Technische Universität Berlin

Neuronale Informationsverarbeitung

Tutoriumsvorbereitung 2

Prof. Dr. Klaus Obermayer und Mitarbeiter

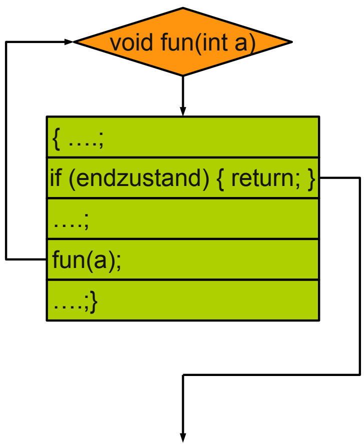
01.05.2017 - 03.05.2017

和C++的rekeursion又一样

Inhaltsverzeichnis

1 Java Wiederholung - Rekursion

- Rekursiver Aufruf einer Methode bedeutet: Sie ruft sich selbst wieder auf.
- Rekursive Methodenaufrufe benötigen:
  1. Einen Grundzustand (eine Abbruchbedingung), bei dem (der) sie aufhören.
  2. Einen rekursiven Funktionsaufruf
- das Ablaufdiagramm sieht folgendermaßen aus:



- Es gibt auch die indirekte Rekursion. Beispielsweise wenn in einer Methode `fun1(..)` eine Methode `fun2(..)` aufruft, in der wiederum die Methode `fun1(..)` aufgerufen wird.
- **Aufgabe:** Schreiben Sie eine rekursive Methode, die für ein `char[]`-Array testet, ob zwei aufeinanderfolgende Buchstaben gleich sind und genau dann true zurück liefert, wenn es zwei aufeinanderfolgende Buchstaben gibt, die gleich sind. Diese soll mit der folgenden Testklasse aufgerufen werden können.

```java
public class TestReihen {
   public static void main(String[] args) {
      char[] word = {'H','a','l','l','o'};
      System.out.println(Reihe.istdoppelt(word,0));
   }
}
```

计算循环的基础条件

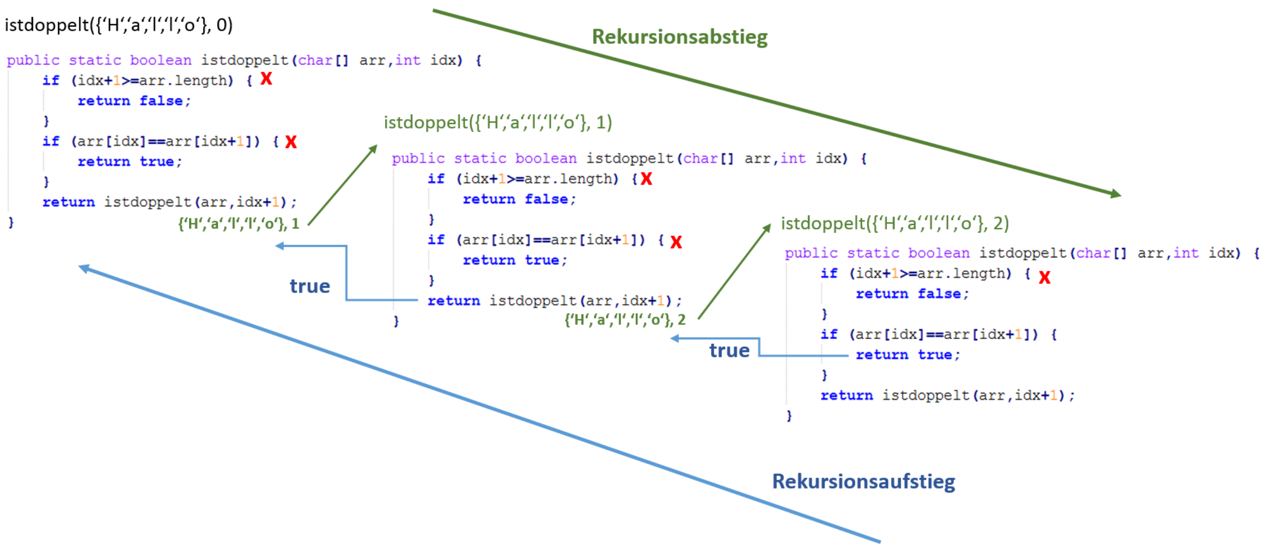
Lösung:

```java

   public class Reihe {

      public static boolean istdoppelt(char[] arr,int idx) {
         if (idx+1>=arr.length) {
            return false;
         }
         if (arr[idx]==arr[idx+1]) {
            return true;
         }
         return istdoppelt(arr,idx+1);
      }
```

return 是循环终止的条件



- In dem konkreten Beispiel kam es insgesamt zu 2 Folge-Methodenaufrufen. Die Rekursionstiefe betrug somit 2.
- Hinweis: Jeder Funktionsaufruf benutzt eigene lokale Variablen!
- Neben linearen Rekursionen gibt es weitere Kategorien von Rekursionen[1]. Baumartige/kaskadenförmige Rekursionen haben mehrere rekursive Aufrufe nebeneinander stehen. Ein Beispiel ist die Fibbonacci-Folge, deren Folgeglieder rekursiv über `fib(n-1)+fib(n-2)` berechnet werden.

