

Aufgabe 1 : (5+5 Punkte)

Wir erweitern die Programmiersprache WHILE um das Konstrukt

`repeat π until b end`

Geben Sie eine

1. SOS-Regel [*rep_sos*]
2. NS-Regel [*rep_ns*]

an, die diesem Konstrukt die “gewohnte” Semantik gibt, ohne bei der Angabe dieser Regeln die Existenz des while-Konstrukts in WHILE auszunutzen.

Aufgabe 2 : (5+5 Punkte)

Wir erweitern die Programmiersprache WHILE um das Konstrukt

`for $x := a_1$ to a_2 do π od`

Definieren Sie die SO- und N-Semantik für dieses Konstrukt so, dass es der “gewohnten” Bedeutung der for-Schleife entspricht.

Aufgabe 3 : (5+5 Punkte)

Beweisen Sie folgendes Lemma aus der Vorlesung zur Konstruktion wohlfundierter Ordnungen:

Sind $(W_1, <_1)$ und $(W_2, <_2)$ zwei wohlfundierte Ordnungen, dann sind auch

- $(W_1 \times W_2, <_{com})$ mit *komponentenweiser* Ordnung definiert durch

$$(m_1, m_2) <_{com} (n_1, n_2) \text{ gdw. } m_1 <_1 n_1 \wedge m_2 <_2 n_2$$

- $(W_1 \times W_2, <_{lex})$ mit *lexikographischer* Ordnung def. durch

$$(m_1, m_2) <_{lex} (n_1, n_2) \text{ gdw.}$$

$$(m_1 <_1 n_1) \vee (m_1 = n_1 \wedge m_2 <_2 n_2)$$

wohlfundierte Ordnungen.

Abgabe: Dienstag, den 26.04.2016, vor der Vorlesung.