

# Constraint Satisfaction mit Petri-Netzen

Rudolf J. Kruse, Kurt Lautenbach

Universität Koblenz-Landau  
Institut für Informatik  
Postfach 201 602  
56016 Koblenz

E-Mail: {kruse, laut}@informatik.uni-koblenz.de

## Zusammenfassung

Ein Constraint Satisfaction Problem (CSP) besteht aus Variablen, Wertebereichen und Constraints, die die Belegungskombinationen der Variablen einschränken. Wir zeigen, wie man CSPs mit Hilfe von Pr/T-Netzen modellieren und lösen kann. In den verwendeten Netzen gibt es neben den konventionellen Stellenmarken Transitionsmarken, die sich entgegen der Kantenrichtung bewegen, wenn Stellen schalten. Auch wenn es offensichtliche Parallelen zwischen den Prozessen in diesen Pr/T-Netzen und den CSP-Lösungsprozessen gibt, sind grundlegende Unterschiede zwischen den beiden Paradigmen zu überwinden.

## 1 Constraint Satisfaction Probleme

Wie Regeln sind Constraints Vorschriften, die die Menge aller Lösungen  $L$  bezogen auf einer Grundmenge  $M$  beschreiben. Während jedoch Regeln generativen Charakter haben und vorgeben, wie man von der leeren Menge ausgehend Elemente aus  $L$  erreicht, schränken Constraints die Menge aller Elemente  $M$  auf die Lösungsmenge  $L$  ein. Wir beziehen uns im folgenden bei der Beschreibung von Constraints auf endliche CSPs nach Tsang [7].

Ein CSP verstehen wir als Tripel  $P = (Z, D, C)$  mit einer endlichen Menge Variablen  $Z$ , einer endlichen Wertebereichsfunktion  $D$  und einer endlichen Menge Constraints  $C$ . Wir bezeichnen den Wertebereich einer Variablen  $z \in Z$  mit  $D_z$ . Die Menge der Constraints  $C$  muß sich nicht auf jedes  $z \in Z$  beziehen und kann sogar leer sein. Die Menge der Constraints, die sich auf zwei Variablen  $y, z \in Z$  beziehen, bezeichnen wir mit  $C_{yz} \subseteq C$ . Die Stelligkeit eines Constraints ergibt sich aus der Anzahl der bezogenen Variablen, z.B. ist  $C_{xyz}$  ein dreistelliges Constraint. Einstellige Constraints beziehen sich nur auf eine bestimmte Variable und schränken deshalb den Wertebereich dieser Variablen unmittelbar ein. Man kann Wertebereichsangaben als einstellige Constraints auffassen.

Die Menge der Constraints oder der möglichen Belegungen für eine bestimmte Menge von Variablen kann man intensional (z.B.  $\{(x, y) \mid x + 1 = y\}$ ) oder extensional (z.B.  $\{(0, 1), (1, 2), \dots\}$ ) beschreiben. Die Darstellungsweisen sind gleichwertig und unerheblich für die CSP-Lösungsmethoden. Die Lösung eines CSP ist eine Belegungskombination, die jedem  $z \in Z$  ein Wert aus  $D_z$  zuweist, so daß  $C$  erfüllt ist. Ein CSP  $P$  ist erfüllbar, falls zu  $P$  eine Belegungskombination mit den beschriebenen Eigenschaften existiert. Oft hat ein CSP mehrere Lösungen. Die Aufgabe kann grundsätzlich darin bestehen irgendeine Lösung, die beste Lösung oder alle Lösungen eines CSP zu finden. Hat  $P$  keine Lösung, ist es überspezifiziert, so daß einzelnen Constraints miteinander unvereinbar sind. Dann bleibt die Möglichkeit  $P$  umzuformulieren, indem ausgewählte Constraints aufgeweicht werden.

