Grundlagen der Systemmodellierung I und II

Grundwissen

System

- statisch: regelhafte strukturierte Zusammenstellung von Objekten
- dynamisch: konkretes Gebilde mit beobachtbarem Verhalten durch Zusammenwirken der Systemteile
  - materiell-energetisches System + Interpretation = informationelles System

Information

- ist potentielles Wissen
- Form → Information: Interpratation (benötigt Interpretationsvereinbarung)
- Information → Form: Codierung

Interpretation

- wertunmittelbar/ wertdiskret: jedem Zeitpunkt eine Bedeutung zuordnen
- wertverlauf/ wertkontinuierlich: dem Verlauf eine Bedeutung zuordnen
- analog zeitdiskret/ zeitkontinuierlich

Formale Sprache

- Typenrepertoire von Formbausteinen (Symbolen, Zeichen)
- Menge K der kombinatorisch möglichen Formstrukturen
- Regelwerk (formales System), das K die Menge F der gewünschten Formstrukturen auf Basis der Form zuordnet

Grammatik

- Terminalrepertoire $rep\ T$
- Superzeichenrepertoire $rep\ S$
- Axiom $\in S$
- Regeln
- Sprachumfang $L = \{w | w \in T^* \wedge w \text{ aus Regeln herleitbar}\}$

Chomsky-Hierarchie

- **Typ 0:** wenigstens eine Regel, die ein Superzeichen durch einen leeren Abschnitt ersetzt
- **Typ 1:** längenmonoton, mind. eine Regel muß kontextabhängig sein (*kontextsensitiv*)
- **Typ 2:** längenmonoton, kontextfrei, nicht regulär
- **Typ 3:** alle Regeln der Form $s \rightarrow t[s]$ mit $s \in rep\ S$ und $t \in rep\ T$
- Komplexität des Sprachumfangs ist die geringstmögliche Komplexität der Grammatik in Chomsky-Hierarchie

Kalkül

- regelhaftes Durchführen von Beweisen ausgehend von als wahr akzeptierten Aussagen (den Axiomen)
- Regeln, die die Ableitung von Aussagen aus gegebenen Aussagen erlauben
- grenzt beweisbare Aussagen, d.h. auf Basis der Form aus Axiomen ableitbar, ab
- Vollständigkeit (alle wahren Aussagen beweisbar), Widerspruchsfreiheit (keine falschen Aussagen), Entscheidbarkeit (gehört Aussage zum Kalkül)

Modell

- Abstraktion eines Systemgebildes, erfasst nur endliche Anzahl ausgewählter Sachverhalte
- System: reales Gebilde, konkrete Ausprägung des Modells
- Systembeschreibung: reales Gebilde, welches Systemmodell identifiziert
- Analysemodell: Modell zu bereits existierendem Objekt (erlaubt Vorhersagen)
- Konstruktionsmodell: Modell zur Vorgabe der Fertigung des Systems

Mengen, Relationen

- Menge: ungeordnete Zusammenstellung von Objekten, willkürlich oder über eine Typ (Klasse)
- Typ: abstraktes Objekt, das nur kennzeichnende Merkmale der Klassenmitglieder aufweist
- Relation: Teilmenge des kartesischen Produktes
- Aussageform: enthält Variablen von deren Belegung Wahrheitswert abhängt
- Struktur: Tupel aus Mengen M_i und Relationen R_i: $S = (M_1, ..., M_n, R_1, ..., R_m)$

	Äquivalenzrelation	Verträglichkeitsrelation	Halbordnung	Vollordnung(*)	Quasiordnung	Schichtung
reflexiv	X	X	O	O	O	
antireflexiv			O	O	O	X
symmetrisch	X	X				
antisymmetrisch			X	X	O	X
transitiv	X	O	X	X	X	O

X = muss sein
O = kann sein
(*) zusätzliche Bedingung: es gibt keine Paare, die nicht geordnet sind

Systemmodellierung

- keine Betrachtung des inneren Aufbaus des Systems
- Beobachung der Eingänge X und Ausgänge Y
- **gerichtetes System:** wenn durch Systemkonstruktion bereits festgelegt ist, was Ein- und Ausgänge sind
- **determiniertes System:** wenn sich aus bestimmtem Verlauf der Eingaben $X(t)$ eindeutig der Verlauf der Ausgaben $Y(t)$ ergibt
- **kausales System:** die Ausgabe $Y(t_1)$ hängt vom Verlauf der Eingaben $X(t)$ bis zum Zeitpunkt t_1 ab
- **zustandsbasiertes Verhaltensmodell:** die vergangene Einflußnahme der Umgebung schlägt sich im Zustand des Systems wieder (vgl. Codeschloß)
- **sequentielles Verhaltensmodell:** zeit- und wertdiskret, nur Betrachtung des Schnittstellenverhaltens

