Why

Daria Walukiewicz-Chrząszcz
Weryfikacja Wspomagana Komputerowo

24 kwietnia 2014

Co to jest Why ?

Why to narzędzie do weryfikacji programów

Co to jest Why ?

Why to narzędzie do weryfikacji programów

`http://why3.lri.fr/`

Co to jest Why ?

Why to narzędzie do weryfikacji programów

`http://why3.lri.fr/`

LRI, Uniwersytet Paris-Sud:
François Bobot, Jean-Christophe Filliâtre, Claude Marché, Guillaume Melquiond, Andrei Paskevich,

- na wejściu dany program imperatywny i jego specyfikacja

Co to jest Why ?

Why to narzędzie do weryfikacji programów

`http://why3.lri.fr/`

LRI, Uniwersytet Paris-Sud:
François Bobot, Jean-Christophe Filliâtre, Claude Marché, Guillaume Melquiond, Andrei Paskevich,

- na wejściu dany program imperatywny i jego specyfikacja
- Why3 generuje VC (verification conditions), term w Coqu zawierający fukcyjne tłumaczenie programu oraz dowód poprawności programu oparty na VC

Co to jest Why ?

Why to narzędzie do weryfikacji programów

`http://why3.lri.fr/`

LRI, Uniwersytet Paris-Sud:
François Bobot, Jean-Christophe Filliâtre, Claude Marché, Guillaume Melquiond, Andrei Paskevich,

- na wejściu dany program imperatywny i jego specyfikacja
- Why3 generuje VC (verification conditions), term w Coqu zawierający fukcyjne tłumaczenie programu oraz dowód poprawności programu oparty na VC
- VC magą być dowodzone w Coqu lub w innych interaktywnych lub automatycznych proverach.

Provery działające z Why3

automatyczne Alt-Ergo, CVC3 E-prover, Gappa, Simplify, SPASS, Vampire, veriT, Yices, Z3,

interaktywne Coq

Logika i programy w Why3

Platforma Why3 składa się zasadniczo z dwóch części:

Logika i programy w Why3

Platforma Why3 składa się zasadniczo z dwóch części:

- **Why** - język logiki pierwszego rzędu z polimorficznymi typami

Logika i programy w Why3

Platforma Why3 składa się zasadniczo z dwóch części:

- **Why** - język logiki pierwszego rzędu z polimorficznymi typami
- **Why3ml** - język programowania – oprócz logiki można używać referencji, pętli, wyjatków, anotacji, można definiować funkcje nieterminujące, terminujące ...

Weryfikacja programów w Why3

Używanie Why3 do weryfikacji programów:

Weryfikacja programów w Why3

Używanie Why3 do weryfikacji programów:

- zapisanie programu w Why3ml,

Weryfikacja programów w Why3

Używanie Why3 do weryfikacji programów:

- zapisanie programu w Why3ml,
- zapisanie specyfikacji (warunki wstępne, końcowe, niezmienniki) używając języka logicznego,

Weryfikacja programów w Why3

Używanie Why3 do weryfikacji programów:

- zapisanie programu w Why3ml,
- zapisanie specyfikacji (warunki wstępne, końcowe, niezmienniki) używając języka logicznego,
- generowane są VC (verification conditions) oparte o najsłabszy warunek wstępny i niezmienniki; terminacja dowodzona w oparciu o varianty

Weryfikacja programów w Why3

Używanie Why3 do weryfikacji programów:

- zapisanie programu w Why3ml,
- zapisanie specyfikacji (warunki wstępne, końcowe, niezmienniki) używając języka logicznego,
- generowane są VC (verification conditions) oparte o najsłabszy warunek wstępny i niezmienniki; terminacja dowodzona w oparciu o varianty
- VC tłumaczone są do używanych proverów

Weryfikacja programów w Why3

Używanie Why3 do weryfikacji programów:

- zapisanie programu w Why3ml,
- zapisanie specyfikacji (warunki wstępne, końcowe, niezmienniki) używając języka logicznego,
- generowane są VC (verification conditions) oparte o najsłabszy warunek wstępny i niezmienniki; terminacja dowodzona w oparciu o varianty
- VC tłumaczone są do używanych proverów
- interfejs graficzny: **why3ide**

Specyfikacje w Why3

Warunki wstępne i końcowe

```
let f x y z
    requires {P} ensures {Q}
    = ....
```

Niezmienniki

```
while ... do invariant {I} ... done
for i = ... do invariant {I(i)} ... done
```

Terminacja pętli

```
variant {t}
```

gdzie t jest wyrażeniem typu int i maleje za każdym obrotem w relacji

$$y \prec x \equiv y < x \wedge 0 \leq x$$