

Prednáška 7: Jazyk Z

Metódy a prostriedky špecifikácie 2013/14

Valentino Vranić

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave

12. november 2013

Obsah prednášky

- 1 Úvod
- 2 Základy jazyka Z
- 3 Štruktúra špecifikácie v jazyku Z
- 4 Špecifikácia operácií
- 5 Spájanie operácií
- 6 Uplatnenie jazyka Z

Úvod

Úvod

- Softvérové systémy sa špecifikujú väčšinou neformálnymi prostriedkami
- Takýto prístup je zdrojom chýb
- Zároveň nie je možné formálnym spôsobom overiť vlastnosti špecifikácie
- Podnet pre uplatnenie matematického formalizmu v špecifikovaní softvéru
- Pozrieme sa na jazyk Z ako príklad formálneho prístupu k špecifikácii

Základy jazyka Z

Jazyk Z

- Jean-Raymond Abrial, Steve Schuman a Bertrand Meyer, 1977
- Zermelova–Fraenkelova teória množín
- Matematické základy
 - Predikátová logika
 - Teória množín
- Štruktúrovanosť
 - Schémy
 - Typy
- Použitie prirodzeného jazyka
 - Pomenovanie identifikátorov
 - Komentár
- Spresňovanie (refinement)

Štruktúra špecifikácie v jazyku Z

Typická organizácia špecifikácie v jazyku Z

- 1 Typy
- 2 Stav a inicializácia systému
- 3 Základné operácie
- 4 Robustné operácie

Príklad: Systém správy balíkov

- Softvérové balíky (knihnice, pomocné programy a pod.) sú potenciálne spoločné pre viac aplikácií – stačí ich inštalovať iba raz¹
- Každá aplikácia pri inštalácii pridáva nové balíky
- Pri odinštalovaní aplikácie sa nesmú odinštalovať balíky, ktoré sú používané zo strany iných aplikácií

¹M. Kiselkov. Systém správy balíkov. Správa k projektu, Metódy a prostriedky špecifikácie 2001/02, FEI STU.

Typy (1)

- V jazyku Z typ je množina
- Základný je len typ celých čísel \mathbb{Z}
- Ostatné typy sa musia definovať
- Daný typ – typ s neznámou štruktúrou (nezaujímavou z hľadiska špecifikácie):

[Balik]

- Typ definovaný vymenovaním:

Sprava ::= OK

| *‘Vazba uz existuje‘* | *‘Vazba neexistuje‘*

| *‘Balik nie je k dispozicii‘* | *‘Balik uz je k dispozicii‘*

| *‘Balik nie je nainstalovany‘* | *‘Balik uz je nainstalovany‘*

| *‘Vazba sa este pouziva‘*

| *‘Nie su nainstalovane potrebne baliky‘*

| *‘Balik sa nemouze odinstalovat‘*

Typy (2)

- Typ pomocou potenčnej množiny

\mathbb{P} *Balik*

- *Balik* je množina všetkých balíkov – jeho *potenčná množina* je množina všetkých podmnožín tejto množiny
- Príklad potenčnej množiny (power set):

$$\mathbb{P}\{1, 2, 3\} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

Typy (3)

- Štruktúrované typy sa vytvárajú pomocou kartézskeho súčinu:

$$\textit{Balik} \times \textit{Balik}$$

- Typy možno pomenovať definíciou skratky:

$$\textit{Vazba} == \textit{Balik} \times \textit{Balik}$$

- Iný spôsob vytvorenia štruktúrovaného typu je schémou

Špecifikácia operácií

Schéma

- Schéma pozostáva z deklarácie premenných a predikátu, ktorý ohraničuje ich hodnoty



- Rovnocenné je horizontálne vyjadrenie:

$$Schema \hat{=} [Deklaracia \mid Predikat]$$

Stav a inicializácia systému (1)

System

DispBalik : \mathbb{P} *Balik*

InstBalik : \mathbb{P} *Balik*

ZzVazba : \mathbb{P} *Vazba*

$ZzVazba \langle \text{InstBalik} \rangle \subseteq \text{InstBalik}$

- Stav systému je daný hodnotami premenných, z ktorých pozostáva
- Dvojbodka znamená, že premenná nadobúda hodnoty z danej množiny