Automaten

- **Mealy-Automat:** zu jeder Eingabe gibt es eine Ausgabe
  - $Z(n+1) = \delta(X(n), Z(n))$
  - $Y(n) = \omega(X(n), Z(n))$
- **Moore-Automat:** zu jedem Zustand gibt es eine Ausgabe
  - $Z(n+1) = \delta(X(n), Z(n))$
  - $Y(n) = \mu(Z(n))$
- **unendlicher Automat:** unendliche Anzahl potentieller Zustände (vgl. Stackverwalter)
- **Eingabebeschränkung:** in bestimmten Situationen können bestimmte Eingaben nicht vorkommen
- **unspezifizierte Ausgabe:** zu bestimmten Eingaben ist keine Ausgabe erforderlich

Zustandsminimierung

- Verschmelzbarkeit = experimentelle Nichtunterscheidbarkeit
- Zustände sind verschmelzbar, wenn sie für alle Eingaben dieselbe Ausgabe haben und in denselben Folgezustand wechseln
- Zustände sind bedingt verschmelzbar, wenn sie sich in den Folgezuständen unterscheiden, diese aber verschmelzbar sein könnten

	Ausgaben	Folgezustände	verschmelzbar
1	$\forall x \in rep\ X : \omega(x, Z_1) = \omega(x, Z_2)$	$\forall x \in rep\ X : \delta(x, Z_1) = \delta(x, Z_2)$	ja
2	$\exists x \in rep\ X : \omega(x, Z_1) \neq \omega(x, Z_2)$	egal	nein
3	$\forall x \in rep\ X : \omega(x, Z_1) = \omega(x, Z_2)$	$\exists x \in rep\ X : \delta(x, Z_1) \neq \delta(x, Z_2)$	bedingt

Petri-Netze

- Petri-Netz (PN): makierter, bipartierter, gerichteter Graph
- Schaltbereitschaft: wenn Einganstellen markiert und Ausgangstellen, die nicht Eingangsstellen sind, unmarkiert sind
- Schalten: Eingangsstellen, die nicht Ausgangsstellen sind, werden unmarkiert und Ausgangsstellen markiert
- Konflikt: mind. zwei schaltbereite Transitionen, bei denen das Schalten der einen die Schaltbereitschaft der anderen aufhebt (Menge von EFGs)
- Nebenläufigkeit: mehrere Schaltbereite Transitionen ohne Konflikt
- Nebenläufigkeitsgrad: maximal nebenläufig schaltbereite Transitionen in der Klasse der Anfangsmarkierung
- Markierungsklasse: alle Markierungen, die direkt oder indirekt von Markierung M aus ereichbar sind (inkl. M selbst)
- Sicheres PN: Wenn alle Eingabestellen einer Transition markiert sind, dann ist die Transition schaltbereit (d.h. keine Ausgangsstellen belegt)
- Tote Transition: in keiner Markierung der Markierungsklasse schaltbereit
- Tote Markierung: eine Markierung bei der keine Transition schaltbereit ist (Endzustand, Deadlock)
- Lebendige Transition: in der Markierungsklasse nur Markierungen, von denen aus die Transition schaltbereit gemacht werden kann
- Äquivalenz von PNs: $PN_1 = PN_2 \Leftrightarrow EFG(PN_1) = EFG(PN_2)$

Petri-Netze mit Kantengewichten und Stellenkapazitäten

- Stellenkapazität: jeder Stelle wird eine Maximalzahl ($\in \mathbb{N}$) der dort gleichzeitig vorliegenden Marken zugeordnet
- Kantengewicht: jeder Kante wird Anzahl der Marken zugeordnet, die beim Schalten über die Kante fließen (müssen!)
- Schaltbereitschaft
  - genügend Marken (gemessen am Kantengewicht) auf jeder Eingangsstelle
  - ausreichend Platz zum Ablegen der Marken auf den Ausgangsstellen
- komplentäre Stelle
  - Kapazität := Kapazität der Originalstelle
  - Anfangsmarkierung := Kapazität der Originalstelle - Anzahl der in Anfangsmarkierung liegenden Marken
  - Verbindung mit Transitionen := Kantengewicht des Originals aber andere Richtung
- Unsicher markierte PN können durch Einführung komplementärer Stellen in äquivalente, sicher markierte PN überführte PN überführt werden!

Operationell erweiterte Petri-Netze

- Operationsakteur (Schrittrepertoire), Steuerakteur (Schrittfolgengenerator)
- Kommunikation
  - flüchtige Ausgaben, anstoßender Eingaben ($Y(n) \leftrightarrow X(n)$, z.B. Gegentaktbetrieb)
  - nicht flüchtige Ausgaben, anstoßende Eingaben ($Y(n) \rightarrow X(n)$)
  - flüchtige Ausgaben, abgetatstete Eingaben
  - nicht flüchtige Ausgaben, abgetatstete Eingaben (z.B. mit zusätzlicher Taktung)

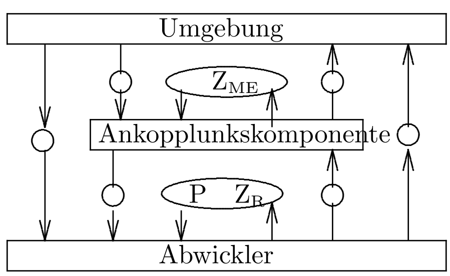
Rollen- und Abwicklersysteme

- Programmzweck
  - ergebnisorientiert
  - prozessorientiert
- Programmformulierung
  - prozedural/ imperativ (nebenläufig/ sequentiell)
  - funktional/ applikativ
  - deklarativ (Wissenbasis + Regeln)

