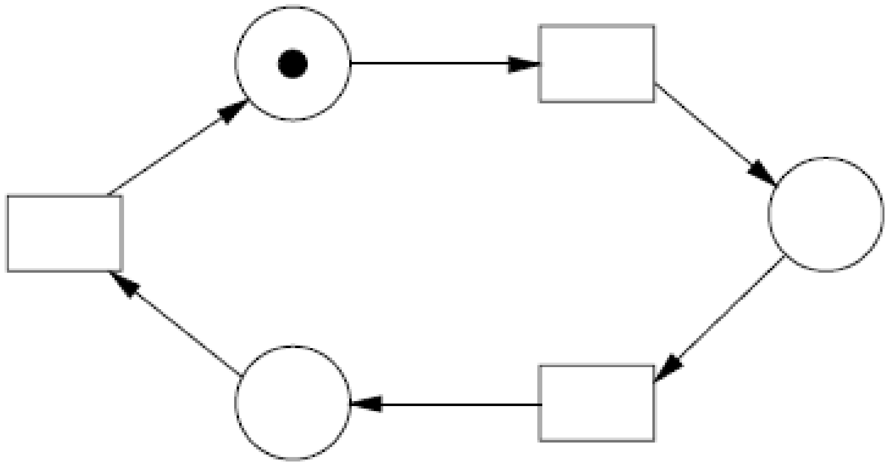
Model-Checking mit Symmetrien



Gliederung

- Grundlegendes
- Gruppen und Symmetrien
- Berechnung von Symmetrien in PN
- Literatur

Grundlegendes (1)

- Explosion des Zustandsraumes
  (*state explosion*)
- Besonders bei nebenläufigen Systemen (*concurrent systems*)
  z.B. Kommunikationsprotokolle:
  Netz besteht aus identischen Prozessen

Grundlegendes (2)

Symmetrie impliziert die Existenz einer nichttrivialen Permutationsgruppe, die den Transitionsgraphen* erhält.

---

*) Systembeschreibung durch Zustände und Transitionen

Grundlegendes (3)

Ziel, Idee ...

Erstellen eines reduzierten Modells des Systems (durch Bildung von Äquivalenzklassen)

➲ Vereinfachte Verifizierung

Gruppen und Symmetrie (1)

➲ Gruppe

- Menge G mit binärer, assoziativer Relation auf der Menge

- Existenz eines inversen (a^{-1})bzw. neutralen Elements (*e*)

- Erzeugendensystem

Gruppen und Symmetrie (2)

- Automorphismus

- bijektiver Homomorphismus auf sich selbst

- zwei Gruppen $(G, \cdot)$ und $(H, \cdot)$, Funktion f: $G \rightarrow H$

-f ist Gruppenhomomorphismus, wenn für alle x, y von G gilt:

  - $f(x \cdot y) = f(x) \cdot f(y)$.

(strukturerhaltende Abbildung)

Gruppen und Symmetrie (3)

- Permutation σ auf eine endliche Menge A ist eine bijektive Selbstabbildung $\sigma : A \to A$

- Menge aller Permutationen auf A wird mit *Sym* A bezeichnet

- *Sym* A ist Gr. mit funktionaler Komposition und Identität als neutralem Element

Gruppen und Symmetrie (4)

- Schreibweisen

- Zyklus

$i_1 \to i_2,\ i_2 \to i_3,\ \ldots,\ i_{k-1} \to i_k,\ i_k \to i_1$

- Transposition (2-Zyklen)

Permutation, die genau zwei Elemente miteinander vertauscht.

- Jede Permutation ist Komposition von Transpositionen.

