Universität Bremen

Korrekte Software: Grundlagen und Methoden SoSe 2016

Serge Autexier
Christoph Lüth

9. Übungsblatt

Ausgabe: 27.06.16
Abgabe: 07.07.16

In diesem letzten, praktischen Aufgabenblatt wollen wir die im letzten Aufgabenblatt theoretisch betrachtete Modellierung von Zeigern in unserem Werkzeug implementieren. Dazu müssen wir die semantische Modellierung so erweitern, dass sie Lokationen und die *Read/Update*-Operationen modellieren.

9.1 *Ungleichheiten* *10 Punkte*

Das Rahmenwerk stellt eine Funktion stateSimp bereit, welche die Zustandsvereinfachungsregeln implementiert:

$$read(upd(\sigma, l, v), l) = v \tag{1}$$
$$l \neq m \longrightarrow read(upd(\sigma, l, v), m) = read(\sigma, m) \tag{2}$$
$$upd(upd(\sigma, l, v), l, w) = upd(\sigma, l, w) \tag{3}$$
$$l \neq m \longrightarrow upd(upd(\sigma, l, v), m, w) = upd(upd(\sigma, m, w), l, v) \tag{4}$$

Für die Regeln (3) und (4) muss die *Ungleichheit* zweier Ausdrücke, welche Lokationen denotieren, bewiesen werden. Dazu gelten folgende Regeln:

- Zwei durch ungleiche Variablen denotierte Lokationen sind immer ungleich;
- Zwei Lokationen unterschiedlichen Typs sind immer ungleich;
- Jede Lokation ist zu ihrer eigenen Adresse ungleich (folgt aus dem vorherigen);
- Wenn `i` und `j` zwei Feldnamen (einer Struktur) sind, dann sind die Selektionen `e.i` und `f.j` immer ungleich (*split heap property*);

Implementieren Sie diese Regeln in der Funktion proveInequality(t1: Term, t2: Term): Boolean.

Passen Sie jetzt Ihre Funktion *weakestPrecondition* so an, dass sie mit der neuen Zustandsmodellierung arbeitet.

9.2 *Stärkste Vorbedingungen* *10 Punkte*

Implementieren Sie zwei Funktionen asp und svc, welche die approximative stärkste Nachbedingung und dazu gehörenden Verifikationsbedingungen berechnen.