



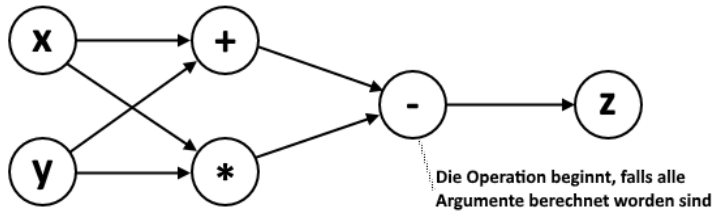
Lösungen zum Übungszettel 8

Aufgabe 1:

- a) Datenflussrechner versuchen, die Möglichkeiten der Parallelverarbeitung ihrer Rechenaufträge durch das nebenläufige Ausführen einer Vielzahl von Threads auszunutzen. Der Datenflussrechner basiert auf dem Datenflussprinzip, welches die Ablaufsteuerung gänzlich anders organisiert. Einzig die Verfügbarkeit der Operanden löst die Ausführung einer Maschinenoperation auf diesen Operanden aus. Die Resultate können dann wieder zur Ausführung anderer Operationen führen. Deshalb wird dieses Operationsprinzip auch als datengesteuert oder datengetrieben (Data-Driven) bezeichnet. Nach dem Operationsprinzip können mehrere Threads von Maschinenbefehlen, die einen sogenannten Datenflussgraphen beschrieben, verarbeitet werden und einen hohen Grad an Nebenläufigkeit besitzen. Daher sind keine Befehlszähler und globale Variablen oder zentraler Speicher verfügbar im Vergleich zu der von-Neumann-Architektur. Zudem können Befehlsfolgen bei parallelen Kontrollflussprinzipien, die nach dem von-Neumann-Prinzip aufgeführt werden, durch explizite parallele Kontrollstrukturen (z.B. fork, ..., join) verknüpft sein und parallel zueinander aufgeführt werden.
- b) Zur Klassifikation von Datenflussrechnern existieren unterschiedliche Ansätze. Die am häufigsten verwendete Klassifikation basiert auf dem in Datenflussrechnern verwendeten Prinzip der Synchronisation von Operationen. Dieser Ansatz unterscheidet statische und dynamische Datenflussrechner. In der Klasse der statischen Datenflussrechner findet nur Parallelismus feiner Granularität Anwendung, während die Klasse der dynamischen Datenflussrechner aufgrund der Granularität des unterstützten Parallelismus weiter differenziert werden kann. Dabei werden Datenflussarchitekturen, die Parallelismus größerer Granularität unterstützten unterschieden. Unter feiner Granularität wird hierbei Parallelismus auf der Instruktionsebene verstanden. Architekturen, die Parallelität auf der Block- oder Prozessebene unterstützen, werden unter dem Sammelbegriff Datenfluss-von-Neumann-Hybridarchitekturen zusammengefasst.

Aufgabe 2:

a)

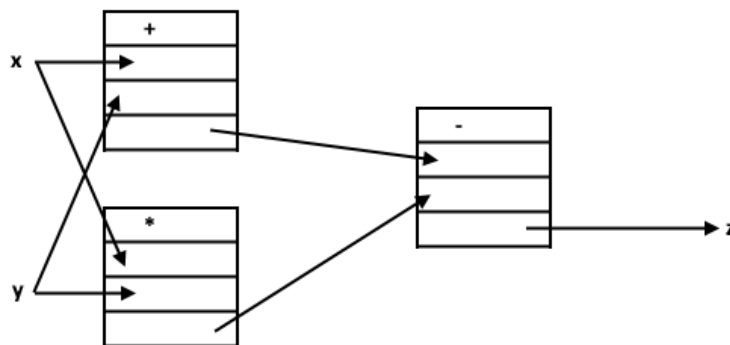


b) T1: (*, x, y; T3/l)

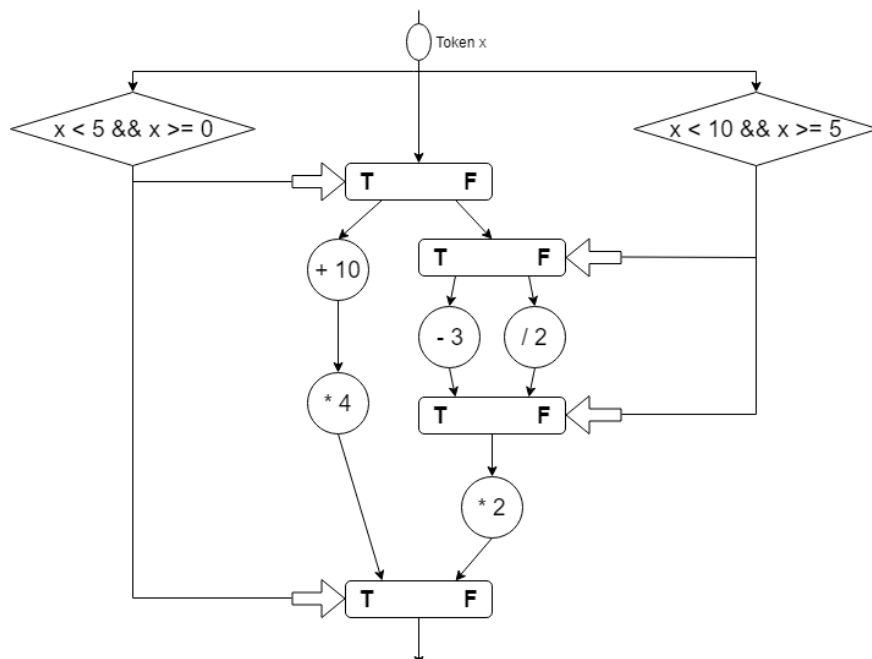
T2: (+, x, y; T3/r)

T3: (-, (), (); z)

c) Verkettete Listen, Immediate-Argumente, Opcodes und Ziel-Listen vom Compiler angelegt



Aufgabe 3:



Hinweis: Die Lösungen sollen in PDF-Form, bzw. Code bis zum Montag (10 Uhr) der jeweils folgenden Woche per Mail an mfriedrichs@techfak.uni-bielefeld.de abgegeben werden. Zu Beginn des nächsten Übungstermins werden diese in offener Runde vorgestellt und diskutiert.