

Sommersemester 2017

CASL**+ Z****+ TLA**

Übung

„Modellierung und Spezifikation“

Robert Prüfer

pruefer@informatik.hu-berlin.de

**Algebraische Spezifikation: Grundidee**

Spezifikation =

Signatur

+ Eigenschaften (beschrieben durch logische Formeln)

Semantik =

Menge von Strukturen

2

Beispiel einer CASL-Spezifikation

```
spec TOTAL_ORDER =
  sort s
  pred ___ < ___ : s × s
  ∀ x, y, z : s
  • x < y ∧ y < x ⇒ x = y
  • x < y ∧ y < z ⇒ x < z
  • x < y ∨ y < x
end
```

Passende Signatur: $\Sigma = (s; <:ss)$ Semantik: alle Σ -Strukturen, die drei Formeln erfüllen

3

Terme einer Spezifikation

```
spec NATURAL =
  free type Nat ::= o | suc(Nat)
end
```

Terme:

0	entspricht 0
suc(0)	entspricht 1
suc(suc(0))	entspricht 2
suc(suc(suc(0)))	entspricht 3
...	

4

Instanziierung einer parametrisierten Spezifikation

```
spec NATURAL =
  free type Nat ::= o | suc(Nat)
end
spec List[sort Elem] =
  op [] : List[Elem]
  op ___::___ : Elem x List[Elem] → List[Elem]
  op ___++___ : List[Elem] x List[Elem] → List[Elem]
end
```

- Was bedeutet **spec New = List[Natural]** ?
- Sorte *Nat* kann verwendet werden
- Alle Konstanten und Operationen aus *Natural* werden übernommen

5

Schlüsselwörter für Operationen

- comm**: Kommutativität
- assoc**: Assoziativität
- unit c**: Konstante c ist neutrales Element

Bsp.:

```
op k:s
op f : s * s -> s, assoc, unit k
```

6

Abgeleitete Axiome

- **%implied** hinter Formeln:
 - Behauptung:
 - Formel gilt und kann aus der Spezifikation abgeleitet werden
 - kann mit Theorembeweiser geprüft werden

Beispiel:

```
spec NATURAL =
  free type Nat ::= 0 | suc(Nat)
  ∀ x : Nat • ¬ 0 = suc(x)      %implied
end
```

7

Z

Δ -Operator:

- Hilfreich, um Veränderungen an einem gegebenen Schema auszudrücken

\exists -Operator:

- Hilfreich, um „Abfragen“ ohne Veränderung auszudrücken

8

TLA

Beispiel:

$$\sigma \begin{cases} x: 1\ 2\ 3\ 4\ \dots \\ y: 0\ 2\ 4\ 6\ \dots \end{cases}$$

$\sigma \models x > y$

$\sigma \not\models \Box x > y$

$\sigma \models x' = y'$

$\sigma \not\models \Box x' = y'$

$\sigma \models \Box y' = y+2$

9

CASL-Tool: HETS

- „Heterogeneous Tool Set“
 - Analyse von CASL-Spezifikationen
 - Einbindung von Theorembweisern
- Spezifikation in beliebigem Editor erstellen und mit der Endung .casl speichern
- Aufrufe
 - hets dateiname (Syntaxcheck)
 - hets -g dateiname (GUI)
- Als Theorembeweiser immer „SPASS“ nutzen (Rechtsklick auf rotem Knoten \rightarrow Prove \rightarrow [...])
- Dokumentation:
 - <http://bit.ly/hets-docs>

10

Logik-Operatoren in HETS

Logik-Operator Darstellung in HETS

\wedge	<code>/\</code>
\vee	<code>\ </code>
\Rightarrow	<code>=></code>
\Leftrightarrow	<code><=></code>
\neg	<code>not</code>
\forall	<code>forall</code>
\exists	<code>exists</code>
Kreuzprodukt	<code>*</code>
bildet ab auf	<code>-></code>

11