Symmetrien in Petrinetzen (1)

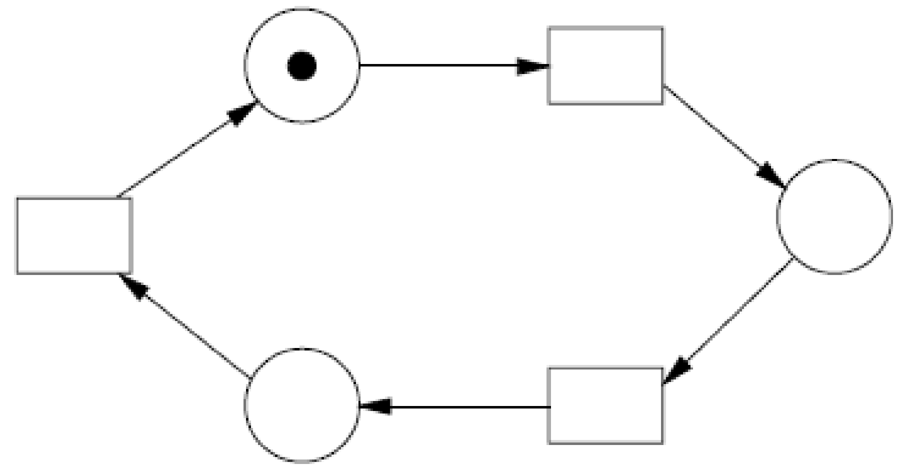
- Symmetrien sind Bijektionen auf der Menge $P \cup T$ von Knoten eines Netzes, die
  - Knotentyp,
  - Pfeilrelation (Transition) und
  - Netzinschriften

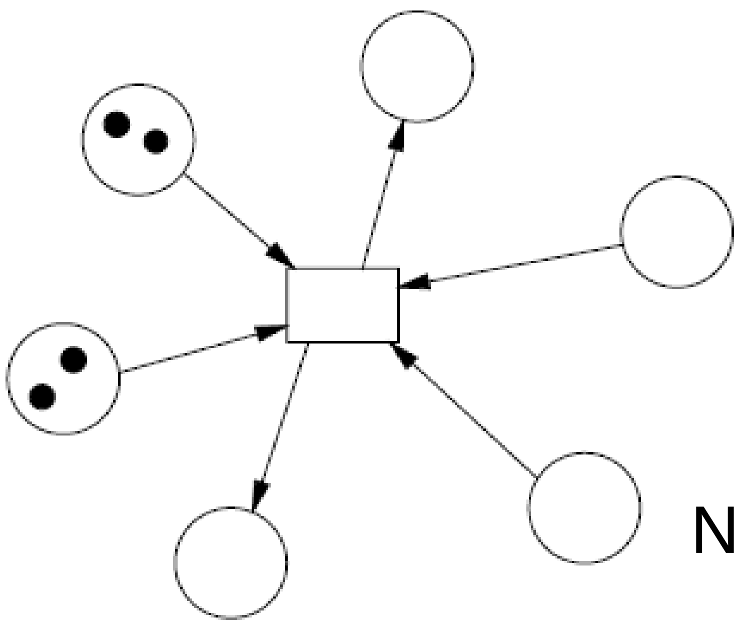
  erhalten[1] (vgl. S. 7).

1) Ausführlicher in [2] beschrieben

Beispiel 1

Netz mit 3 Symmetrien





Netz mit 48 Symmetrien

Symmetrien in Petrinetzen (2)

Für die Beantwortung der Fragen nach

- ➲ Erreichbarkeit von Markierungen und
- ➲ Eigenschaften bestimmter Knoten in PN

werden Symmetrien benutzt.

Symmetrien in Petrinetzen (3)

Problem während der Berechnung des reduzierten Zustandsraumes:

Gegeben: Menge M von (Repräsentanten bereits berechneter Klassen von) Zuständen und ein (gerade erzeugter) Zustand m^*

Symmetrien in Petrinetzen (4)

Problem während der Berechnung des reduzierten Zustandsraumes:

Frage: Gibt es ein m aus M und ein σ aus Σ, so dass $m^* = \sigma(m)$ gilt?

