

Automatisierte Logik und Programmierung II

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Sommersemester 2006

Blatt 1 — Abgabetermin: —

Aufgabe 1.1 (ML Spielereien)

Programmieren Sie die folgenden elementaren ML Funktionen

1.1-a `member: * -> * list -> bool`

testet, ob ein Element `x` in einer Liste `l` vorkommt.

1.1-b `select: int -> * list -> *`

bestimmt das `n`-te Element einer Liste `l`.

1.1-c `replicate: * -> int -> * list`

erzeugt bei Eingabe von `x` und `n` die `n`-elementige Liste `[x; ... x]`

1.1-d `find: (*-> bool) -> * list -> *`

findet bei Eingabe einer Funktion `f` und einer Liste `l` das erste Element `x` von `l`, für das `f(x)=true` ist.

1.1-e `map: (*->**) -> * list -> ** list`

wendet eine Funktion `f` auf alle Elemente einer Liste `l` an.

1.1-f `lookup: (*#**) list -> * -> **`

findet bei Eingabe eines Labels `x` das erste Element `y`, für das `(x,y)` in einer Liste `l` liegt.

Aufgabe 1.2

In Einheit 9 hatten wir die Curry-Howard Isomorphie zwischen den logischen Regeln und den Regeln der Typentheorie besprochen und die Logikoperatoren als definitorische Erweiterung der Sprache der Typentheorie eingeführt. Wir wollen nun auch das Inferenzsystem der Typentheorie um logische Inferenzregeln erweitern.

1.2-a Programmieren Sie mit Hilfe der Taktik `Unfold` ein Tactical `CC: tactic -> tactic`, welches eine angegebene Taktik zur Anwendung bringt, nachdem zuvor eine Definition in der Konklusion einer Sequenz aufgelöst wurde.

Programmieren Sie unter Verwendung der Taktiken `Fold` und `Unfold` ein Tactical `CH: (int->tactic) -> int -> tactic`, welches dasselbe für Taktiken leistet, die auf Hypothesen operieren.

1.2-b Implementieren Sie mit Hilfe von `CC` und `CH` die Regeln der Prädikatenlogik als Taktiken.

Hinweis: Bedenken Sie beim Auffalten, daß Eliminationsregeln oft eine Kopie der eliminierten Hypothese stehen lassen. hinterlassen, Universenlevel nicht anzugeben sind, und – außer bei den Quantoren – keine Variablendeklarationen hinterlassen.

Verwenden Sie die Variable `Umax` um das maximale Universenlevel zu bestimmen, wo dies nötig ist, und die vordefinierte Taktik `Wf`, um Wohlgeformtheitsziele beweisen zu lassen. Mit der Taktik `mkInv i` können Sie die deklarierte Variable der Hypothese `i` unsichtbar machen (es entsteht eine Variable `%x...`, die in Deklarationen vom Display-Mechanismus unterdrückt wird).

Aufgabe 1.3

In Einheit 12 haben wir die Tacticals `Try`, `Progress` und `Repeat` beschrieben. Geben Sie eine ML-Implementierung dieser Tacticals an.