \in \mathcal{C} \mid S \cap Q \neq \emptyset\}$

$R_{\mathcal{C}} = \{(s_0, s_n) \mid \exists \sigma = s0 \ldots s_n \,.\, \text{None}_{\mathcal{C}}(\sigma) \cup \text{Some}_{\mathcal{C}}(\sigma) = \mathcal{C}\}$

# Tretiranje fairnessa

$\text{None}_{\mathcal{C}}(S) := \{(P, Q) \in \mathcal{C} \mid S \cap P = \emptyset\}$
$\text{Some}_{\mathcal{C}}(S) := \{(P, Q) \in \mathcal{C} \mid S \cap Q \neq \emptyset\}$

$R_{\mathcal{C}} = \{(s_0, s_n) \mid \exists \sigma = s0 \ldots s_n \,.\, \text{None}_{\mathcal{C}}(\sigma) \cup \text{Some}_{\mathcal{C}}(\sigma) = \mathcal{C}\}$

**Teorem**

*Program $P$ terminira pod fairness uvjetima $\mathcal{C}$ ako postoji disjunktivno dobro utemeljena relacija $T$ t.d. $R_{\mathcal{C}} \subseteq T$.*

B. Cook, A. Gotsman, A. Podelski, A. Rybalchenko, M. Vardi: Proving That Programs Eventually Do Something Good. POPL'07.

# Pregled

# Prijedlog projekta

2. faza:

- Izrada CLI model checkera za liveness
- Arhitektura poput one u Terminatoru
- Koristiti gotove vanjske alate za sintezu rangirajućih funkcija
- Iskoristiti safety model checker iz 1. faze

# Prijedlog projekta

$1\frac{1}{2}$. faza:

- CLI model checker osjetljiv na .NET memorijski model (.NET memorijski model je jako relaksiran)
- Problem 1: Neformalan i neprecizan opis memorijskog modela (ECMA-335 CLI specifikacija)
- Problem 2: Stvarne implementacije memorijskog modela u praksi daju znatno čvršće garancije od onih koje su propisane specifikacijom — što uzeti kao referentan model?
- Problem 3: Nedostatak dobrog formalnog pristupa problemu
- Pokušati iskoristiti postojeća znanja iz Java svijeta

# Literatura

B. Cook, A. Gotsman, A. Podelski, A. Rybalchenko, M. Vardi: Proving That Programs Eventually Do Something Good. POPL'07.

B. Cook, A. Podelski, A. Rybalchenko: Termination proofs for systems code. PLDI'06.

M. Y. Vardi: Verification of Concurrent Programs: The Automata-Theoretic Framework. APAL 51(1-2), 79–98, 1991.

M. Y. Vardi, P. Wolper: An automata-theoretic approach to automatic program verification. LICS'86.

A. Podelski, A. Rybalchenko: A complete method for the synthesis of linear ranking functions. VMCAI'04.

A. Podelski, A. Rybalchenko: Transition invariants. LICS'04.

A. Podelski, A. Rybalchenko: Transition predicate abstraction and fair termination. POPL'05.