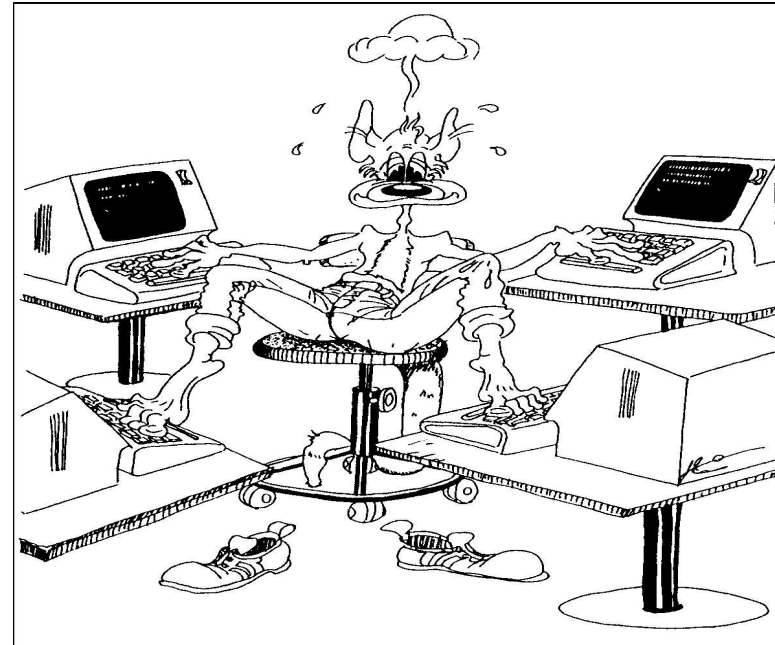


Nebenläufigkeit



Vortrag
Thomas Schirrmann

Seminar Systementwurf
Dozentin Daniela Weinberg

Gliederung

1. Einführung
2. Modellierung
 - 2.1. POMSET
 - 2.2. Transitionssystem
 - 2.3. Petrinetz
 - 2.4. abstraktes nebenläufiges Programm
3. Fairness
 - 3.1. Unvoreingenommenheit
 - 3.2. Gerechtigkeit
 - 3.3. Strong Fairness
4. Kommunikation
 - 4.1. gemeinsame Variablen
 - 4.2. Message Passing (Nachrichtenaustausch)
5. typische Fehler
6. Zusammenfassung
7. Quellen

Einführung

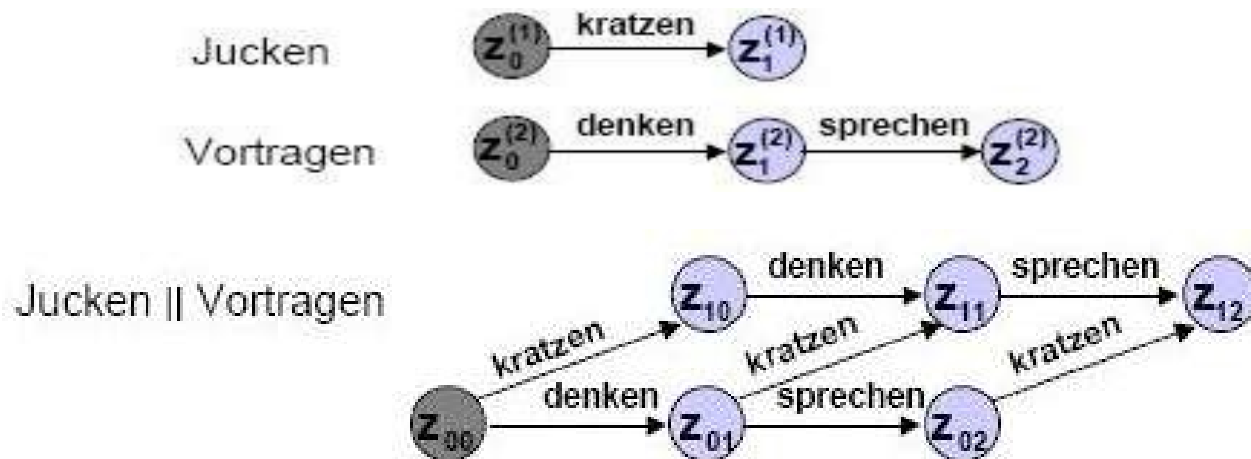
- Programmablauf
 - sequentiell
 - nebenläufig
 - parallel
- Konzepte Nebenläufigkeit
 - Endlosbetrieb
 - Kommunikation
 - Konkurrenz und Kooperation