Stav a inicializácia systému (2)

Inicializacia

System'

ZzVazba' = \emptyset

DispBalik' = \emptyset

InstBalik' = \emptyset

- Apostrof vyjadruje stav po operácii
- Takto môžeme vyjadriť *dôsledok* (postcondition)

Operácie a konvencie Ξ a Δ

- Operácie pracujú nad stavom systému – preto v nich zahrňame schému stavu systému
- Či operácia mení stav systému alebo nie, určuje spôsob jeho zahrnutia: konvencia Ξ alebo Δ



Operácie bez zmeny stavu (1)

- Zistenie, ktoré balíky sú potrebné, nemení stav systému
- Vstupné premenné sú označené otáznikom, výstupné výkričníkom

PotrebneBaliky

\exists System

balik? : *Balik*

potrebne! : \mathbb{P} *Balik*

$potrebne! = ZzVazba(\{balik?\}) \setminus InstBalik$

Operácie bez zmeny stavu (2)

- Ani zistenie, ktoré balíky sú potrebné, ale nie sú k dispozícii (tiež nemení stav systému):

PotrebneBalikyNieKDispozicii _____

\exists System

balik? : *Balik*

potrebne! : \mathbb{P} *Balik*

potrebne! = $(ZzVazba(\{balik?\}) \setminus InstBalik) \setminus DispBalik$

Operácie so zmenou stavu (1)

- Pridanie balíka do množiny balíkov, ktoré sú k dispozícii, stav systému mení
- Pri zmene stavu sa uvádzajú aj *predpoklady* (preconditions) – obsahujú len premenné bez apostrofu

RegistrujBalik

Δ *System*

balik? : *Balik*

balik? \notin *DispBalik*

$DispBalik' = DispBalik \cup \{balik?\}$

Operácie so zmenou stavu (2)

- Odober balík z množiny balíkov, ktoré sú k dispozícii

VymazBalik

Δ *System*

balik? : *Balik*

balik? \in *DispBalik*

$DispBalik' = DispBalik \setminus \{balik?\}$

Operácie so zmenou stavu (3)

- Pridaj väzbu do množiny väzieb

RegistrujVazbu

Δ *System*

vazba? : *Vazba*

$vazba? \notin ZzVazba$

$ZzVazba' = ZzVazba \cup \{vazba?\}$

Operácie so zmenou stavu (4)

- Odober väzbu z množiny väzieb

VymazVazbu

Δ System

vazba? : *Vazba*

vazba? \in *ZzVazba*

first vazba? \notin *DispBalik* \wedge *first vazba?* \notin *InstBalik*

ZzVazba' = *ZzVazba* \setminus {*vazba?*}

Operácie so zmenou stavu (5)

- Nainštaluj balík – inštalácia sa nesmie vykonať, pokiaľ nie sú nainštalované všetky balíky, od ktorých je balík závislý

Instalacia

Δ System

balik? : *Balik*

balik? \notin *InstBalik*

balik? \in *DispBalik*

$ZzVazba(\text{InstBalik} \cup \{\text{balik?}\}) \setminus \text{InstBalik} = \emptyset$

$\text{InstBalik}' = \text{InstBalik} \cup \{\text{balik?}\}$

Operácie so zmenou stavu (6)

- Odinštaluj balík – balík sa nesmie odinštalovať, pokiaľ je nainštalovaný iný od neho závislý balík

Odinstalovanie

Δ System

balik? : *Balik*

balik? \in *InstBalik*

$\text{dom}(ZzVazba \triangleright \{balik?\}) \cap InstBalik = \emptyset$

$InstBalik' = InstBalik \setminus \{balik?\}$

Spájanie operácií

Spájanie operácií

- Uvedené operácie nedávajú používateľovi najavo nič o výsledku
- V jazyku Z výstup modelujeme pomocou výstupných premenných
- Tieto operácie sú určené na spájanie so základnými operáciami

Operácie zabezpečujúce hlásenia (1)