Symmetrien in Petrinetzen (5)

Algorithmen zur Lösung der Frage

A) **FOR ALL** σ aus $\sum$ **DO**

  IF $\sigma(m^*)$ *aus M* **THEN**

    RETURN yes;

  END IF

END FOR

RETURN no;

Symmetrien in Petrinetzen (6)

Algorithmen zur Lösung der Frage

B) **FOR ALL** *m* aus *M* **DO**

IF ex. ein σ *aus* Σ mit $\sigma(m) = m^*$ **THEN**

RETURN yes;

END IF

END FOR

RETURN no;

Symmetrien in Petrinetzen (7)

Zu A	Zu B
○ Suche von $\sigma(m^*)$ in M erfolgt in Hash- oder Baumstruktur von M (effizient) ➲ Laufzeit bestim-mt durch $\lvert\Sigma\rvert$	○ Symmetrie muss ‚on demand' berechnet werden

Symmetrien in Petrinetzen (8)

➲ Beschreibung von Sym. durch Constraints

- Constraints können mir den Eigenschaften der Sym. (F.10) gleichgesetzt werden.
- Restriktionen auf Symmetrien können kombiniert werden
- Es gibt eine Normalform für Constraints

Symmetrien in Petrinetzen (9)

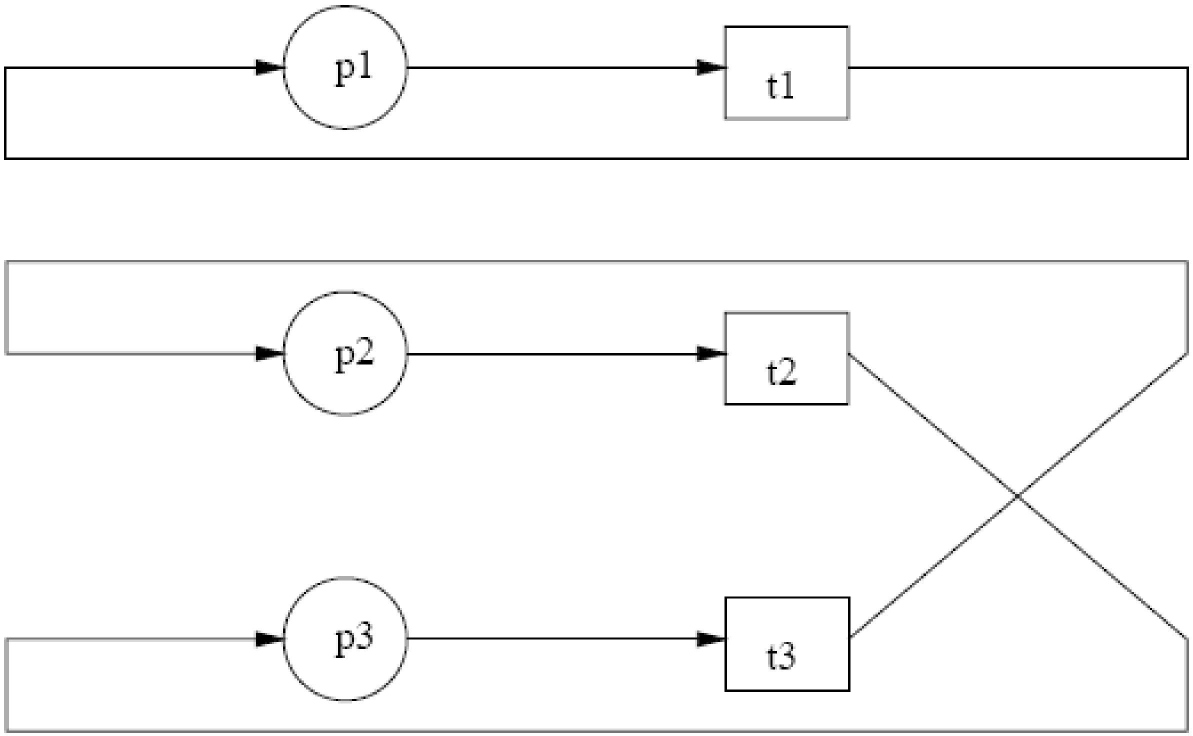
Allgemein besitzen Constraints folgende Form: $A \mapsto B$ mit $A, B \subseteq P \cup T$

