



Verifikation nebenläufiger Programme

Sommersemester 2007

Serie 1

Mi 4. April 2007

Thema: Garbage Collection - Mutual Exclusion

Ausgabetermin: Mi 4. April 2007

Abgabe: Mo 16. April 2007 10:00 Uhr

Aufgabe 1 Tragen Sie sich für die Übung in die StudiDB ein.

Aufgabe 2 (5 + 1 Punkte) Im Kontext von Abschnitt 1.4.3 (S. 13ff.) sei der folgende Garbage-Collection-Algorithmus gegeben:

$$\begin{aligned} M_3 &\equiv \langle \text{redirect edge to the new target} \rangle; \\ &\quad \langle \text{blacken the new target} \rangle \\ C_3 &\equiv C_2 \end{aligned}$$

1. Beweisen Sie informal die Korrektheit des Algorithmus, indem Sie die Argumente zur Korrektheit des „größeren“ Algorithmus in Abschnitt 1.4.3 modifizieren und eine neue (schwächere) Invariante finden, die von den Aktionen des neuen Mutators M_3 nicht beeinflusst wird.

Hinweis: Beachten Sie, daß man eine Kante nur auf einen von einer Wurzel aus erreichbaren Knoten lenken kann (*add* und *remove* werden über den *nil*-Knoten dargestellt). Die Antworten auf folgende Fragen können hilfreich sein:

- Falls im aktuellen Collector-Durchlauf noch kein weiterer Knoten schwarz gefärbt wurde, wieviele Kanten kann es maximal geben, die von einem bereits besuchten, schwarzen Knoten zu einem weißen Knoten führen?
- Ist ein solcher weißer Knoten noch durch andere Knoten erreichbar?
- Was ist die nächste Aktion des Mutators?

2. Bleibt der Algorithmus auch korrekt wenn zwei M_3 parallel zu C_3 arbeiten? Begründen Sie Ihre Behauptung.

Aufgabe 3 (2 Punkte) Begründen Sie, warum in Zeile 11 von Szymanskis Algorithmus (Example 1.6, S. 23f) ein Test nur auf $flag[j] < 2$ nicht ausreicht. Geben Sie also an, welches Problem aufgrund des Weglassens von $flag[j] > 3$ auftritt.



Verifikation nebenläufiger Programme

Sommersemester 2007

Serie 1

Mi 4. April 2007

Thema: Garbage Collection - Mutual Exclusion

Ausgabetermin: Mi 4. April 2007

Abgabe: Mo 16. April 2007 10:00 Uhr

Aufgabe 1 Tragen Sie sich für die Übung in die StudiDB ein.

Aufgabe 2 (5 + 1 Punkte) Im Kontext von Abschnitt 1.4.3 (S. 13ff.) sei der folgende Garbage-Collection-Algorithmus gegeben:

$$\begin{aligned} M_3 &\equiv \langle \text{redirect edge to the new target} \rangle; \\ &\quad \langle \text{blacken the new target} \rangle \\ C_3 &\equiv C_2 \end{aligned}$$

1. Beweisen Sie informal die Korrektheit des Algorithmus, indem Sie die Argumente zur Korrektheit des „größeren“ Algorithmus in Abschnitt 1.4.3 modifizieren und eine neue (schwächere) Invariante finden, die von den Aktionen des neuen Mutators M_3 nicht beeinflusst wird.

Hinweis: Beachten Sie, daß man eine Kante nur auf einen von einer Wurzel aus erreichbaren Knoten lenken kann (*add* und *remove* werden über den *nil*-Knoten dargestellt). Die Antworten auf folgende Fragen können hilfreich sein:

- Falls im aktuellen Collector-Durchlauf noch kein weiterer Knoten schwarz gefärbt wurde, wieviele Kanten kann es maximal geben, die von einem bereits besuchten, schwarzen Knoten zu einem weißen Knoten führen?
- Ist ein solcher weißer Knoten noch durch andere Knoten erreichbar?
- Was ist die nächste Aktion des Mutators?

2. Bleibt der Algorithmus auch korrekt wenn zwei M_3 parallel zu C_3 arbeiten? Begründen Sie Ihre Behauptung.

Aufgabe 3 (2 Punkte) Begründen Sie, warum in Zeile 11 von Szymanskis Algorithmus (Example 1.6, S. 23f) ein Test nur auf $flag[j] < 2$ nicht ausreicht. Geben Sie also an, welches Problem aufgrund des Weglassens von $flag[j] > 3$ auftritt.