A Generic Framework for Symbolic Execution

Jan Tušil

8. prosince 2017

1. Intro

2. Jazyk, logika, sémantika
   - Logika konfigurací

MojeIntro

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

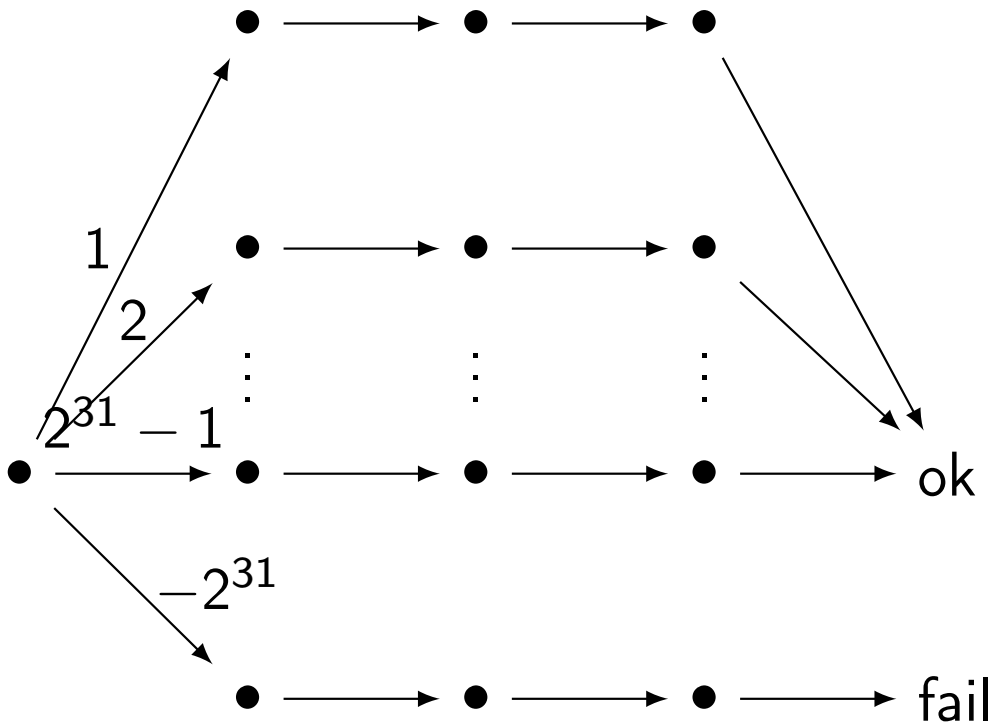
Může assert selhat?

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

Signatura

Poznámka (Sortování)

Říkáme-li o množině X, že je Y-sortovaná, máme tím na mysli, že existuje funkce SortOf $: X \to Y$.

Signatura

Poznámka (Sortování)

Říkáme-li o množině X, že je Y-sortovaná, máme tím na mysli, že existuje funkce SortOf $: X \to Y$.

Definice

Vícedruhová algebraická signatura je tvořena množinou sortů S spolu s $S^* \times S$-sortovanou množinou Σ funkčních symbolů. Symbol T_Σ označuje Σ-algebru uzavřených termů, $T_{\Sigma,s}$ množinu termů sortu s, $T_\Sigma(Var)$ volnou Σ-algebru termů s proměnnými.

Signatura

Poznámka (Sortování)

Říkáme-li o množině X, že je Y-sortovaná, máme tím na mysli, že existuje funkce SortOf : $X \to Y$.

Definice

Vícedruhová algebraická signatura je tvořena množinou sortů S spolu s $S^* \times S$-sortovanou množinou Σ funkčních symbolů. Symbol T_Σ označuje Σ-algebru uzavřených termů, $T_{\Sigma,s}$ množinu termů sortu s, $T_\Sigma(Var)$ volnou Σ-algebru termů s proměnnými.

```
Plant ::= favouriteFood(Animal)
Animal ::= mother(Animal) | father(Animal)
```

Signatura

Poznámka (Sortování)

Říkáme-li o množině X, že je Y-sortovaná, máme tím na mysli, že existuje funkce SortOf : $X \to Y$.

Definice

Vícedruhová algebraická signatura je tvořena množinou sortů S spolu s $S^* \times S$-sortovanou množinou Σ funkčních symbolů. Symbol T_Σ označuje Σ-algebru uzavřených termů, $T_{\Sigma,s}$ množinu termů sortu s, $T_\Sigma(Var)$ volnou Σ-algebru termů s proměnnými.

```
Plant ::= favouriteFood(Animal)
Animal ::= mother(Animal) | father(Animal)
```

Příklad.

Vícedruhová FOL

Definice

Signatura provořádové logiky (Σ, Π) *je tvořena algebraickou signaturou* Σ *a* S^**-sortovanou množinou predikátových symbolů* Π.