### 1.1 CSP-Lösungsverfahren

Ein Problem wird typischerweise dann als CSP formuliert, wenn kein spezifischer Algorithmus für das Problem bekannt ist. Die Constraints bilden dann eine deklarative Beschreibung der Lösungsmenge ohne anzugeben, wie man diese Lösungen erhält. Man kann die existierenden Lösungsmethoden für endliche CSPs in drei Klassen einteilen:

### Problemreduktion

Ein CSP  $P$  auf ein CSP  $P'$  reduzieren bedeutet, Belegungskombinationen zu streichen, die in keiner Lösung vorkommen.  $P'$  bleibt zu  $P$  äquivalent in dem Sinne, daß die Variablenmengen und die Menge der Lösungen gleich sind, wird aber leichter lösbar, weil weniger Belegungsmöglichkeiten betrachtet werden müssen.

### Ergebnissuche

Die Lösung eines CSP wird gesucht, indem systematisch Belegungskombinationen für  $Z$  über Belegungskombinationen zu Teilmengen  $Z' \subset Z$  ermittelt werden, die den zugehörigen Constraints genügen. Basismethode ist das chronologische Backtracking, zu dem es diverse Verbesserungen gibt. Beispielsweise kann man den Vorgang als Suchbaum organisieren und darin gleichförmige Teilbäume identifizieren. Suchstrategien werden inzwischen gut verstanden und eignen sich besonders bei Problemen, zu denen nicht alle sondern nur irgendeine oder die beste Lösung gesucht wird.

### Ergebniserzeugung

Die Ergebniserzeugung entspricht der gleichzeitigen Suche in mehreren Ästen des Suchbaumes, wobei zwischendurch immer wieder Problemreduktion betrieben wird. Charakteristisch ist, daß die Menge aller Lösungen konstruktiv ermittelt wird. Die Strategie der Ergebniserzeugung ist somit eine Kombination aus Ergebnissuche und Problemreduktion und vereinigt die Vorteile beider Ansätze miteinander.

## 1.2 Constraint Propagation

Wir werden im folgenden auf die Strategie von Freuder [1] aus der Klasse der Ergebniserzeugungsmethoden näher eingehen. Als Grundidee wird zu einem CSP  $P = (Z, D, C)$  der Potenzmengen-Verband  $G = (V, E)$  über  $Z$  bzgl.  $(\cup, \cap)$  inkrementell aufgebaut. Jeder Knoten enthält die Menge der jeweils möglichen Belegungskombinationen für die Variablen, die er repräsentiert. Wir bezeichnen die Menge aller Knoten zu gleichgroßen Variablenmengen als Schicht. Der Graph wird im folgenden als MP-Graph bezeichnet, weil er zum Schluß das minimale Problem zu  $P$  angibt. Er ist nicht mit dem Constraint-(Hyper-)Graph zu verwechseln, der in der Literatur im Zusammenhang mit CSPs auch häufig besprochen wird und andere Anwendungen hat. Die Struktur eines MP-Graphen für ein CSP mit  $Z = \{a, b, c\}$  wird in Abbildung 1 (links) gezeigt.

Der MP-Graph wird von den einelementigen Variablenmengen ausgehend schichtweise aufgebaut. Nachdem eine neue Schicht hinzugekommen ist, werden für jeden neuen Knoten die möglichen Belegungskombinationen zusammengestellt. Grundlage hierfür sind die Belegungskombinationen der Knoten aus der vorausgehenden Schicht, die Teilmengen bzw. Projektionen des aktuellen Knotens repräsentieren und somit über Kanten mit dem aktuellen Knoten verbunden sind. Die aus diesen Belegungskombinationen ermittelbaren neuen Belegungskombinationen müssen zusätzlich den Constraints für die aktuelle Variablenmenge genügen. Beispielsweise ergibt sich die Belegung des Knotens  $V_{abc}$ , der für die Variablenmenge  $\{a, b, c\}$  steht, aus denjenigen konsistenten Kombinationen der Belegungen von  $V_{ab}, V_{ac}$  und  $V_{bc}$ , die zu dem Constraint  $C_{abc}$  passen.

Anschließend wird für die Knoten der aktuellen Schicht geprüft, ob sich anhand der ermittelten, noch möglichen Belegungen implizit neue Constraints ergeben. Diese Constraints werden in die älteren Schichten und in der Folge zu den Nachbarknoten propagiert. Sie schränken die Menge möglicher Belegungen dort weiter ein, so daß jeder Knoten nach dem jeweils aktuellen Kenntnisstand stets nur die tatsächlich noch möglichen Belegungen zeigt. Eine Propagation kann neue Constraints ergeben und damit neue Propagationen auslösen.