- Niektoré operácie zabezpečujúce hlásenia sú úplne generické

Uspech

sp! : *Sprava*

sp! = 'OK'

Operácie zabezpečujúce hlásenia (2)

- Kontrola či väzba už existuje alebo neexistuje

VazbaExistuje

\exists System

vazba? : *Vazba*

sp! : *Sprava*

vazba? \in *ZzVazba*

sp! = ‘*Vazba uz existuje*‘

Operácie zabezpečujúce hlásenia (3)

VazbaNeexistuje

\exists System

vazba? : *Vazba*

sp! : *Sprava*

vazba? \notin *ZzVazba*

sp! = ‘*Vazba neexistuje*‘

Operácie zabezpečujúce hlásenia (4)

- Kontrola, či je balík registrovaný (je v množine balíkov ktoré sú k dispozícií) alebo nie je registrovaný

BalikJeReg

\exists System

balik? : *Balik*

sp! : *Sprava*

balik? \in *DispBalik*

sp! = ‘*Balik uz je k dispozicii*‘

Operácie zabezpečujúce hlásenia (5)

BalikNieJeReg

\exists System

balik? : *Balik*

sp! : *Sprava*

balik? \notin *DispBalik*

sp! = ‘*Balik nie je k dispozicii*‘

Operácie zabezpečujúce hlásenia (6)

- Kontrola, či je balík nainštalovaný, resp. nie je nainštalovaný

BalikJelInst

\exists System

balik? : *Balik*

sp! : *Sprava*

balik? \in *InstBalik*

sp! = ‘*Balik uz je nainstalovany*‘

Operácie zabezpečujúce hlásenia (7)

BalikNieJeInst

\exists System

balik? : *Balik*

sp! : *Sprava*

balik? \notin *InstBalik*

sp! = ‘*Balik nie je nainstalovany*‘

Operácie zabezpečujúce hlásenia (8)

- Kontrola, či sa väzba ešte používa

VazbaSaPouziva

\exists System

vazba? : *Vazba*

sp! : *Sprava*

first vazba? \in *DispBalik* \vee *first vazba?* \in *InstBalik*

sp! = 'Vazba sa este pouziva'

Operácie zabezpečujúce hlásenia (9)

- Kontrola, či sú nainštalované všetky potrebné balíky

ChybajuBaliky

\exists System

balik? : *Balik*

sp! : *Sprava*

$ZzVazba(\text{InstBalik} \cup \{balik?\}) \setminus \text{InstBalik} \neq \emptyset$

sp! = 'Nie su nainstalovane potrebne baliky'

Operácie zabezpečujúce hlásenia (10)

- Kontrola, či sa balík môže odinštalovať

BalikSaPouziva

\exists System

balik? : *Balik*

sp! : *Sprava*

$\text{dom}(ZzVazba \triangleright \{balik?\}) \cap InstBalik \neq \emptyset$

sp! = '*Balik sa nemouze odinstalovat*'

Spájanie špecifikácií operácií

- Špecifikácie operácií môžeme spájať logickými spojkami
- Výsledná schéma obsahuje všetky premenné v deklaračnej časti
- Predikáty sa spoja príslušnou logickou spojkou
- Toto je spôsob *oddelenia záležitostí* (separation of concerns)
 - Hlavná logika operácie je prezentovaná v samostatnej schéme
 - Každá ďalšia záležitosť, ako napr. výpisy a ošetrenia chybových stavov, je tiež prezentovaná v samostatnej schéme
 - Spojenie týchto záležitostí je zabezpečené spojením schém príslušnými logickými spojkami (samostatnou konštrukciou)
- Súvisí to s aspektovo-orientovaným prístupom

Robustné operácie (1)

$$\text{RegistrujBalikRO} \hat{=} (\text{RegistrujBalik} \wedge \text{Uspech}) \vee \text{BalikJeReg}$$
$$\text{VymazBalikRO} \hat{=} (\text{VymazBalik} \wedge \text{Uspech}) \vee \text{BalikNieJeReg}$$
$$\text{RegistrujVazbuRO} \hat{=} (\text{RegistrujVazbu} \wedge \text{Uspech}) \vee \text{VazbaExistuje}$$

