

Formale Grundlagen der Informatik 3

Prof. Dr. Christoph Walther / Visar Januzaj, Nathan Wasser
Technische Universität Darmstadt — Wintersemester 2011/12

Praktische Übung 4

Version 1 vom 13.01.2012

Diese Übung wird in der **KW 05 (30.01.–03.02.2012)** testiert. Reichen Sie dazu Ihre mit **veriFun 3.4** erstellte **.vf**-Lösungsdatei bis spätestens **Sonntag, 29.01.2012 23:59 Uhr** online im Kursportal ein. Später eintreffende Lösungen können nicht akzeptiert werden.

Es gelten die gleichen Testattermine bzw. Praktikumsgruppen G1-G90 wie in der letzten Testatwoche.

Bei Fragen und Problemen können Sie neben den Poolbetreuungszeiten und Sprechstunden der Tutoren auch das Forum zur Veranstaltung nutzen.

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Mit der Abgabe einer Lösung bestätigen Sie, dass Sie der alleinige Autor des gesamten Materials sind. Bei Unklarheiten zu diesem Thema finden Sie weiterführende Informationen unter <http://www.informatik.tu-darmstadt.de/Plagiarism> oder sprechen Sie Ihren Betreuer an.

Praktische Übung 4.1 (Terminierung des Davis-Putnam-Verfahrens)

In der Praktischen Übung 3 haben Sie die Korrektheit und Vollständigkeit des Davis-Putnam-Verfahrens bewiesen. Die Terminierung des Verfahrens war in der vorgegebenen Datei automatisch durch **veriFun** bewiesen. Die Terminierung einer leicht veränderten Implementierung des Davis-Putnam-Verfahrens kann jedoch nicht automatisch bewiesen werden. Beweisen Sie nun, dass die Prozedur *DavisPutnam* in der Datei **Praxis4.vf** terminiert.

Praktische Übung 4.2 (Terminierung eines Normalisierungsverfahrens für If-Ausdrücke)

Im Rumpf einer \mathcal{L} -Prozedur dürfen Bedingungen in bedingten Ausdrücken keine bedingten Ausdrücke enthalten. In Kapitel 5 auf Folie 9 wird implizit ein Verfahren angegeben, wie man aus einem beliebig verschachtelten If-Ausdruck einen sogenannten *normalisierten* If-Ausdruck gewinnt: Man ersetzt einen Teilausdruck der Form $\text{if}\{\text{if}\{a, b, c\}, d, e\}$ durch $\text{if}\{a, \text{if}\{b, d, e\}, \text{if}\{c, d, e\}\}$, bis keine solchen unerlaubten Teilausdrücke mehr vorkommen.

Die Prozedur *normalize* in der Datei **Praxis4.vf** implementiert diese Normalisierung von If-Ausdrücken. Zeigen Sie die Terminierung dieser Prozedur.

Hinweis: Vergleichen Sie die Prozedur *normalize* mit der Prozedur *flatten* (Kapitel 8, Beispiel 7). Beiden Prozeduren ist gemeinsam, dass sie eine Normalform herstellen; es gilt $\forall x : \text{sexpr } \text{flatten}(\text{flatten}(x)) = \text{flatten}(x)$ sowie $\forall x : \text{Conditional } \text{normalize}(\text{normalize}(x)) = \text{normalize}(x)$. Zur Herstellung der Normalform werden in beiden Prozeduren jeweils zwei Fälle unterschieden: ein „Umordnungsfall“ und ein „trivialer Fall“. Machen Sie sich zunächst klar, warum der in Beispiel 7 skizzierte Lösungsansatz für den Terminierungsbeweis von *flatten* erfolgreich ist: Was bleibt gleich groß? Was wird kleiner?

Der Terminierungsbeweis für *normalize* funktioniert nach dem gleichen Prinzip. Machen Sie sich anhand von Beispielen für *normalize* klar: Was bleibt gleich groß? Was wird kleiner? Formulieren Sie anschließend geeignete Prozeduren, mittels derer Sie Maßterme für *normalize* angeben können. (Bei *flatten* wurde nur die Prozedur *#nodes* benötigt, um die Maßterme für *flatten* anzugeben. Für *normalize* reicht dies nicht – hier benötigen Sie zwei Prozeduren.)