Mit diesen drei Phasen, Schichtaufbau, Belegungsbestimmung und Constraint Propagation, wird der Graph Schicht für Schicht entwickelt. Nachdem die letzte Schicht, die immer aus genau einem, alle Variablen repräsentierenden Knoten besteht, ermittelt ist und die Belegungen bestimmt wurden, ist das CSP gelöst. Führt man eine letzte Propagation durch, ändert sich die Lösungsmenge zwar nicht mehr, der MP-Graph repräsentiert dann aber das minimale Problem zu  $P$ .

## 2 Constraint-Propagation-Petri-Netze

Für die Modellierung mit Petri-Netzen hat das vorgestellte Verfahren von Freuder vielversprechende Eigenschaften. Es gilt das *Lokalitätsprinzip* im MP-Graph, denn bei der Ermittlung möglicher Belegungskombinationen und bei der Verbreitung neuer Constraints gehen jeweils nur die aktuellen Werte der direkten Nachbarknoten ein. Man kann außerdem die Sichtweise einnehmen, daß Belegungskombinationen und Constraints im Graphen wie Marken im Petri-Netz *versendet* werden und in der Folge Neuberechnungen auslösen.

Es gibt jedoch auch prinzipielle Unterschiede zwischen dem CSP-Algorithmus und den Sachverhalten in Petri-Netzen. Der MP-Graph hat Kanten, die zunächst ungerichtet sind, im Laufe der Belegungsermittlung und Constraint-Verbreitung jedoch in beiden Richtungen verwendet werden. Zwischen zwei Knoten  $V_a$  und  $V_{ab}$  wird z. B. zunächst die Menge möglicher Belegungen für  $V_{ab}$  aus den Belegungen zu  $V_a$  berechnet, wobei die zugehörige Kante als  $(a, ab) \in E$  verwendet wird. Danach wird evtl. ein Constraint von  $V_{ab}$  aus im Netz verbreitet, wobei die Kante in Richtung  $(ab, a) \in E$  benutzt wird. Man kann also die Sichtweise einnehmen, daß sich im Laufe der Berechnungen ungerichtete Kanten zur einen oder anderen Seite ausrichten, oder daß man sich auf eine Kantenrichtung festlegt und im Laufe der Berechnungen Werte entlang und entgegen der Kantenrichtung verbreitet werden. Diese Sichtweise führt zu einer Modellierung mit höheren Netzen in denen es Stellen- und Transitionsmarken gibt.

### 2.1 Pr/T-Netze mit Stellen- und Transitions-Marken

In Netzen mit Stellen- und Transitions-Marken existieren zusätzlich Marken, die auf Transitionen liegen und sich entgegen der Kantenrichtung bewegen, wenn Stellen schalten. Während sich bei S/T-Netzen dieser Art einander gegenüberliegende Marken generell blockieren, blockieren sich in Pr/T-Netzen mit zwei Markensorten nur gegenüberliegende *gleiche* Marken. Unterschiedliche Marken passieren einander und können sich dabei gegenseitig beeinflussen.

Die Einführung von Transitionsmarken und einer Stellenschaltregel ist über die Dualisierung markierter Netze motiviert, ein Konzept, das von C. A. Petri selbst stammt. Die Idee, Stellenmarken in das duale Netz als Transitionsmarken zu übernehmen, wurde auch für höhere Netze bereits in [4], [5] und [6] von Lautenbach und Pagnioni vorgestellt. In [2] verwenden wir Transitionsmarken um in S/T-Netzen diagnostisch zu schließen, also um beobachtete Fehler auf mögliche Ursachen zurückzuführen. Eine Modellierung Bayesscher Netze über Pr/T-Netze mit Stellen- und Transitionsmarken, die eng mit den hier beschriebenen Ideen verwandt ist, wird in [3] vorgestellt.

