

Informatik I Übung, Woche 48:
Nachbesprechung Aufgabe 10.1:
Implementation von `wechselgeld.pas`

Giuseppe Accaputo

26. November, 2015

Wechselgeld: Definition

Wechselgeld: Wir sind im Besitz von m Münzen, und möchten herausfinden, auf wie viele Arten man einen Betrag n mit den vorhandenen Münzen bezahlen kann.

Dynamisches Wechselgeld: Algorithmus

1. *Verankerung*: Ohne Münzen kann man den Betrag 0 genau einmal darstellen; Beträge grösser als 0 können nicht dargestellt werden ohne Münzen

Sei m_i die Münze, die im i -ten Schritt ($i > 0$) betrachtet wird.

2. Falls m_i grösser ist als der aktuell betrachtete Betrag, dann berechnet man, auf wie viele Arten man den Betrag x nur mit den Münzen m_1, \dots, m_{i-1} darstellen kann
3. Ansonsten gibt es zwei Fälle zu unterscheiden:
 - 3.1 Fall 1: man verwendet m_i nicht, und berechnet, auf wie viele Arten man den Betrag x nur mit den Münzen m_1, \dots, m_{i-1} darstellen kann
 - 3.2 Fall 2: man verwendet m_i mindestens einmal, und berechnet, auf wie viele Arten man den Betrag $x - m_i$ mit allen Münzen m_1, \dots, m_i darstellen kann

Dynamisches Wechselgeld: Setup

Für die dynamische Programmierung verwenden wir die Tabelle anz und setzen diese wie folgt auf:

- ▶ **Zeile:** Anzahl Münzen, welche in Betracht gezogen werden
- ▶ **Spalte:** Betrag
- ▶ **Eintrag** $anz(i, j)$ enthält die Anzahl Möglichkeiten um den Betrag j (Spalte j) mit den gegebenen Münzen m_1, m_2, \dots, m_i darzustellen (Zeile i)

Dynamisches Wechselgeld: Tabelle anz

- ▶ Anzahl Münzen: 7, d.h. m_1, \dots, m_7
- ▶ Max. Betrag: n

anz :=

	0	1	2	...	$n - 1$	n
0						
1						
2						
...						
7						

- ▶ **Wichtig:** Dimension der Tabelle ist $(7 + 1) \times (n + 1)$ wegen der *Verankerung*

Dynamisches Wechselgeld: Algorithmus

- ▶ Für alle Spalten j (Beträge) und Zeilen i (Anzahl Münzen) in der Tabelle tu folgendes:

1. Falls keine Münzen verwendet werden ($i = 0$):

- ▶ Falls $j = 0$:

```
anz[i, j] := 1;
```

- ▶ Falls $j > 0$:

```
anz[i, j] := 0;
```

2. Sei m_i die aktuelle Münze ($i > 0$). Falls $m_i > j$ dann:

```
anz[i, j] := anz[i-1, j];
```

3. Ansonsten:

- 3.1 Verwende m_i nicht:

```
a1 := anz[i-1, j];
```

- 3.2 Verwende m_i mindestens einmal:

```
a2 := anz[i, j - muenzen[i-1]];
```

- ▶ Speichere die Summe ab:

```
anz[i, j] := a1 + a2;
```

Dynamisches Wechselgeld: Münzen

- ▶ Die Münzen sind in einem Array `muenzen` gespeichert, das wie folgt definiert ist (siehe `wechselgeld.pas` auf der Vorlesungsseite):

```
muenzen : ARRAY [0..6] OF INTEGER;
```

- ▶ `muenzen` hat die Länge 7
- ▶ Die Münze m_i befindet sich an der Position $(i - 1)$ im Array (mit Indizes $0, \dots, 6$), d.h. `muenzen[i-1] := m_i`
 - ▶ `muenzen[0] := m_1`
 - ▶ `muenzen[1] := m_2`
 - ▶ ...
 - ▶ `muenzen[6] := m_7`

Dynamisches Wechselgeld: Münzen

- ▶ Zeilen in der Tabelle gehen von $i = 0, 1 \dots, 7$, wobei 7 der Anzahl Münzen entspricht und $i = 0$ der *Verankerung* entspricht (wir verwenden keine Münzen)
 - ▶ FOR-Schleife geht also von $i := 0$ TO Length(muenzen)
- ▶ `muenzen[i-1] := mi` wird verwendet, weil bei `muenzen[7]` wir einen Fehlzugriff generieren würden, da der letzte Eintrag von `muenzen` sich in der Zelle `muenzen[6]` befindet
- ▶ Ab $i = 1$ wird eine Münze verwendet, d.h. wir müssen auf's Array zugreifen. Da die Indizes im Array von 0 bis 6 gehen, müssen wir auf die erste Münze m_1 mittels `muenzen[1-1] = muenzen[0]` zugreifen.