Modellierung

- Schreibweisen:

- nebenläufig: $P_1 \parallel P_2 \parallel \dots \parallel P_n$

- konkateniert: $P_1 P_2 \dots P_n$ oder $P_1; P_2; \dots; P_n$



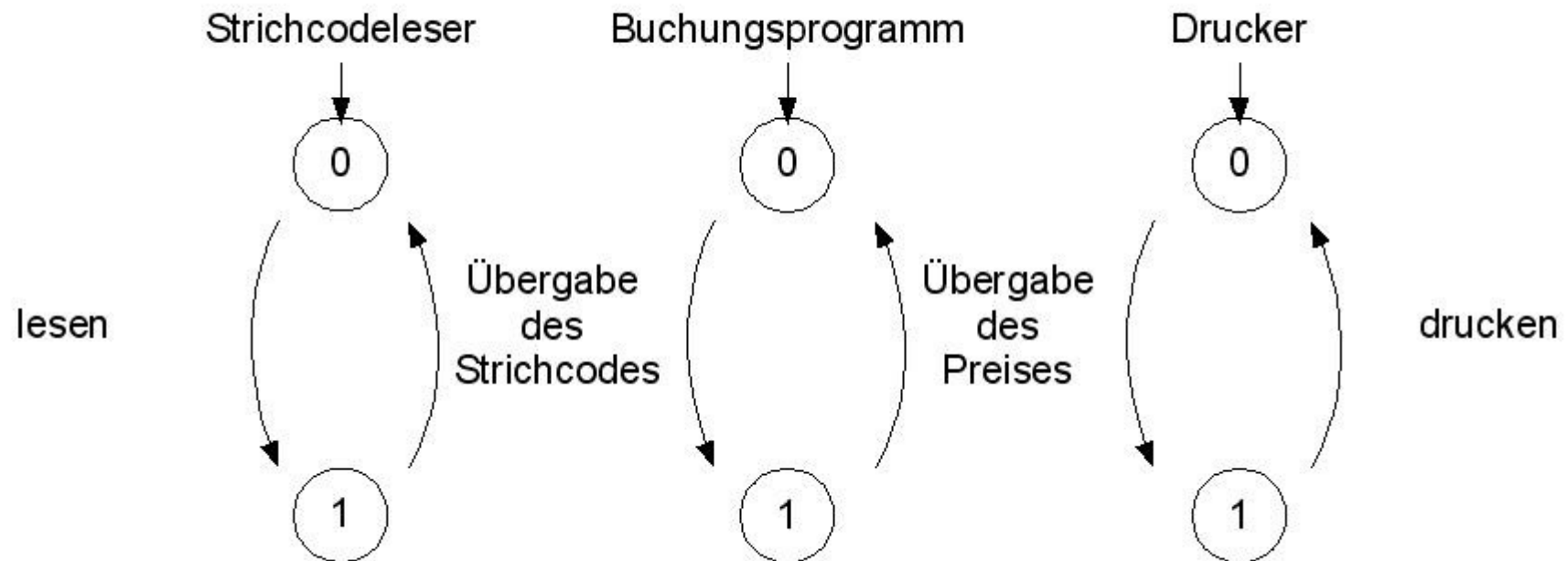
- Granulierung wichtig (Definition eines atomaren Befehls)

Modellierung – POMSET

- viele Verzahnungsmöglichkeiten bei Nebenläufigkeit
- POMSET (Partially Ordered Multiset)
- Einschränkung der Reihenfolge von Prozessen durch die Ordnung
- Schreibweise: $N (A_1, A_2, \dots, A_n)$ wobei P_i Prozess ($i=1\dots n$)

