

Übungsblatt 1

Softwaretechnik II

October 23, 2008

Installieren Sie PROB und den BE4-Editor auf Ihrem Rechner. Beide Werkzeuge finden Sie auf der folgenden Webseite: <http://www.stups.uni-duesseldorf.de/ProB/>

Aufgabe 1 Seien P , Q und R Aussagen. Vervollständigen Sie die folgenden logischen Äquivalenzen und geben Sie ihre Namen an. Beispiel $P \wedge Q$: $P \wedge Q = Q \wedge P$ Kommutativität von \wedge .

a.) $P \vee P =$

b.) $P \wedge \neg P =$

c.) $P \wedge \text{true} =$

d.) $\neg(\neg P) =$

e.) $P \vee (Q \wedge P) =$

f.) $(P \wedge Q) \wedge R =$

g.) $P \vee (Q \wedge R) =$

h.) $\neg(P \vee Q) =$

i.) $\neg P \vee Q =$

j.) $P \Leftrightarrow Q =$

Aufgabe 2 Welche der folgenden Aussagen sind für beliebige Instanzen der freien Variablen wahr (d.h. worauf auch immer sich die freien Variablen beziehen).

a.) $(\text{member} \subseteq \text{list} \wedge \text{new} \in \text{list}) \Rightarrow (\text{member} \cup \{\text{new}\} \subseteq \text{list})$

b.) $(\text{member} \subseteq \text{list} \wedge \text{new} \in \text{list}) \Rightarrow \text{new} \in \text{member}$

- c.) $new \in list \Rightarrow \{new\} \subseteq list$
- d.) $\forall n.(n \in member \Rightarrow \exists s.(s \in \mathcal{P}(member) \wedge n \in s))$
- e.) $\forall n.(n \in member \Rightarrow \exists s.(s \in \mathcal{P}(member) \wedge s \neq \{\} \wedge n \notin s))$
- f.) $serve < next \Rightarrow (serve \leq next)[serve + 1/serve]$
- g.) $member \cap list = \{\} \wedge new \notin member \Rightarrow (member \cap list = \{\})[list \cup \{new\}/list]$

Aufgabe 3 Gegeben sei die folgende Maschine.

```

MACHINE          Ticket

VARIABLES        serve, next

INVARIANT        serve : NAT & next : NAT & serve <= next

INITIALISATION   serve, next := 0,0

OPERATIONS
  ss <-- serve_next =
    PRE serve < next
    THEN ss,serve := serve+1,serve+1
    END ;
  tt <-- take_ticket
    tt,next := next,next+1
END

```

- a.) Geben Sie die Maschine in PROB ein und fügen Sie eine Abfrageoperation hinzu, die die Anzahl der noch wartenden Personen zurückgibt.
- b.) Fügen Sie eine weitere Zustandsvariable, *record*, zur Maschine hinzu. Diese Variable soll angeben, wieviele Personen seit der letzten *serve_next* Operation ein Ticket genommen haben. Die Variable wird also inkrementiert, wenn eine Person ein Ticket nimmt und auf Null zurückgesetzt, wenn eine Person bedient wird. Geben Sie auch eine Abfrageoperation an, die den Wert der Variablen *record* zurückgibt.

- c.) Verändern Sie die Maschine *Ticket* so, dass Sie eine Operation *set_limit* hinzufügen können, die eine natürliche Zahl n als Eingabewert hat, sodass die Gesamtzahl der vergebenen Tickets (d.h. der Wert von *next*) nicht größer als dieser Wert sein kann. Das Initiallimit soll 10 sein.

Überprüfen Sie die Funktionalität Ihrer Maschine mit Hilfe von ProB.

Aufgabe 4 Berechnen Sie die folgenden schwächsten Vorbedingungen.

- a.) $[serve := serve + new](serve \leq next)$
 b.) $[serve, next := serve + new, next + 1](serve \leq next)$
 c.) $[x, y := 3, 11](\forall x.(x \in \mathbb{N} \Rightarrow y < x^2 + 4))$
 d.) $[IF\ x > 7\ THEN\ x := x - 4\ ELSE\ x := x + 3\ END](x > 12)$
 e.) $[IF\ x > 7\ THEN\ x, y := x - 4, x + 2\ ELSE\ y := y + 3\ END](y > x)$
 f.)

$$\left[\begin{array}{l} \text{CASE } i \text{ mod } 2 \text{ OF} \\ \text{EITHER } 0 \text{ THEN } ans := even \\ \text{OR } 1 \text{ THEN } ans := odd \\ \text{END} \end{array} \right] (ans = even)$$

Aufgabe 5 Gegeben sei ein Parkhaus mit 120 Parkplätzen. Geben sie eine abstrakte Maschine an, die ein Kontrollsystem für ankommende Fahrzeuge spezifiziert. Das System soll die Anzahl der im Parkhaus befindlichen Fahrzeuge festhalten und folgende Operationen zur Verfügung stellen.

- **enter**, registriert wenn ein Fahrzeug ankommt. Ist das Parkhaus voll, dann dürfen keine weiteren Fahrzeuge eingelassen werden.
- **leave**, registriert wenn ein Fahrzeug das Parkhaus verlässt.
- **nn** \leftarrow **query**, gibt die Anzahl der im Parkhaus befindlichen Fahrzeuge zurück

Hinweis: Zum Animieren und Testen ihres Modells mit ProB können Sie die maximale Zahl der Fahrzeuge im Parkhaus auch kleiner wählen.