

Aufgabe 1: Spezifikation und Modellierung

(35 Punkte)

1. Kurze Fragen.

- a) (4 Punkte) Wir haben formale Modelle zur Beschreibung von asynchronen nebenläufigen Vorgängen kennengelernt. Erklären Sie kurz die Begriffe *formales Modell* und *nebenläufiger Vorgang*.

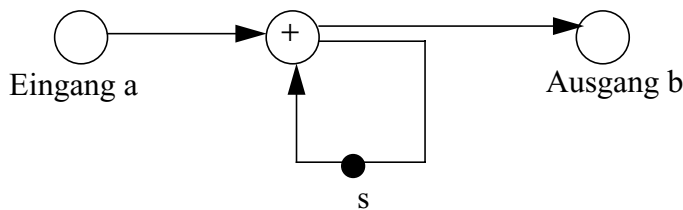
Formales Modell: Ein Modell repräsentiert bestimmte Aspekte des modellierten Systems (Abstraktion). Ein formales Modell repräsentiert diese Aspekte in eindeutiger Form.

Nebenläufiger Vorgang: Ein nebenläufiger Vorgang ist ein Vorgang, der zeitlich unabhängig von anderen Vorgängen im System ablaufen kann

- b) (3 Punkte) Die Zahl der Marken in einem markierten Graphen bleibt immer konstant. Was können Sie über einen solchen Graphen aussagen?

Ein Beispiel für einen solchen Graphen wäre ein gerichteter Ring von Knoten, die durch jeweils eine Kante miteinander verbunden sind. Allgemeiner besitzt ein markierter Graph diese Eigenschaft, falls jeder aktivierbarer Knoten jeweils die gleiche Zahl von Eingangskanten und Ausgangskanten besitzt.

2. Markierte Graphen



Im oben abgebildeten markierten Graphen, stellt der mit + gekennzeichnete Knoten eine Addition der beiden Eingangsdaten des Knotens dar.

- a) (4 Punkte) Am Eingang a wird eine Zahlensequenz eingelesen wobei $a(k)$ den k -ten Wert bezeichnet. Geben Sie die Ausgangssequenz $b(k)$ als Funktion der Eingangswerte an.

$$b(1) = a(1) + s.$$

$$b(k) = a(k) + b(k-1) \quad \text{für } k > 1$$

wobei s der Datenwert der Anfangsmarkierung ist.

- b) (6 Punkte) Die Anfangsmarkierung mit dem Wert s wird durch n Markierungen ersetzt. Geben Sie die Rekursionsformel für die Ausgangssequenz $b(k)$ an.

für $n=0$ ist die Ausgangssequenz leer.

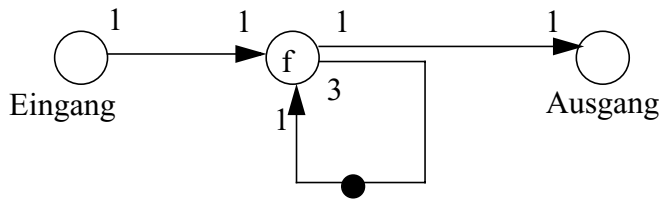
für $n > 0$:

$$b(k) = a(k) + s_k \quad \text{für } k \leq n$$

$$b(k) = a(k) + b(k-n) \quad \text{für } k > n$$

wobei $s_1 \dots s_n$ die Datenwerte der Anfangsmarkierungen sind.

3. Synchrone Datenflussgraphen



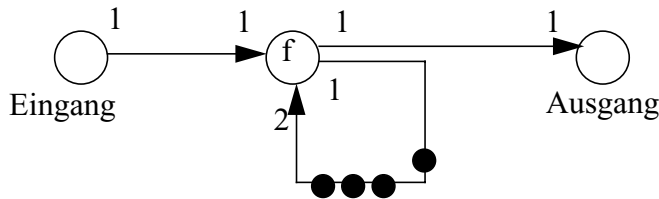
Der oben abgebildete synchrone Datenflussgraph ist aus dem bereits verwendeten markierten Graphen abgeleitet. Der Knoten führt die Funktion f auf den Eingangsdaten aus.

- a) (4 Punkte) Welche Probleme könnten entstehen, wenn dieser Graph in ein Softwareprogramm umgesetzt wird?

Der Speicherbedarf steigt bei einer fortgesetzten Ausführung unbegrenzt an.

- b) (2 Punkte) Ändern sie die Raten und die Anfangsmarkierung **in der Schleife** so, dass der Ausgangsknoten höchstens 3 Mal feuern kann. Eine Rate mit dem Wert 0 ist nicht zulässig.

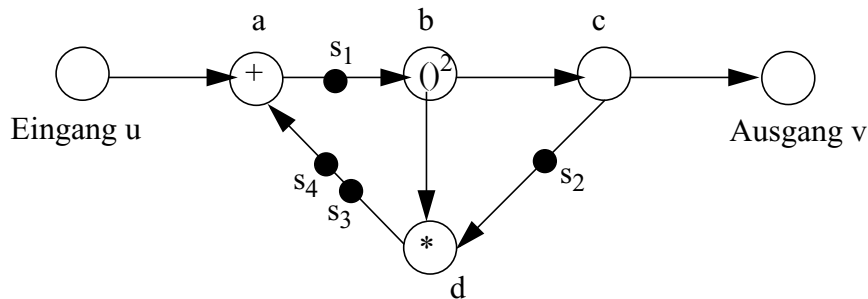
(zum Beispiel)



- c) (4 Punkte) Setzen sie den Graphen aus Teilaufgabe (b) in ein Programm in der Sprache C um. Dabei können sie die Prozeduraufrufe `readInput()`, `writeOutput(...)`, `readQueue()` und `writeQueue(...)` verwenden, die ein Datum vom Eingang zurückliefern, ein Datum auf den Ausgang schreiben, ein Datum von einer Warteschlange lesen beziehungsweise ein Datum in eine Warteschlange schreiben. Die Lese-Prozeduren warten jeweils, bis die entsprechenden Daten vorliegen. Am Anfang ist die Warteschlange leer.

```
void sdf()
{
    writeQueue(s1); writeQueue(s2); writeQueue(s3); writeQueue(s4);
    while(true)
    {
        a := readQueue(); b := readQueue(); c := readInput();
        f(*out1, *out2, a, b, c);
        writeOutput(out1);
        writeQueue(out2);
    }
}
```

4. In der nächsten Teilaufgabe geht es um den folgenden markierten Graphen:



s_1 bis s_4 sind die Datenwerte der Anfangsmarkierungen. Die Knoten a , b , c und d führen die Funktionen *Addition*, *Quadrieren*, *Weiterleiten* und *Multiplizieren* auf den jeweiligen Eingangsdaten aus. Im Fall mehrerer Ausgangskanten haben die bei einer Feuerung generierten Daten die gleichen Werte.

a) (8 Punkte) Am Eingang u wird eine Zahlensequenz $u(k)$ eingelesen. Geben Sie die Ausgangssequenz $v(k)$ als Funktion der Eingangswerte an.

$$v(1) = (s_1)^2$$

$$v(2) = (u(1) + s_4)^2$$

$$v(3) = (u(2) + s_3)^2$$

$$v(4) = (u(3) + y(1) * s_2)^2$$

$$v(k) = (u(k-1) + y(k-3) * y(k-4))^2 \quad \text{für } k > 4$$