2 KV-Tafeln

Zur Veranschaulichung logischer Funktionen eignen sich graphische Darstellungen bzw. Diagramme. Eine gute Verbildlichung bietet die sogenannte Karnaugh-Veitch-Tafel (KV-Tafel). Diese grafische Form bietet Voteile bei der Behandlung der Normalformen. Beispielsweise können Normalformen daraus zum Teil direkt abgelesen werden. Zudem dient das Verfahren zur Vereinfachung von Normalformen in einen minimalen logischen Ausdruck.

2.1 Erstellen von KV-Tafeln

Die KV-Tafel ist eine Matrix von Wahrheitswerten. Jedem Element der Matrix ist eindeutig eine Belegung der Variablen der Funktionen zugeordnet, die am Rand des Schemas eingetragen werden.

Vorgehen:

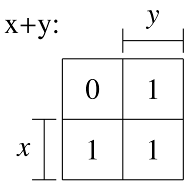
- Zunächst wird für die erste Variable je ein Feld für das positive und negative Literal erstellt.
- Für jede weitere Variable wird die komplette Tafel an der längsten Kante gespiegelt. Der Originalhälfte und der Spiegelhälfte werden wieder je das positive und das negative Literal der weiteren Variablen zugewiesen.

Nehmen wir zum Beispiel die bekannte Wahrheitstabelle einer Disjunktion:

[1] `https://de.wikipedia.org/wiki/Rekursion#Formen_der_Rekursion`

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

Die zugehörige KV-Tafel könnte somit folgendermaßen aussehen:



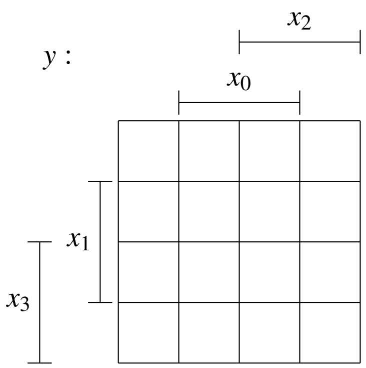
Jede Variable kommt dabei in einfacher und negierter Form vor. In den Zeilen/Spalten, wo sich die Balken befinden, wird den Variablen der Wahrheitswert 1 zugeordnet. Dadurch besteht die untere Zeile und die letzte Spalte nur aus Feldern mit dem Wert 1. Nur in dem Feld, bei dem beide Variablen negiert sind, wird der Wahrheitswert 0 eingetragen.

Jedes Feld repräsentiert eine mögliche Kombination der Eingangsvariablen (≙ Zeile in einer Wertetabelle). Die Anzahl der Felder lässt sich dabei folgendermaßen bestimmen: $2^{Anzahl der Eingangsvariablen}$. In dem vorigen Beispiel waren die Eingangsvariablen x und y. Die Gesamtzahl der Felder ist also $2^2 = 4$.

Aufgabe: Wie viele Felder hat die KV-Tafel für folgenden booleschen Ausdruck?

$$y = (\overline{x_0} \cdot \overline{x_2} \cdot \overline{x_3}) + (\overline{x_1} \cdot \overline{x_2} \cdot x_3) + (\overline{x_0} \cdot x_1 \cdot x_2 \cdot x_3) + (\overline{x_0} \cdot x_1 \cdot \overline{x_2} \cdot x_3)$$

Lösung:
4 Eingangsvariablen, also: $2^4 = 16$ Felder
Die zugehörige KV-Tafel hat also folgende Form:



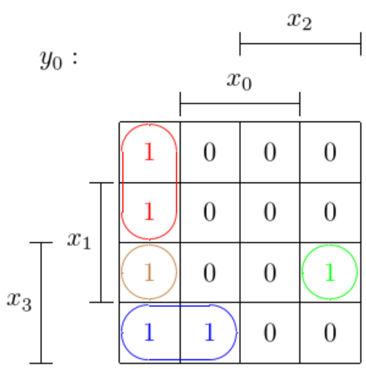
对叠思想 猜下面4×8的
(把4×4的当模吧)

2.1.1 Erstellen aus einer DNF

Für jeden Konjunktionsterm: In die Felder eine 1 eintragen, für die der Konjunktionsterm wahr ist. In alle anderen Felder wird anschließend eine 0 eingetragen.