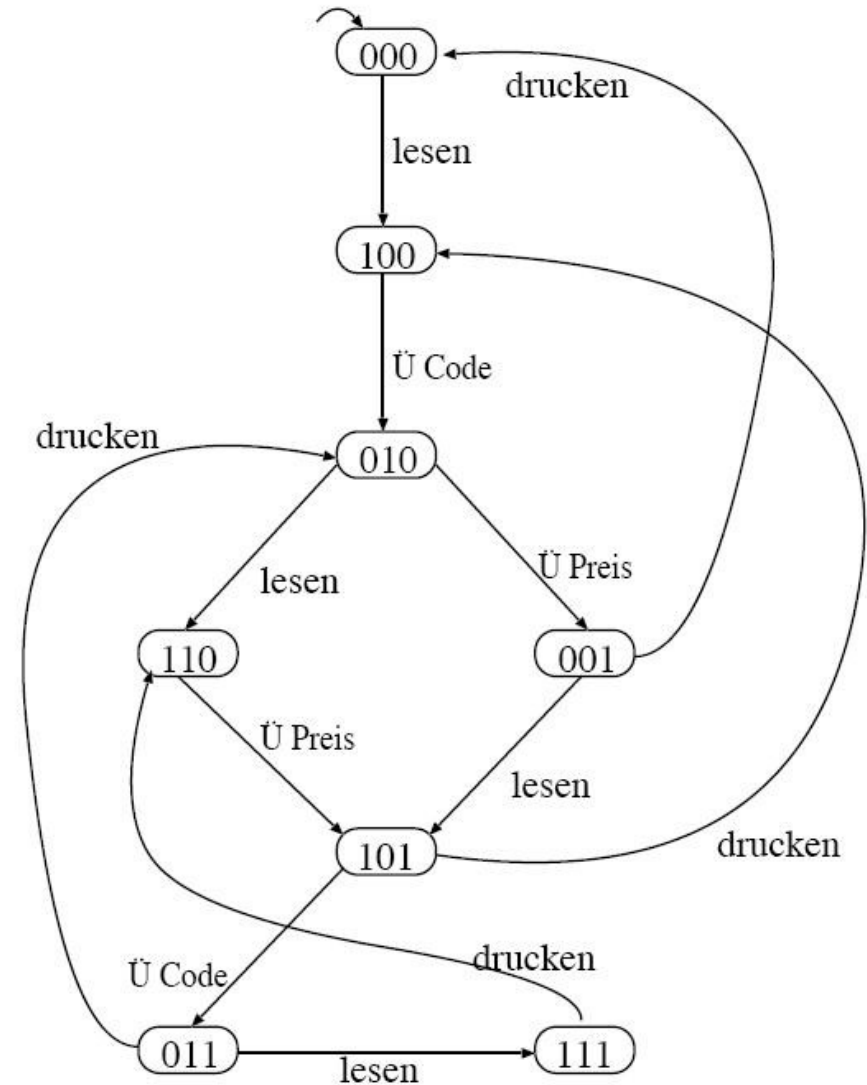
Modellierung – Transitionssystem

- Beispiel Kassensystem
- modelliert als 3 unabhängige Zustandsautomaten



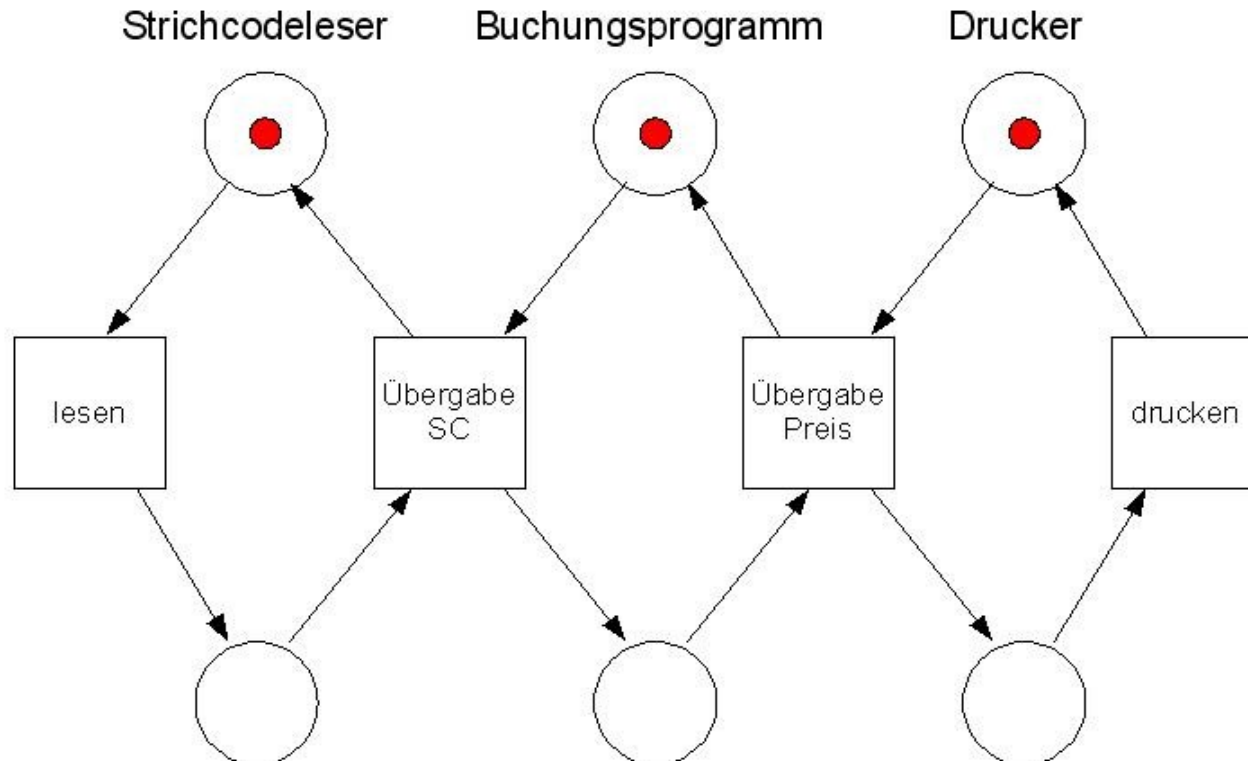
Modellierung – Transitionssystem (Fort.)