Dynamisches Wechselgeld: Step by Step

- ▶ **Recall:** Eintrag $\text{anz}(i, j)$ enthält die Anzahl Möglichkeiten um den Betrag j mit i Münzen darzustellen, also z.B. im Falle von $i > 0$ mit den Münzen m_1, m_2, \dots, m_i und im Falle von $i = 0$ ohne Münzen

Dynamisches Wechselgeld: Step by Step, Fall 1

1. Falls keine Münzen verwendet werden ($i = 0$):

▶ Falls $j = 0$:

```
anz[i, j] := 1;
```

▶ Falls $j > 0$:

```
anz[i, j] := 0;
```

Erklärung: Falls keine Münze vorhanden ist, dann gibt es zwei Möglichkeiten:

1. Falls $j = 0$, dann gibt es genau eine Möglichkeit um den Betrag 0 ohne Münzen darzustellen
2. Falls $j > 0$, dann gibt es keine Möglichkeit um diesen Betrag ohne Münzen darzustellen

Dynamisches Wechselgeld: Step by Step, Fall 2

2. Sonst falls $m_i > j$ dann:

```
anz[i, j] := anz[i-1, j];
```

Erklärung: Falls der Wert der i -ten Münze grösser als der aktuelle Betrag j ist, dann können wir die Münze nicht verwenden. In diesem Fall ist die Anzahl Möglichkeiten dieselbe wie im Fall, in welchem die Münze m_i weggelassen wird; die Zelle $anz[i-1, j]$ enthält die Anzahl Möglichkeiten die man hat, um den Betrag j mit den Münzen m_1, \dots, m_{i-1} (ohne m_i) darzustellen.

Dynamisches Wechselgeld: Step by Step, Fall 3.1

3. Ansonsten:

3.1 Verwende m_i nicht:

```
a1 := anz[i-1, j];
```

Erklärung: Wir verwenden die Münze m_i nicht, d.h. wir verwenden wie vorhin die Anzahl Möglichkeiten aus Zelle $\text{anz}[i-1, j]$ ab; $\text{anz}[i-1, j]$ enthält nämlich die Anzahl Möglichkeiten die man hat, um den Betrag j mit den Münzen m_1, \dots, m_{i-1} (ohne m_i) darzustellen.

Dynamisches Wechselgeld: Step by Step, Fall 3.2

3. Ansonsten:

3.1 ...

3.2 Verwende m_i mindestens einmal:

```
a2 := anz[i, j - muenzen[i-1]];
```

Erklärung: m_i wird mindestens einmal verwendet, d.h. wir berechnen auf wie viele Arten man den Betrag $(j - m_i)$ mit allen Münzen m_1, \dots, m_i darstellen kann.

Dynamisches Wechselgeld: Step by Step, Ergebnis aus Fall 3.1 und 3.2

3. Ansonsten:

3.1 ...

3.2 ...

- ▶ Speichere die Summe ab:

```
anz[i, j] := a1 + a2;
```

Erklärung: Nachdem wir die Anzahl Möglichkeiten ausgerechnet haben, die vorhanden sind um den Betrag j ohne der Münze m_i darzustellen (Fall 3.1) und die Anzahl Möglichkeiten die vorhanden sind, wenn wir die Münze m_i mind. einmal verwenden (Fall 3.2), so berechnet sich die Anzahl Möglichkeiten um den Betrag j mit den Münzen m_1, \dots, m_i darzustellen aus der Summe der beiden Fälle.

Dynamisches Wechselgeld: Endresultat auslesen

Das Endresultat, also die Anzahl Möglichkeiten, die vorhanden sind um den Betrag n mit allen 7 Münzen m_1, m_2, \dots, m_7 darzustellen befindet sich in der Zelle `anz[7, n]`