A Generic Framework for Symbolic Execution

Jan Tušil

7. prosince 2017

MojeIntro

```
int x,y;
x = get();
y = -x;
y = -y;
assert(x == y);
```

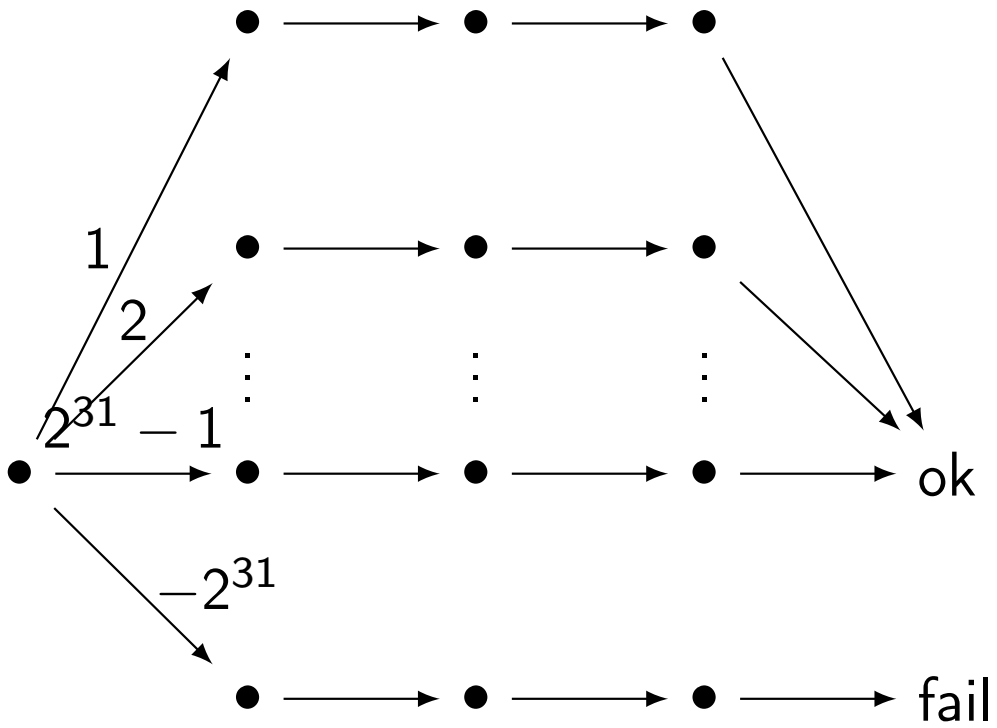
Může assert selhat?

Operační sémantika

$$OpSem : Program \rightarrow TransitionSystem$$

Operační sémantika

$$OpSem : Program \rightarrow TransitionSystem$$



Konfigurace

$$\langle x = get(); \curvearrowright y = -x; \curvearrowright y = -y; \curvearrowright assert(x == y); \rangle_{\,\mathsf{k}} \langle x = 0, y = 0 \rangle_{\,\mathsf{env}}$$

Konfigurace

$$\langle x = get(); \curvearrowright y = -x; \curvearrowright y = -y; \curvearrowright assert(x == y); \rangle_{\,\mathsf{k}} \langle x = 0, y = 0 \rangle_{\,\mathsf{env}}$$

Ukázka

Symbolická Konfigurace

$$\langle y = -x; \curvearrowright y = -y; \curvearrowright \mathit{assert}(x == y); \rangle_{\,\mathsf{k}} \langle x = X, y = 0 \rangle_{\,\mathsf{env}}$$

Symbolická Konfigurace

$$\langle y = -x; \curvearrowright y = -y; \curvearrowright assert(x == y); \rangle_{\,k} \langle x = X, y = 0 \rangle_{\,env}$$

Ukázka

Symbolická exekuce

Princip (Pokrytí)

Každému (potenciálně nekonečnému) konkrétnímu běhu odpovídá nějaký symbolický běh.

Symbolická exekuce

Princip (Pokrytí)

Každému (potenciálně nekonečnému) konkrétnímu běhu odpovídá nějaký symbolický běh.

Princip (Přesnost)

Každému konečnému symbolickému běhu odpovídá nějaký konkrétní běh.

Symbolická exekuce

Princip (Pokrytí)

Každému (potenciálně nekonečnému) konkrétnímu běhu odpovídá nějaký symbolický běh.

Princip (Přesnost)

Každému konečnému symbolickému běhu odpovídá nějaký konkrétní běh.

Nekonečné běhy - koindukce

FOL

$$\phi ::= \top \mid p(t_1, \ldots, t_n) \mid \neg\phi \mid \phi \wedge \phi \mid (\exists X)\, \phi \tag{1}$$

Matching Logic - logika konfigurací

Signature ML: 123

$$\varphi ::= \pi \mid \top \mid p(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi \wedge \varphi \mid (\exists V)\, \varphi \tag{2}$$