

9. Praktische Übung zur

Logische und funktionale Programmierung

GRUPPENÜBUNGEN:

(G 32)

Formuliere für das argumentlose Prädikat `entweder_oder` 2 Regeln! Das Prädikat `entweder_oder` soll durch genau eine der beiden Regeln ableitbar sein. Dabei soll das Prädikat `entweder_oder` durch die erste Regel ableitbar sein, wenn die Prädikate `a` und `b` ableitbar sind. Kann das Prädikat `a` nicht abgeleitet werden, soll das Prädikat `entweder_oder` durch die 2. Regel - mit dem Prädikat `c` im Regelrumpf - abgeleitet werden. Schlägt die Ableitbarkeits-Prüfung des Prädikats `b` in der ersten Regel fehl, soll es nicht möglich sein, das Prädikat `entweder_oder` mit der 2. Regel abzuleiten. Die Ableitbarkeits-Prüfung des Prädikats `entweder_oder` soll fehlschlagen, wenn entweder das Prädikat `b` oder das Prädikat `c` nicht abgeleitet werden kann.

(G 33)

Gegeben sei folgendes Programm:

`p(X) :- ¬q(X) .`

`p(X) :- ¬r(X) .`

`q(X) :- ¬p(X) .`

`q(a) .`

`r(b) .`

Wodurch können wir erreichen, dass die Anfrage `p(a)` mit `yes` beantwortet wird?

(G 34)

- a) Wie können wir im folgenden Programm den `cut` durch den Einsatz des Standard-Prädikats `\+` ersetzen?

`a :- ¬b, !, c .`

`a :- ¬d .`

Wie ist die logische Interpretation dieser Klauseln?

- b) Wie ist die logische Interpretation, wenn die Reihenfolge der Klauseln vertauscht wird?
- c) Wie können wir die folgende Regel durch den Einsatz der Standard-Prädikate `cut` und `fail` ersetzen?

`a :- ¬\+(b) .`