- zusammenfassen der Teilprozesse
- keine echte nebenläufigkeit modelliert
- nichtdeterministische Entscheidungen



Modellierung – Petrinetz (Fort.)

- Überführung in Petrinetz
- Synchronisationsgraph (marked graph)



Modellierung – Abstraktes nebenläufiges Programm

- Tripel $(M, \Phi_{\text{START}}, \Phi)$
- M : Multiprozessstruktur mit Beachtung der Zeit
- Φ_{START} : atomare Annahmen zum Startzeitpunkt
- Φ : Fairnesseinschränkungen

Fairness

- mehrere Teilsysteme
- Koordination notwendig
- k nebenläufige Prozesse – faire Ausführung?

Fairness – Unvoreingenommenheit

- Englisch: Impartiality
- Definition:
Eine unendliche Folge ist unvoreingenommen, wenn jeder Teilprozess k während der Berechnung unendlich oft ausgeführt wird.
- LTL: $\bigwedge_{i=1}^k (F \text{ executed}_i)$

Fairness – Gerechtigkeit

- Englisch: Justice oder Weak Fairness

- Definition:

Ein unendlicher Berechnungsprozess ist gerecht, wenn jeder Teilprozess k , der aktiviert wird, danach unendlich oft ausgeführt wird.

- LTL: $\bigwedge_{i=1}^k (G \text{ enabled}_i \rightarrow \overset{\infty}{F} \text{ executed}_i)$

Fairness – Strong Fairness

- meist nur Fairness
- Definition:
Ein unendlicher Berechnungsprozess ist fair, wenn jeder Teilprozess k , der (unendlich oft) aktiviert wird, dann unendlich oft ausgeführt wird.
- LTL: $\bigwedge_{i=1}^k (\overset{\infty}{F} \text{ enabled}_i \rightarrow \overset{\infty}{F} \text{ executed}_i)$

Kommunikation

- Datenaustausch zwischen Prozessen über Prozessgrenzen hinweg
- grundlegende Varianten:
 - Verwendung gemeinsamer Variablen
 - Message Passing (Nachrichtenaustausch)

Kommunikation – gemeinsame Variablen

- schnell
- über Semaphoren erreichbar
- ohne zusätzlichen Prozess machbar
- nicht verteilbar (gemeinsamer Zugriff auf Speicher)

Kommunikation – Nachrichtenaustausch

- synchron/asynchron
- langsamer
- Überwachung mittels Kommunikationsprozess
- problemlos verteilbar
- räumliche Trennung kein Problem

Kommunikation – Nachrichtenaustausch (Fort.)

- Arten des Nachrichtenaustauschs
 - Kanal/Mailbox
 - globaler Versand (Broadcast)
 - Fernaufruf (RPC)

typische Fehler

- Deadlock
- Livelock
- Race-Condition (darunter Lost-Update-Problem)

Zusammenfassung

- Vorteile
 - höhere Geschwindigkeiten dank Nebenläufigkeit
 - Realisierung *Verteilter Systeme*
- Nachteile
 - komplexere Modellierung notwendig
 - fehleranfälliger als sequentielle Programme

Quellen

- Emerson, E. A.: Temporal And Modal Logic (2005).
- Plotkin, G.; Pratt, V.: Teams Can See Pomsets (1990).
- Barland; Greiner; Vardi: Concurrent Processes: Basic Issues. URL: <http://cnx.org/content/m12312/latest/> [25.11.2006].
- Baier, C.: Model Checking. Verifikation paralleler Systeme (2006).