

Formale Techniken der Software-Entwicklung  
Übungsblatt 7  
Besprechung am 19.06.2015

**Aufgabe 1:**

In der Vorlesung wurde eine einfache Theorie für den Snark-Theorembeweiser vorgestellt, mit der sich Listen ähnlich wie in z.B. ACL2 beschreiben lassen. Insbesondere findet man in der Datei `snark-prog.lisp` im Folien-Repository die folgenden Aussagen:

```
(assert '(car? (cons ?x ?xs) ?x))  
(assert '(cdr? (cons ?x ?xs) ?xs))  
(assert '(member? ?x (cons ?x ?xs)))  
(assert '(implied-by (member? ?x (cons ?y ?xs))  
  (member? ?x ?xs))))
```

Hierbei sind `car`, `cdr` und `member` jeweils 2-stellige Prädikate und `cons` ein Funktionssymbol. Diese Aussagen können in reiner Logik-Schreibweise als *Horn-Klauseln* aufgefasst werden, d.h. als Klauseln, die jeweils nur ein positives Literal enthalten. Das ergibt die Klauselmenge  $\mathcal{K}^1$  :

$$\mathcal{K} = \{ \text{car}(\text{cons}(x, xs), x), \\ \text{cdr}(\text{cons}(x, xs), xs), \\ \text{member}(x, \text{cons}(x, xs)), \\ \text{member}(x, xs) \implies \text{member}(x, \text{cons}(y, xs)) \}$$

- (a) Geben Sie ausgehend von den Klauseln in  $\mathcal{K}$  zusätzliche Klauseln an, die die Semantik für folgenden Prädikate definieren:

`last(x, l):`

ist wahr, wenn  $x$  das letzte Element in der Liste  $l$  ist.

`butlast(p, l):`

ist wahr, wenn  $p$  das Präfix der Liste  $l$  ohne das letzte Element ist.

`palindrome(l):`

ist wahr, wenn die Liste  $l$  ein *Palindrom* darstellt, also z.B.  $[A, B, C, B, A]$  oder  $[A, B, B, A]$ .

- (b) Verwenden Sie den prädikatenlogischen Resolutionskalkül, um folgende Aussage zu beweisen:

$$\text{butlast}(\text{cons}(a, \text{nil}), \text{cons}(a, \text{cons}(b, \text{nil})))$$

---

<sup>1</sup>Eigentlich ist das vierte Element von  $\mathcal{K}$  keine Klausel, weil eine Implikation enthalten ist. Die Darstellung hier ist jedoch etwas lesbarer als die tatsächliche Klauselform.

**Aufgabe 2:**

Modellieren Sie ein einfaches Telefonbuch-System in ACL2. Dabei sollen folgende Funktionen unterstützt werden:

*add:*

fügt einen Eintrag für einen angegebenen *Namen* mit einer angegebenen *Telefonnummer* zu einem Telefonbuch hinzu.

*contains:*

überprüft, ob ein Telefonbuch einen Eintrag für einen bestimmten Namen enthält.

*lookup:*

gibt die Telefonnummer für einen angegebenen Namen zurück.

- (a) Spezifizieren Sie die oben angegebenen Features als ACL2-Funktionen.
- (b) Geben Sie ACL2-Theoreme an, um folgende Aussagen zu beweisen:
  1. Nach dem Hinzufügen eines Eintrags für einen bestimmten Namen mit *add* ergibt ein Test mit *contains*, dass tatsächlich ein Eintrag mit diesem Namen im Telefonbuch existiert.
  2. Nach dem Hinzufügen eines Eintrags für einen Namen *name* mit der Nummer *num* ergibt die Suche mit *lookup* für den Namen *name* dieselbe Telefonnummer *num*.