Petri-Netz-Synthese

Synthese von Petri-Netz-Modellen

(Regionentheorie 1)

Rico Bergmann

03.12.2007

Einführung

- Interface-Synthese für embedded systems, z.B. in
  - Fahrzeugtechnik
  - Haushaltselektronik
  - Unterhaltungselektronik

Einführung

- Modellierung
  - In einer „einfachen Sprache" modellieren
  - Eigenschaften prüfen
  - Bei PNen: kleinere Netze finden

- Algorithmen
  - Sind Deadlocks vorhanden?
  - ...

Das Problem (informell)

Finde zu einem gegebenem System (in Form eines Transitionssystems) ein Petri-Netz mit demselben Verhalten.

Begriffe

- Sei P = (S, T, F, M_0) ein Petri-Netz
- Wir betrachten nur elementare beschriftete System-Netze
  - Schleifenfrei: $\forall t \in T : \bullet t \cap t \bullet = \emptyset$
  - Einfach: $(\bullet x = \bullet y \wedge x \bullet = y \bullet) \Rightarrow x = y$
  - Ohne isolierte Transitionen
- $M_0 \subseteq S$

Begriffe

- Elementare Transitionssysteme (eTS)
  - $T = (K, L, G, k_0), G \subseteq (K \times L \times K), k_0 \in K$
  - Endlicher gerichteter Graph
  - Kanten-beschriftet (L)
  - Initialisiert (k_0)
  - Zusätzlich: nur erreichbare Knoten
  - Keine „Self-Loops"
  - Keine Mehrfachkanten

Begriffe

- Zur Erinnerung:

Finde zu einem gegebenem System (in Form eines Transitionssystems) ein Petri-Netz **mit demselben Verhalten**.

- Graphisomorphismus
  - T und T�' seien eTS
  - Bijektive Abb. f:K→K' heißt Isomorphismus
    - $f(k_0) = k_0'$
    - $(h,l,k) \in G \Leftrightarrow (f(h),l,f(k)) \in G'$
  - Notation: T $\cong$ T'

Abstecher: Graphisomorphie

- Liegt das Graphismorphie-Problem in P oder NP?
- Vermutlich ist es nur schwer lösbar
- Aber: wir haben es einfach
  - Spezialfall: wir suchen einen speziellen Isomorphismus

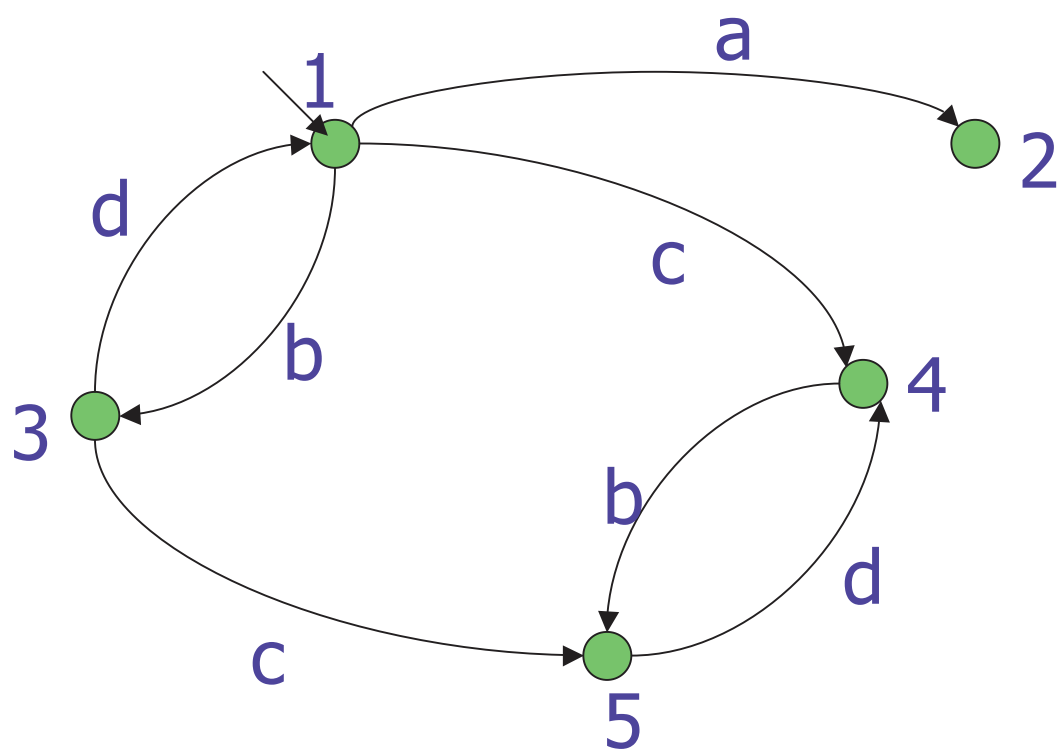
Das Problem (formal)