Pr/T-Netze mit Stellen- und Transitions-Marken verwenden wir im folgenden um die vorgestellte Strategie zur Lösung von CSPs zu modellieren. Die konventionellen Stellen-Marken repräsentieren dabei die jeweils möglichen Belegungskombinationen, während die im Pr/T-Netz entgegen der Kantenrichtung fließenden Transitionsmarken die im Graph ebenfalls entgegen der Kantenrichtung propagierten Constraints modellieren. Wenn sich eine Stellen- und eine Transitionsmarke begegnen, beeinflussen sie sich dahingehend, daß die Constraints aus der Transitionsmarke die Belegungskombinationen der Stellenmarke weiter einschränken.

### 2.2 Netzstruktur

Jeden Knoten des Graphen modellieren wir als ein Pr/T-Teilnetz bestehend aus einer Stelle und einer Transition die über ein Kantenpaar miteinander verbunden sind. Die Randknoten zweier Teilnetze werden genau dann über eine Doppelkante miteinander verbunden, wenn ihre Repräsentanten im MP-Graph ebenfalls miteinander verbunden sind. Das Netz zu einem CSP kann über diese konstruktive Beschreibung automatisch erzeugt werden.

Ein MP-Graph und das entsprechende Pr/T-Netz sind beispielhaft für ein abstraktes CSP mit  $Z = \{a, b, c\}$  in Abb. 1 dargestellt. Jeder Knoten  $V_A \in V$  des MP-Graphs modelliert eine bestimmte Variablenkombination  $A \subseteq V$ . Dazu gehört im Netz eine Stelle  $s_A$ , die die möglichen Belegungskombinationen zu  $A$  enthält, und eine Transition  $t_A$ , die die Constraints bzgl.  $A$  trägt.

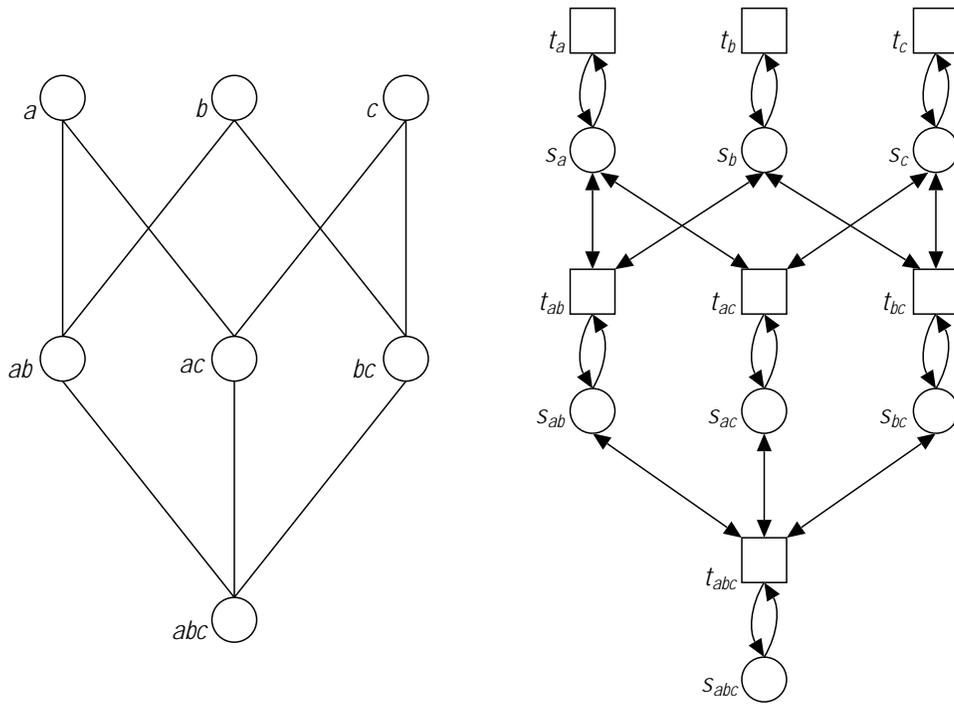


Abbildung 1: MP-Graph und entsprechende Netzstruktur für ein CSP mit  $Z = \{a, b, c\}$

Wenn die Transition  $t_A$  schaltet, berechnet sie die möglichen Belegungen für  $A$  neu, greift dabei auf die aktuellen Belegungen der Teilmengen von  $A$  zu und berücksichtigt die Constraints für  $A$ , die sie selbst enthält. Wenn die Stelle  $s_A$  schaltet, berechnet sie die Constraints für  $A$  neu, greift dabei auf die geltenden Constraints der Obermengen von  $A$  zu und aktualisiert die möglichen Belegungskombinationen für  $A$ , die sie selbst enthält.

