ALP IV: Übung 1

Tutor: Christoph Beuck

Lisa Dohrmann (4130066), Adrian Neumann (4140810), Naja v. Schumde (4127652)

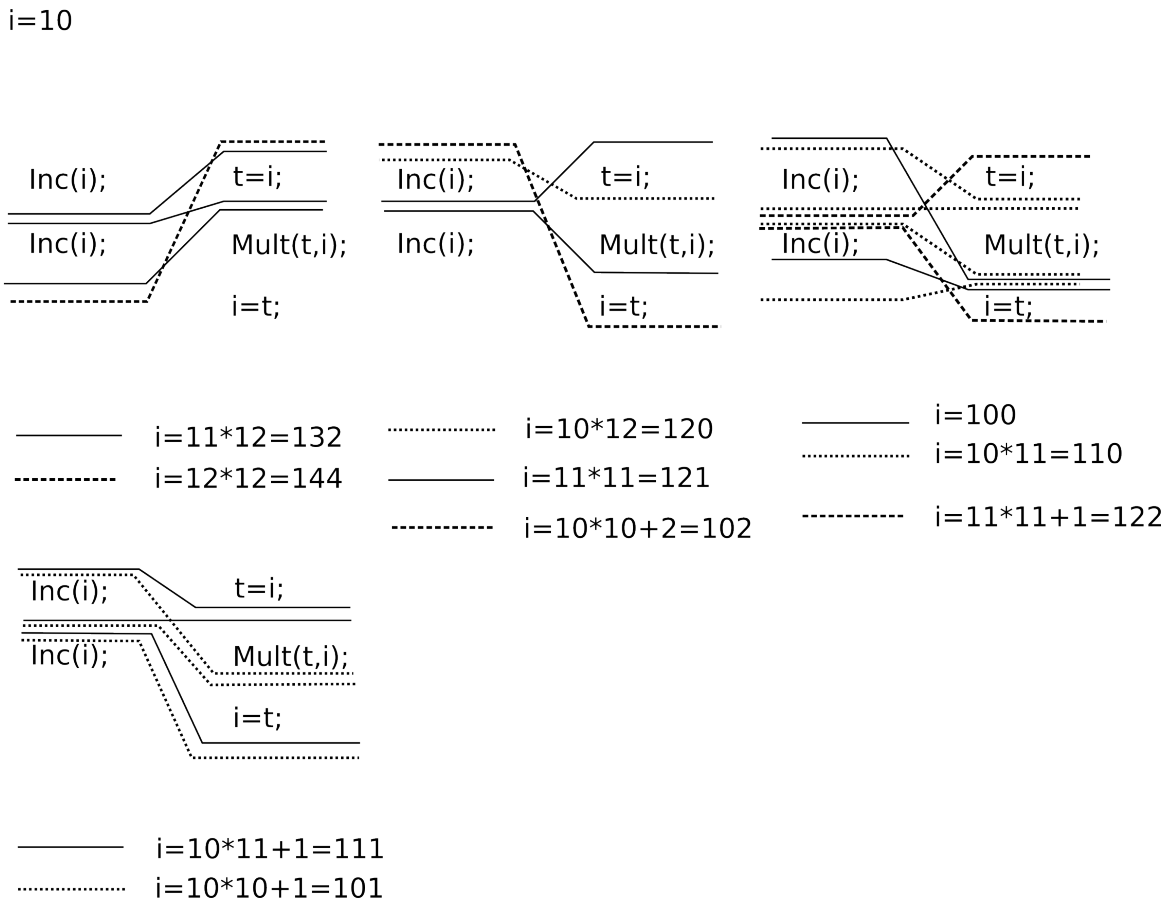
7. Mai 2008

Aufgabe 1

Je mehr voneinander abhängige Anweisungen nebenläufig ausgeführt werden, desto mehr verschiedene Ergebnisse können auch auftreten.
Wenden wir die Transformationen auf das Beispiel an, erhalten wir:

```
i = 10;
co INC(i); INC(i); || t = i; MUL(t,i); i = t; oc
```

Die folgenden Zeitschnitte produzieren nun 10 unterschiedliche Ergebnisse für `i`.