Vícedruhová FOL

Definice

Signatura provořádové logiky (Σ, Π) *je tvořena algebraickou signaturou* Σ *a* S^**-sortovanou množinou predikátových symbolů* Π*.*

Definice (Formule)

Množina formulí nad signaturou (Σ, Π) *je definována:*

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

kde p *označuje predikátové symboly,* X *podmnožiny proměnných a* t_i Σ*-termy s volnými proměnnými.*

Vícedruhová FOL

Definice

Signatura provořádové logiky (Σ, Π) *je tvořena algebraickou signaturou* Σ *a* S^**-sortovanou množinou predikátových symbolů* Π*.*

Definice (Formule)

Množina formulí nad signaturou (Σ, Π) *je definována:*

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

kde p *označuje predikátové symboly,* X *podmnožiny proměnných a* t_i *Σ-termy s volnými proměnnými.*

Příklad (StaršíNež).

Vícedruhová FOL

Definice

Signatura provořádové logiky (Σ, Π) *je tvořena algebraickou signaturou* Σ *a* S^**-sortovanou množinou predikátových symbolů* Π*.*

Definice (Formule)

Množina formulí nad signaturou (Σ, Π) *je definována:*

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

kde p *označuje predikátové symboly,* X *podmnožiny proměnných a* t_i *Σ-termy s volnými proměnnými.*

Příklad (StaršíNež). Předpokládáme predikát rovnosti.

Modely

Analogie s realizacemi jazyků v MA007.

Definice

Model signatury (Σ, Π)*je* Σ*-algebra* $\mathcal{T}$ *spolu s realizací* $\mathcal{T}_p \subseteq \mathcal{T}_{s_1} \times \cdots \times \mathcal{T}_{s_n}$ *pro každý predikátový symbol* $p \in \Pi_{s_1 \ldots s_n}$.

Modely

Analogie s realizacemi jazyků v MA007.

Definice

Model signatury (Σ, Π)*je* Σ*-algebra* $\mathcal{T}$ *spolu s realizací* $\mathcal{T}_p \subseteq \mathcal{T}_{s_1} \times \cdots \times \mathcal{T}_{s_n}$ *pro každý predikátový symbol* $p \in \Pi_{s_1 \ldots s_n}$.

Zvířecí příklad.

Modely

Analogie s realizacemi jazyků v MA007.

Definice

Model signatury (Σ, Π)*je* Σ*-algebra* $\mathcal{T}$ *spolu s realizací* $\mathcal{T}_p \subseteq \mathcal{T}_{s_1} \times \cdots \times \mathcal{T}_{s_n}$ *pro každý predikátový symbol* $p \in \Pi_{s_1 \ldots s_n}$.

Zvířecí příklad.

Definice (Relace splnitelnosti)

Pro (Σ, Π)*-formuli* ϕ, (Σ, Π)*-model* $\mathcal{T}$ *a valuaci* $\rho : Var \to \mathcal{T}$ *definujeme relaci splnitelnosti* $\rho \models \phi$ *jako obvykle. (Valuaci* ρ *přirozeně rozšiřujeme na morfismus* Σ*-algeber* $\rho : T_\Sigma(Var) \to \mathcal{T}$.)

Modely

Analogie s realizacemi jazyků v MA007.

Definice

Model signatury (Σ, Π)*je* Σ*-algebra* $\mathcal{T}$ *spolu s realizací* $\mathcal{T}_p \subseteq \mathcal{T}_{s_1} \times \cdots \times \mathcal{T}_{s_n}$ *pro každý predikátový symbol* $p \in \Pi_{s_1 \ldots s_n}$.

Zvířecí příklad.

Definice (Relace splnitelnosti)

Pro (Σ, Π)*-formuli* ϕ, (Σ, Π)*-model* $\mathcal{T}$ *a valuaci* $\rho : Var \to \mathcal{T}$ *definujeme relaci splnitelnosti* $\rho \models \phi$ *jako obvykle. (Valuaci* ρ *přirozeně rozšiřujeme na morfismus* Σ*-algeber* $\rho : T_\Sigma(Var) \to \mathcal{T}$*.)*

Zkusme to na tabuli.

Modely

Analogie s realizacemi jazyků v MA007.

Definice

Model signatury (Σ, Π)*je* Σ*-algebra* $\mathcal{T}$ *spolu s realizací* $\mathcal{T}_p \subseteq \mathcal{T}_{s_1} \times \cdots \times \mathcal{T}_{s_n}$ *pro každý predikátový symbol* $p \in \Pi_{s_1 \ldots s_n}$.

Zvířecí příklad.

Definice (Relace splnitelnosti)

Pro (Σ, Π)*-formuli* ϕ, (Σ, Π)*-model* $\mathcal{T}$ *a valuaci* $\rho : Var \to \mathcal{T}$ *definujeme relaci splnitelnosti* $\rho \models \phi$ *jako obvykle. (Valuaci* ρ *přirozeně rozšiřujeme na morfismus* Σ*-algeber* $\rho : T_\Sigma(Var) \to \mathcal{T}$.)

Zkusme to na tabuli. Příklad.

Matching Logic - logika konfigurací

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