- Sei $T=(K, L, G, k_0)$ ein eTS
  - Dann ist ein ePN $P=(S, T, F, M_0)$ gesucht
  - Mit $ER(P) \cong T$
    - ER(P) ... Erreichbarkeitsgraph von P
- Das synthetisierte PN soll klein sein
- Falls es P gibt, bez. wir T als abstrakten Zustandsgraphen

Regionen

- Sei T eTS
  - R, Teilmenge der Knoten von T, heißt Region gdw.
    - Für Kanten (h,l,k), (h`,l,k`) mit gleichem Label l gilt:
      - $h \in R$ und $k \notin R$, dann $h` \in R$ und $k` \notin R$ (exiting event)
      - $h \notin R$ und $k \in R$, dann $h` \notin R$ und $k` \in R$ (entering event)
- $\emptyset$ und K heißen triviale Regionen
- Reg(T) .. Menge aller Regionen von T

Regionen Beispiel



Petri-Netz-Synthese

- Idee: Regionen → Plätze
- Ein m-generiertes ePN ist
  - $sy(T,m) := (m,L,F,c_0)$, m.. Menge von Regionen
  - Wobei für $R \in m$, $l \in L$ gilt
    - $(R,l) \in F$ gdw. $R \in {}^\circ l$
    - $(l,R) \in F$ gdw. $R \in l^\circ$
    - $R \in c_0$ gdw. $k_0 \in R$

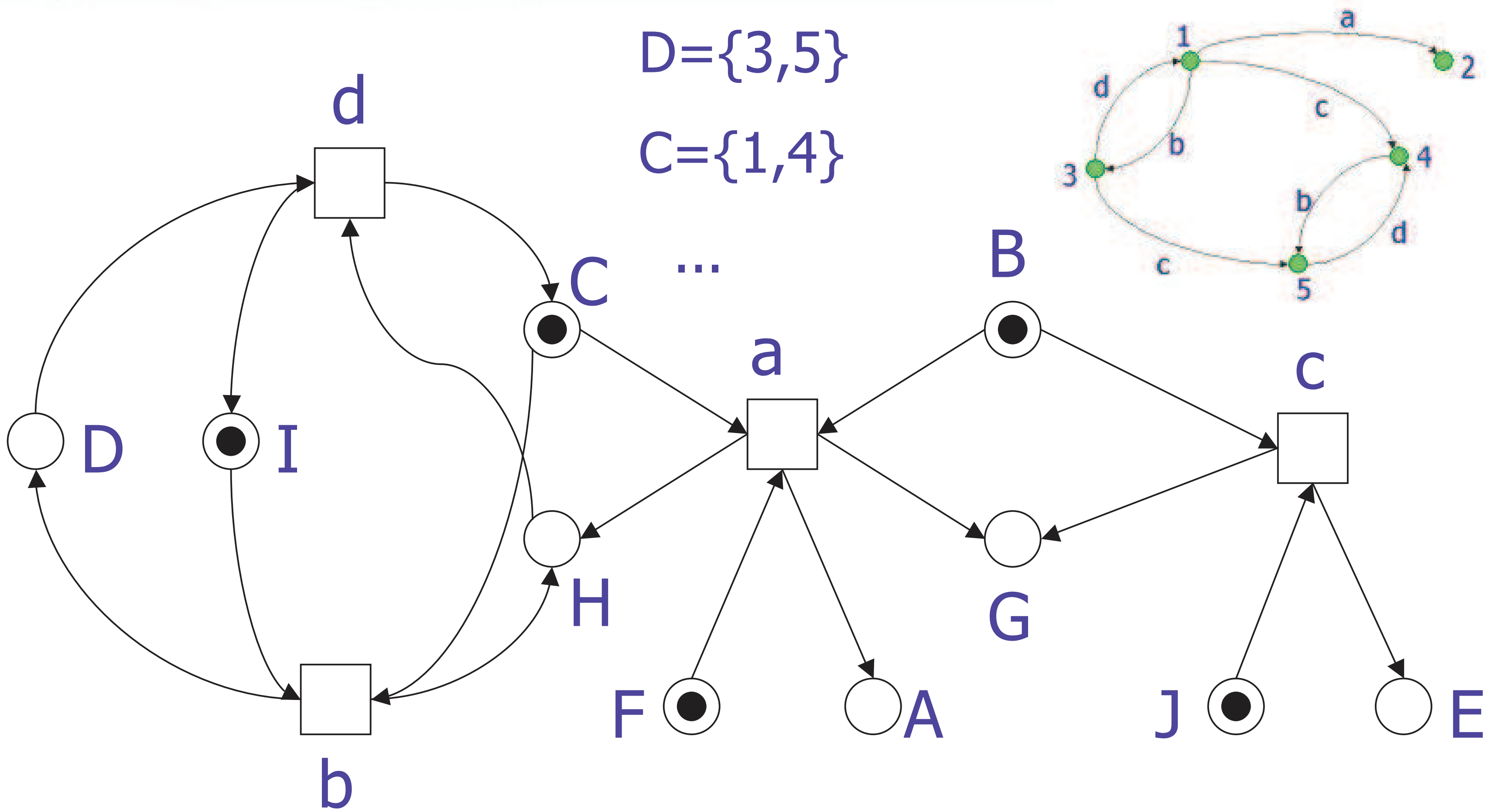
${}^\circ l$ ..Menge der Regionen mit l als exiting event (l° analog)