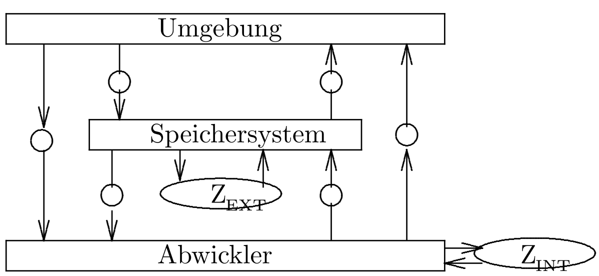
Prozeduraler Abwickler

- `HOLEN/ FETCH` - nächsten Befehl holen
- `SEPARIEREN/ DECODE` - Änderungen des Befehlszählers und operationeller Variablen ermitteln
- `OPERIEREN/ EXECUTE` - neuen Befehlszähler bestimmen, op. Variablen ändern
- universeller Befehlsatz:
  - vollständige Abfragbarkeit: $\forall w \in W($ if$(j^* = w)$ jmp i
  - vollständige Setzbarkeit: $\forall w \in W(i^* = w)$
  - entspicht vollständiger Algebra $(W, f_1, ..., f_n)$: mittels f_i lässt sich jede Funktion $W^n \rightarrow W$ formulieren

Materiell energetischer Abwickler



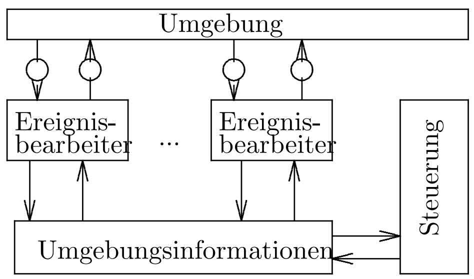
Informationeller Abwickler

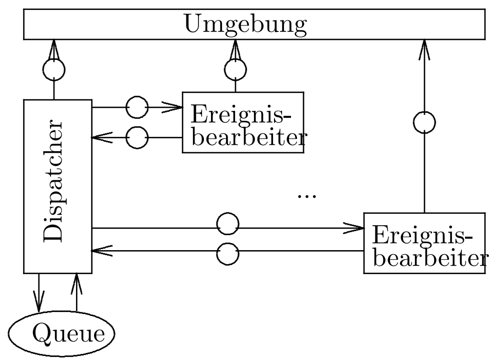


Prozeduraler multiplexfähiger Abwickler

- Frequenzmultiplex: Kanal i wird auf bestimmte Trägerfrequen F_i abgebildet, Kanal M bietet ausreichende Anzahl Frequanzen
- Zeitmultiplex: Kanal M kann nur eine Nachricht Transportieren, Nachrichten müssen zwangsseqzentialisiert werden, bei Multiplex-Abwicklern verwendet
- Umschalten (Rollenwechsel) des Abwicklers nach Dringlichkeit, gezielte Auswahl der Ereignisbearbeiter und Unterbrechung (synchroner Interrupt)
- Abwicklerumschaltung:
  - `RETTEN` - Kontext des Abwicklers zu späteren Wiederaufnahme sichern
  - `AUSWÄHLEN` - des nächsten abzuwickenden Programms
  - `BELEGEN` - des Abwicklerspeichern mit passendem Kontext

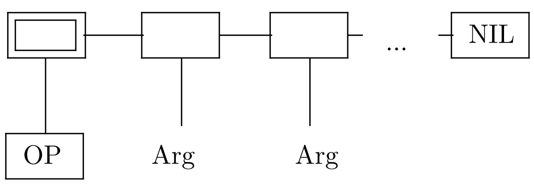
(1) Umgebungsbedingte Nebenlaeufigkeit (2) Systembedingete Nebenlaeufigkeit





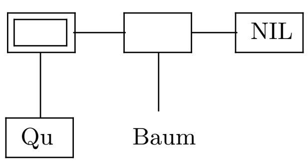
Funktionale Programmierung/ Baumformat

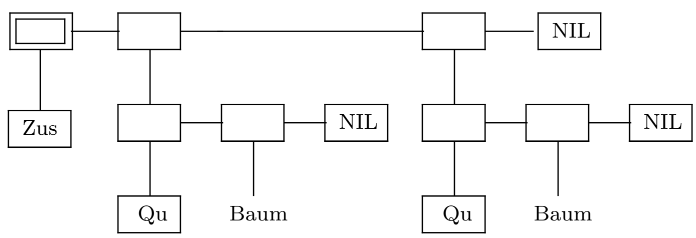
Prinzipieller Aufbau:



Baumfunktionen:

(1) Quote (Baum als Wert) (2) Zusammenfuegen





(3) Li (linker Teilbaum)
(4) Re (rechter Teilbaum)
(5) Eval (Auswertung des Baumes)