
Compilerbau

Blatt 11

Abgabe: 23.1.2002

1. [10 Punkte] Wegen der linearen Fädelung der Continuations ist es nicht nötig, die Continuation-Objekte in der Umgebung abzuspeichern. Stattdessen kann die momentane Continuation durch eine Kette von Continuation-Objekten repräsentiert werden. Ändern Sie den Interpreter `Cps-int-closures` so ab, daß die momentane Continuation durch eine Kette aus Continuation-Objekten repräsentiert wird, deren Anfang den Auswertungsfunktionen als Parameter übergeben wird. Passen Sie auch das Exception-Handling an, indem sie die Handler-Objekte um Verweise in die Kette erweitern.
2. [20 Punkte] Ersetzen Sie in den Auswertungsfunktionen im Interpreter `Cps-int-closures` die Parameter momentane Umgebung, momentane Continuation und momentaner Exception-Handler durch globale Variablen, welche später durch Register repräsentiert werden.
3. [10 Punkte] Die Codeerzeugung vereinfacht sich, wenn alle Anwendungen offener Primitiva *vollständig* sind, d.h. daß ein n -stelliges offenes Primitivum p auch immer in der Form $p e_1 \dots e_n$ angewendet wird. Dies ist nicht immer der Fall. Schreiben Sie eine Funktion, welche ein Lambda- oder Cps-Programm in ein äquivalentes Programm transformiert, in dem alle Anwendungen offener Primitiva vollständig sind. Verwenden Sie das Modul `Camlprim` um Informationen über die Primitiva zu erhalten.
4. [20 Punkte] Geben Sie eine Semantik für den Lambda-Kalkül an, welche Primitiva für das Erzeugen und Setzen von Referenzen bereitstellt.

Hinweis: Es reicht nicht, α zu erweitern. (Warum?)

Anleitung: Schreiben Sie zunächst eine Semantik in CPS. Erweitern Sie `Val` um Adressen und ändern die Semantik so, daß neben der Continuation auch ein *Speicher*, eine Abbildung von Adressen auf Werten, verwaltet wird.

Wie läßt sich an der Semantik ablesen, daß die Implementierung von Referenzen durch Seiteneffekte korrekt ist?