Die Entfaltung von Pr/T-Netzen mit Stellen- und Transitions-Marken auf gewöhnliche Pr/T-Netze, die z. B. zu Implementierungszwecken vorgenommen werden kann, ist auf direktem Wege möglich und wird in [3] beschrieben. Zu jeder Stelle wird dabei eine Transition angegeben, die das Schaltverhalten der Stelle modelliert, und zu jeder Transition wird eine Stelle eingeführt, die die T-Marken der Transition beherbergt. Jeder neue Knoten ist über ein Kantenpaar mit seinem Partner verbunden. Das von den neuen Knoten aufgespannte Netz beschreibt das Dual zum ursprünglichen Netz.

### 2.3 Markenstruktur, Schaltregel und Simulation

Da wir die Belegungsbestimmung und Constraint Propagation mit Petri-Netz-Prozessen modellieren wollen, sind Probleme wegen prinzipiellen Unterschieden im Verhalten beider Paradigmen zu lösen. Bei der Propagation wird ein Wert bereits neu berechnet, sobald sich *ein* Nachbarwert geändert hat. Demgegenüber ist eine Transition im Petri-Netz nur dann aktiviert, wenn *alle* Stellen aus dem Vorbereich markiert sind.

Dieser Diskrepanz begegnen wir wie in [3], indem wir die Marken  $m$  als Paar  $m = \langle m_d, m_f \rangle$  strukturieren, bestehend aus einer Datenkomponente  $m_d$  und einer Flußkomponente  $m_f$ .  $m_d$  enthält bei Stellenmarken die Menge der jeweiligen Belegungskombinationen und bei Transitionsmarken die Menge der jeweiligen Constraints.  $m_f$  modelliert die Aktivierungssteuerung und enthält eine Menge von sog. Propagationsmarken, die den Stand der Propagationsvorgänge im Netz zeigen. Eine Menge von Propagationsmarken bildet also die eine Komponente einer Pr/T-Marke und ist nicht mit dieser zu verwechseln. Die Aktivierungsbedingungen beziehen sich ausschließlich auf die Flußkomponenten der aktuellen Markierung, während beim Schalten sowohl die Datenkomponenten als

auch die Flußkomponenten der beteiligten Marken verändert werden. Die Datenkomponente einer Stelle enthält eine Menge von Belegungskombinationen und die Datenkomponente einer Transition enthält eine Mengen von Constraints. Wir beschreiben die Schaltregel genauer anhand der Abbildung 2.

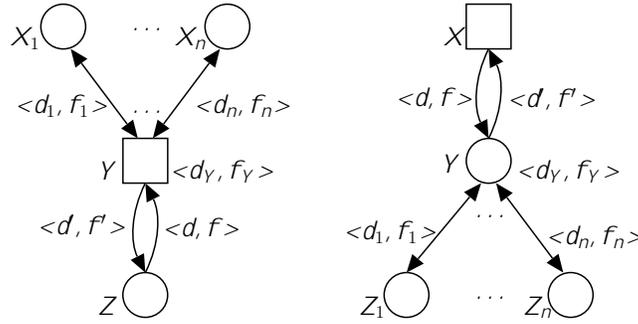


Abbildung 2: Markenstruktur und Kantenbeschriftung zu einer Transition Y (links) und einer Stelle Y, jeweils mit allgemeiner Umgebung, wie sie in Propagations-Pr/T-Netzen vorkommen kann.

Sei  $Y$  eine Transition (Stelle).