Beispiele:

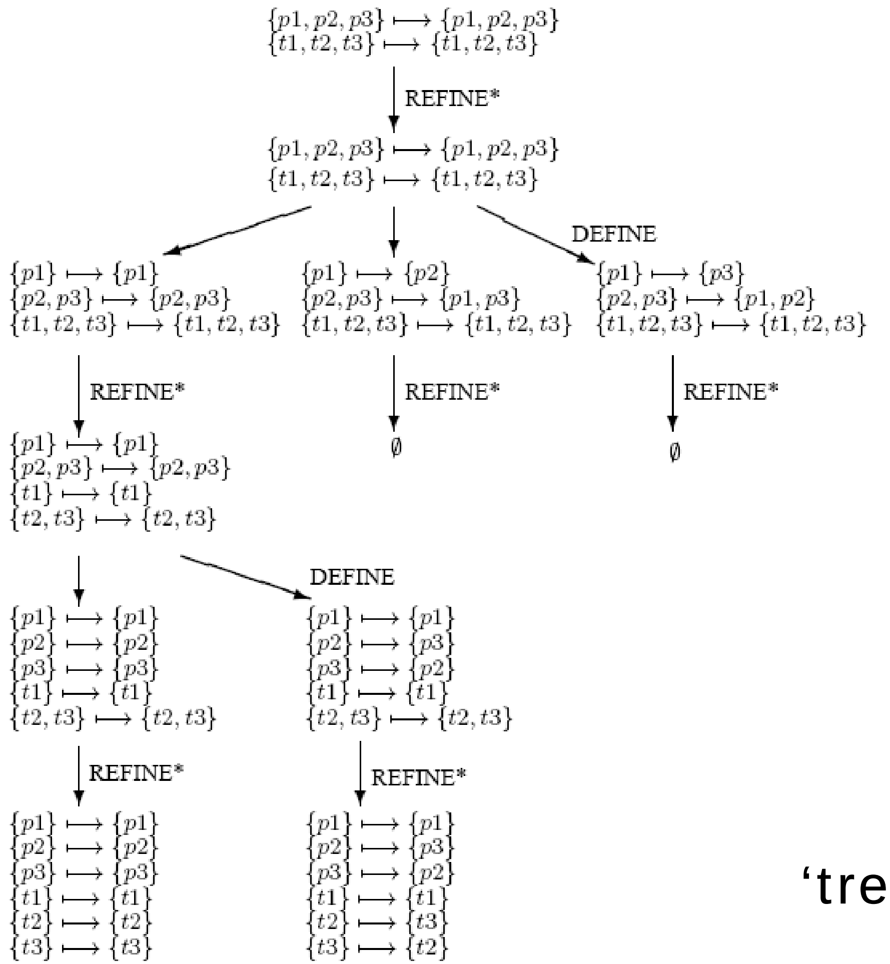
- $C_0 = \{P \mapsto P, T \mapsto T\}^*$
- $C_1 = \{\{x \in P \cup T, \chi(x) = i\} \mapsto \{x \mid x \in P \cup T, \chi(x) = i\} \mid i \in \mathrm{I}\}^*$

*) zur Bedeutung und für weitere Bsp. vgl [2]

Beispiel 2



Symmetrien in Petrinetzen (10)



‘tree exploration‘

Literatur

[1] M. Clarke, Jr, Edmund et al.(1999): *Model Checking*. Cambridge, Mass.: MIT Press

[2] Schmidt, Karsten(2000): *How to calculate symmetries of Petri nets*. In: Acta Informatica 36, S. 545– 590