PN-Synthese Beispiel

D={3,5}

C={1,4}

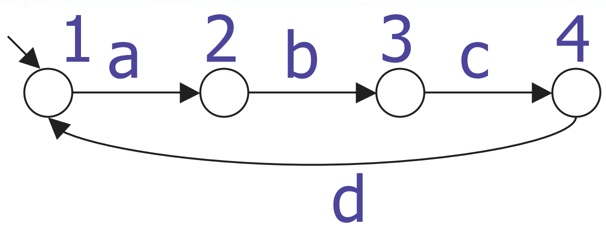
...

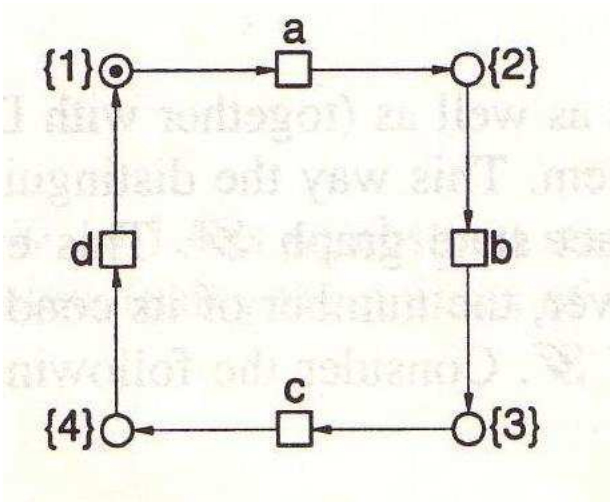


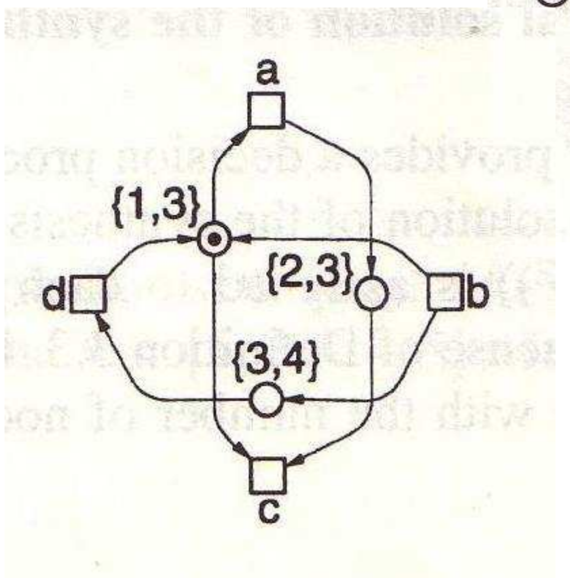
PN-Synthese Isomorphismus

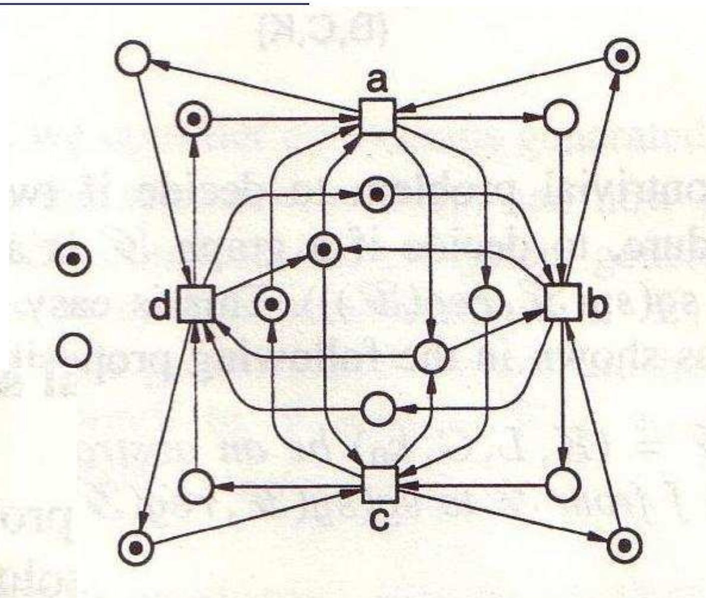
- Falls T abstrakter Zustandsgraph, dann
  - Existiert genau ein Isomorphismus f
  - Von T nach ER[sy(T, reg(T))]
  - Mit $f(k)=\{R \in reg(T) \mid k \in R\}$
- Beweisidee:
  - Zeige für jeden bel. Isomorphismus g gilt: g=f
    - Initialzustände per Definition
    - Induktiv über erreichbare Zustände:
      - $(h,l,k) \in G,\ f(h)=g(h) \Rightarrow f(k) = g(k)$

PN-Synthese Beispiel T_4









Reduced Petri Nets

- Problem:
  - ePN können sehr groß werden (exponentiell)
- Lösung:
  - Verkleinerte ePN:
    - Jedes Event tritt auf
    - p1, p2 Plätze des ePN → es existiert eine Markierung M mit $p1 \in M$, $p2 \notin M$ oder $p1 \notin M$, $p2 \in M$

Reduced Petri Nets

- Idee zur Verkleinerung:
  - Weglassen überflüssiger Plätze