Für: $y_0 = (\overline{x_0} \cdot \overline{x_2} \cdot \overline{x_3}) + (\overline{x_1} \cdot \overline{x_2} \cdot x_3) + (\overline{x_0} \cdot x_1 \cdot x_2 \cdot x_3) + (\overline{x_0} \cdot x_1 \cdot \overline{x_2} \cdot x_3)$

Es ergibt sich also folgendes KV-Diagramm (die Farben zeigen, welche Einsen aus welchen Konjunktionstermen entstanden sind):

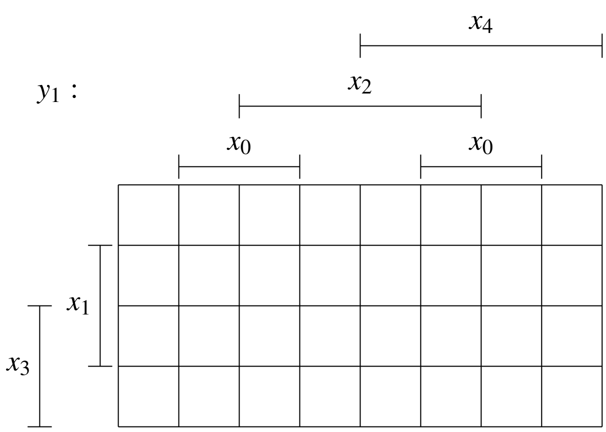


2.1.2 Erstellen aus einer KNF

Wie wir im letzten Abschnitt gesehen haben, geben die Balken diejenigen Bereiche an, wo der Wahrheitswert 1 zugeordnet wird. Bei der KNF werden Einträge mit dem Wahrheitswert 1 jedoch mit den negierten Variablen gebildet. Damit folgt für jeden Disjunktionsterm: In die Felder eine 0 eintragen, für die der Disjunktionsterm falsch ist. In alle anderen Felder wird anschließend eine 1 eingetragen.

Aufgabe: Erstellen Sie zu folgendem booleschen Ausdruck das zugehörige KV-Diagramm!

$$y_1 = (\overline{x_0} + \overline{x_2} + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + x_4) \cdot (\overline{x_1} + x_2 + x_3 + x_4)$$



Lösung: 反着来找 和 ubung1 里对应的

$$y_1 = (\overline{x_0} + \overline{x_2} + \overline{x_3}) \cdot (\overline{x_1} + \overline{x_2} + x_4) \cdot (\overline{x_1} + x_2 + x_3 + x_4)$$

y_1 :

x_4 x_2 x_0 x_0 x_1 x_3

1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	1

2.2 Vereinfachen von KV-Tafeln

Um von einer KV-Tafel zu einer zugehörigen DNF (bzw. KNF) zu kommen, müssen alle Einsen (bzw. Nullen) zusammengefasst werden. Dabei sind folgende Regeln zu beachten:

- die einzelnen Felder können zu Rechtecken zusammengefasst werden, wobei in einem Rechteck, was aus mehreren Feldern bestehen kann, sich nur Einsen (bzw. Nullen) befinden dürfen
- die Rechtecke müssen eine Größe von 2^i, $i \in \mathbb{N}$ haben 2, 4, 8, 16
- Rechtecke können auch über den Rand gehen (z.B. links raus aus der Tafel und rechts in der gleichen Zeile wieder rein)
- maximal große Rechtecke wählen
- Die verschiedenen Rechtecke dürfen sich überlappen

Zur DNF: Aus jedem Rechteck einen Konjunktionsterm erstellen. Die Konjunktionsterme durch Disjunktionen verknüpfen.
Zur KNF: Aus jedem Rechteck einen Disjunktionsterm erstellen. Die Disjunktionsterme durch Konjunktionen verknüpfen.

2.2.1 Übung: Vereinfachen zu einer DNF

Aufgabe: Erstellen Sie aus folgender KV-Tafel einen möglichst minimalen booleschen Ausdruck in DNF.

y_0 :

x_2 x_0 x_1 x_3

1	0	0	0
1	0	0	0
1	0	0	1
1	1	0	0

Lösung:

$$y_0 = (\overline{x_0} \cdot \overline{x_2}) + (\overline{x_1} \cdot \overline{x_2} \cdot x_3) + (\overline{x_0} \cdot x_1 \cdot x_3)$$

y_0 :

			x_0	x_0, x_2	x_2
		1	0	0	0
	x_1	1	0	0	0
x_3	x_1	1	0	0	1
x_3		1	1	0	0

2.2.2 Übung: Vereinfachen zu einer KNF

Aufgabe: Erstellen Sie aus folgender KV-Tafel einen möglichst minimalen booleschen Ausdruck in KNF.

y_0 :

			x_0	x_0, x_2	x_2
		1	0	0	0
	x_1	1	0	0	0
x_3	x_1	1	0	0	1
x_3		1	1	0	0

KNF 要反着找

Lösung:

$$y_0 = (x_1 + \bar{x}_2) \cdot (\bar{x}_0 + \bar{x}_1) \cdot (\bar{x}_0 + x_3) \cdot (\bar{x}_2 + x_3)$$

y_0 :

			x_0	x_0, x_2	x_2
		1	0	0	0
	x_1	1	0	0	0
x_3	x_1	1	0	0	1
x_3		1	1	0	0