

Concurrent Haskell und Software Transactional Memory

Aufgabenblatt

Sébastien Braun

Proseminar Fortgeschrittene Programmierung
4. Februar 2008

Aufgabe 1: Concurrent Haskell

Unter den vorgestellten Kommunikationsmechanismen befinden sich mit der `MVar` und dem `Chan` zwei Primitive, die entweder *höchstens einen* oder *beliebig viele* Werte aufnehmen können.

Implementieren Sie mit Hilfe der `MVar`-Abstraktion einen Typ `BChan α`. Ein `BChan α` soll wie ein `Chan α` funktionieren, mit der zusätzlichen Einschränkung, dass eine obere Schranke für die Anzahl der enthaltenen Datenelemente eingehalten wird.

Folgende Operationen sind zu definieren:

```
newBChan :: Int → IO (BChan α)
readBChan :: BChan α → IO α
writeBChan :: BChan α → α → IO ()
```

`newBChan n` soll einen neuen `BChan` mit der Kapazität `n` erzeugen.

`readBChan chan` soll einen Datenwert aus dem `BChan chan` lesen. Falls kein Datenwert vorhanden ist, sollte die Funktion auf einen Wert warten.

`writeBChan chan d` soll den Datenwert `d` an den `BChan chan` anhängen. Falls die Kapazität erschöpft ist, sollte die Funktion warten, bis Platz frei ist.

Sie dürfen bei der Implementierung einen unbeschränkten `Chan` verwenden.

Aufgabe 2: Software Transactional Memory

Verwenden Sie das Treiberprogramm `ProdCons.hs`.

Es handelt sich um ein Producer-Consumer-Szenario. Es laufen mehrere Producer-Prozesse. Diese sollen ihre Ergebnisse an einen Consumer-Prozess senden.

Jeder Producer sendet seine Ergebnisse auf einem eigenen TChan und der Consumer kann nur einen TChan lesen.

Schreiben Sie eine Funktion `merge`, die die Ergebnisse der Producer aus ihren jeweiligen TChans liest und an den TChan für den Consumer sendet.

Die Producer senden unregelmäßig. Versuchen Sie, eine Lösung zu finden, die jeden eingehenden Datenwert so schnell wie möglich weiterleitet.

TChans

TChans sind die STM-Variante von Chans. Sie funktionieren genauso, mit dem Unterschied, dass die Operationen `readTChan` und `writeTChan` heißen, und die Funktionen in der STM-Monade operieren.

Siehe <http://tinyurl.com/yt8dss> für Einzelheiten über TChans