Robustné operácie (2)

$$\text{VymazVazbuRO} \hat{=} (\text{VymazVazbu} \wedge \text{Uspech}) \vee \text{VazbaNeexistuje}$$
$$\text{PotrebneBalikyRO} \hat{=} \text{PotrebneBaliky} \wedge \text{Uspech}$$
$$\begin{aligned} \text{PotrebneBalikyNieKDispoziciiRO} &\hat{=} \\ \text{PotrebneBalikyNieKDispozicii} &\wedge \text{Uspech} \end{aligned}$$

Robustné operácie (3)

$$\begin{aligned} \text{InstalaciaRO} \hat{=} & (\text{Instalacia} \wedge \text{Uspech}) \vee \text{BalikJelInst} \\ & \vee \text{BalikNieJeReg} \vee \text{ChybajuBaliky} \end{aligned}$$
$$\begin{aligned} \text{OdinstalovanieRO} \hat{=} & (\text{Odinstalovanie} \wedge \text{Uspech}) \\ & \vee \text{BalikNieJelInst} \vee \text{BalikSaPouziva} \end{aligned}$$

Uplatnenie jazyka Z

Ďalšie mechanizmy jazyka Z

- V prednáške bol prezentovaný len zlomok mechanizmov jazyka Z
- Matematické dokazovanie vlastností špecifikácie
- Plné využitie predikátovej logiky
- Definovanie objektov generickou skratkou, axiomatickou definíciou a generickou definíciou
- Relácie a funkcie
- Postupnosti
- Multimnožiny
- Voľné typy
- Schéma ako typ
- Kompozícia schém

Nástroje na prácu v jazyku Z

- Jazyk Z od vzniku bol spojený s \LaTeX om, čo umožnilo automatizovanú prácu so špecifikáciou v jazyku Z
- \LaTeX štýl pre jazyk Z – `z-eves.sty`, upravená verzia `zed-csp.sty`
- \LaTeX syntax pre jazyk Z:
<http://staffwww.dcs.shef.ac.uk/people/A.Simons/z2sal/zdocs/Z%20of%20ZeTa.pdf>
- Nástroj Z/EVES na tvorbu a verifikáciu špecifikácií v jazyku Z
 - <http://fmt.cs.utwente.nl/courses/fmse/>
 - <http://www.uni-koblenz.de/~winter/Lehre/SS01/ZEVes/ZEVes.html>

Známe prípady uplatnenia jazyka Z

- Jazyk Z a formálne metódy určite neprevládajú, ale predsa sa používali a používajú
- CICS – Customer Information Control System, IBM
- D. Craigen et al. Industrial applications of formal methods to model, design and analyze computer systems: an international survey. Noyes Publications, 1995.

Implementácia špecifikácie

- Formalizmus jazyka Z pomáha zabezpečiť konzistentnosť špecifikácie
- Chyby však môžu vzniknúť pri jej implementácii
- Vzdialenosť jazyka Z od implementácie tomu žiaľ napomáha
- Metóda B je príklad formálneho prístupu bližšieho implementácii

Sumarizácia

Sumarizácia

- Jazyk Z ako príklad prostriedku na vyjadrenie formálnej špecifikácie
- Umožňuje štruktúrované vyjadrenie špecifikácie matematickým formalizmom
- Je možné formálne – matematickým dôkazom – overovať vlastnosti špecifikácie
- Operácie sú špecifikované prostredníctvom invariantov, predpokladov a dôsledkov – toto preniklo aj do programovania
- Podpora oddelenia záležitostí – aj pretínajúcich: výsledná operácia sa získa spojením parciálnych operácií
- Jestvujú aj ďalšie formálne prístupy, ktoré však boli ovplyvnené jazykom Z:
 - Object-Z
 - Metóda B (B-Method)
 - Alloy

Literatúra

- J. Woodcock and J. Davies. Using Z: Specification, Refinement, and Proof. Prentice-Hall International Series in Computer Science, 1996.
<http://www.usingz.com/>
- J. M. Spivey. The Z Notation: Reference Manual.
<http://spivey.oriel.ox.ac.uk/mike/zrm/>