  - $m \subseteq reg(T)$ ist zulässig, gdw. sy(T, reg(T)) und sy(T, m) „isomorph"
  - $R \in reg(T)$ ist redundant, falls $m \setminus \{R\}$ zulässig

Redundante Regionen

- Charakterisierung redundanter Regionen:
    - R sei Region, dann bezeichne R` := K \ R
  - R und R` sind Regionen → R ist redundant
  - R, R_i Regionen, $R = R_1 \cap R_2$, $R` = R_3 \cap R_4$ → R ist redundant
  - R, R_i Regionen, $R = R_1 \cup R_2$, $R` = R_3 \cup R_4$ → R ist redundant
  - R, R_i Regionen, $R = R_1 \cap R_2$, für alle $(h,l,k) \in R \times L \times R`$ gilt $k \notin R_1$ und $k \notin R_2$ → R ist redundant

Zusammenfassung

- PN-Synthese
  - Geg. eTS T
  - Konstruiere ein ePN P zu T
  - Isomorphismus zwischen P und T
  - Entferne redundante Regionen
  - Ausgabe: „kleines" ePN

Ausblick

- PN-Synthese
  - Geg. e~~T~~S T
  - Konstruiere ein e~~P~~N P zu T
  - Isom~~o~~rphismus zwischen P und T
  - Entferne redundante Regionen
  - Ausgabe: „kle~~in~~es“ ePN