Aufgabe 2

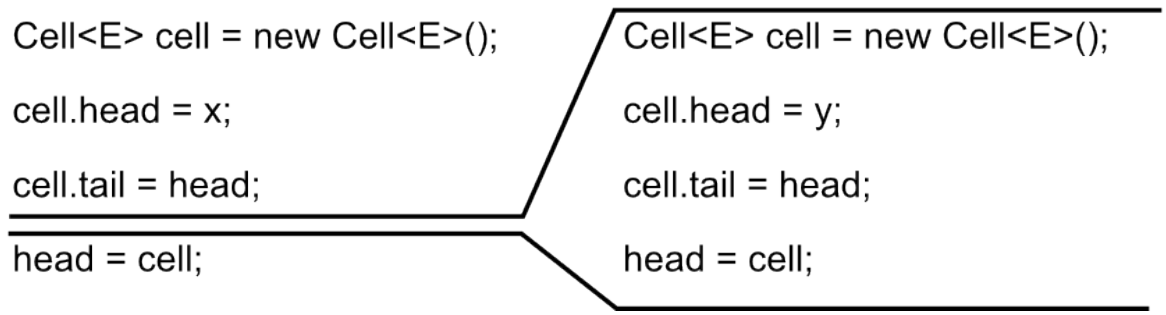
a) Sowas kann passieren, da die Parameterübergabe an `printSquare()` nebenläufig ausgeführt wird. Wenn also der Thread, der eigentlich `2 4` ausgeben sollte, so lange blockiert, bis `x` in der while-Schleife schon auf den Wert 3 erhöht wurde, wird 2x `printSquare(3)` aufgerufen.

b) Solch unerwartete Werte können auftreten, wenn `tabulate()` schon fertig und die auf dem Stack liegende Variable x ungültig geworden ist, wenn wir in `printSquare` darauf zugreifen wollen.

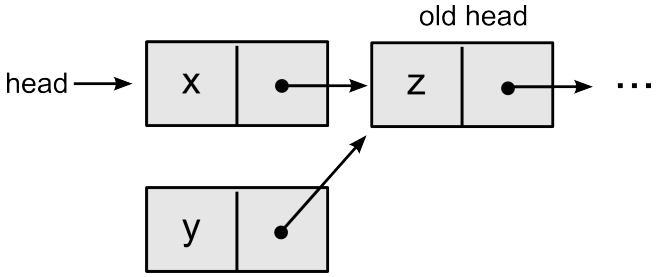
c) Ja, dieser Ansatz löst die genannten Probleme, denn es werden unabhängige Kopien der Werte erzeugt, die vom Elternprozess nicht mehr verändert werden können. Problematisch kann es lediglich werden, wenn man Referenzen übergibt. Dann können Referenzen in anderen Threads das referenzierte Objekt noch immer verändern und es kommt zum üblichen unvorhersagbaren Verhalten.

Aufgabe 3

Der folgende Zeitschnitt zeigt einen möglichen Ablauf der parallelen Ausführung von `objects.add(x)` (links) und `objects.add(y)` (rechts).

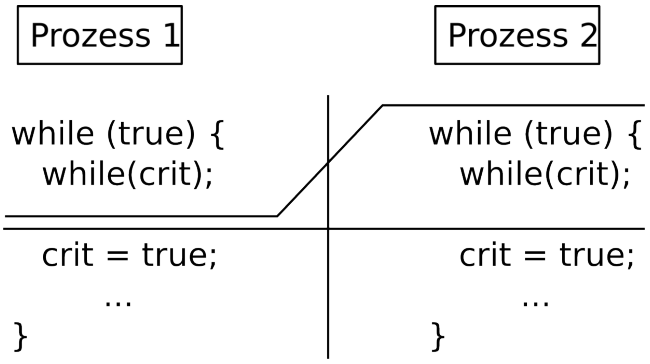


Bei diesem Zeitschnitt würde die Zelle mit `y` komplett verloren gehen, da `head` mit der Zelle mit Inhalt `x` überschrieben wird. Etwa so:



Aufgabe 4

Ansatz A: Hier kann es leicht zu Problemen kommen, denn ein gegenseitiger Ausschluss ist nicht garantiert. Man betrachte z.B. folgenden möglichen Zeitschnitt:



Hier treten beide Prozesse in den kritischen Bereich ein, da Prozess 2 die Variable `crit` liest, bevor Prozess 1 sie auf `true` gesetzt hat.

Ansatz B: Bei der zweiten Variante kann es zu einer Verklemmung kommen, wenn `crit1` und `crit2` auf `true` gesetzt werden, bevor wenigstens einer der beiden Prozesse die while-Schleife passiert hat. Beide Prozesse stecken damit in der „Warteschleife“ fest, bis das Programm abgebrochen wird.
Bei folgendem Zeitschnitt tritt genau die angesprochene Verklemmung ein.