- Sei  $F = \bigcup_{i=1}^m f_i$  die Menge aller Propagationsmarken, die im Vorbereich (Nachbereich) von  $Y$  vorkommen.
- $Y$  ist aktiviert, wenn der Vorbereich (Nachbereich) von  $Y$  mindestens eine Propagationsmarke enthält, die nicht in der Flußkomponente von  $Y$  enthalten ist:  $Y$  ist aktiviert, gdw.  $F - f_Y \neq \emptyset$ .
- Wenn  $Y$  schaltet und eine Transition ist, berechnet eine Funktion  $g_t$  aus den Belegungskombinationen des Vorbereichs unter Berücksichtigung der eigenen Constraints die möglichen Belegungskombinationen der Nachbereichsstelle:  $d' := g_t(d_Y, d_1, \dots, d_n)$ .
- Wenn  $Y$  schaltet und eine Stelle ist, projiziert eine Funktion  $g_{s1}$  die Constraints der Nachbereichs-Transitionen den beteiligten Variablen entsprechend auf den Vorbereich. Diese Constraints werden außerdem über eine Funktion  $g_{s2}$  auf die eigenen Belegungskombinationen angewendet:  $d' := g_{s1}(d_1, \dots, d_n)$  und  $d_Y := g_{s2}(d_Y, d')$ .
- Sei  $f^*$  eine der Propagationsmarken, die  $Y$  aktivieren. Die Propagationsmarke  $f^*$  breitet sich im Netz weiter aus und aktiviert Nachbarknoten, indem sie der Flußkomponente von  $Y$  und der des Nachbereichsknotens (Vorbereichsknotens) hinzugefügt wird: Sei  $f^* \in F - f_Y$ , dann ist  $f' := f \cup \{f^*\}$  und  $f_x := f_x \cup \{f^*\}$ .

Ein CSP  $P = (Z, D, C)$  wird wie bei dem Algorithmus von Freuder gelöst, indem zunächst das Constraint-Propagation-Netz zu den einzelnen Variablen aus  $Z$  aufgebaut wird. Dieses Netz entspricht der ersten Schicht im MP-Graph. Davon ausgehend wird inkrementell das Netz um die nächste Schicht erweitert, die neuen Belegungskombinationen werden bestimmt und sich daraus ergebende Constraints werden im Netz verbreitet, bis das Netz vollständig aufgebaut ist. In der ersten Schicht ist die Anfangsmarkierung für jede Stelle der gesamte Wertebereich der zugehörigen Variablen und für jede Transition die Menge der einstelligen Constraints, die sich auf die zugehörige Variable beziehen.

Die initialen Belegungskombinationen einer neuen Schicht werden über eine S-Marken-Simulation durchgeführt. Nachdem eine Schicht neu hinzugekommen ist, sind wegen der fehlenden Propagationsmarken automatisch genau die Transitionen dieser Schicht aktiviert, die die gewünschte Belegungsbestimmung durchführen. Ist im Netz kein Knoten mehr aktiviert, kann man die Stellen der neuen Schicht mit eigenen, neuen Propagationsmarken schalten lassen und somit die rückwärtsgerichtete Verbreitung der sich ergebenden impliziten Constraints initiieren. Die angesprochenen Stellen gehören jeweils zum unteren Rand des Netzes und sind aus struktureller Sicht aktiviert,

weil sie einen leeren Nachbereich haben. Für ein Schalten nach den oben beschriebenen Regeln fehlt jedoch die aktivierende Flußmarke. Neue Propagationen sind solange zu initiieren, bis sich im Netz nichts mehr ändert. Folglich kommen für eine Initiierung nur diejenigen Stellen des unteren Randes in Frage, deren Belegungskombinationen seit der letzten Propagation verändert wurden.

Als Anfangsmarkierung legen wir auf jede Stelle und jede Transition einer neu aufgebauten Schicht die Marke  $\langle \emptyset, \emptyset \rangle$ , um auszudrücken, daß noch keine Belegungskombinationen bzw. Constraints für die neuen Stellen bzw. Transitionen bekannt sind und die neuen Knoten noch an keiner Propagation beteiligt waren. Aus der Anfangsmarkierung und der beschriebenen Netzstruktur folgt, daß auf jedem Knoten während des gesamten Problemlösungsprozesses genau eine Marke liegt. Wenn eine Stelle oder Transition schaltet, wird jede beteiligte Marke entweder unverändert zurückgelegt oder durch genau eine andere Marke ersetzt.

### 3 Zusammenfassung

Wir modellieren ein CSP als Pr/T-Netz mit Stellen- und Transitionsmarken. Die Netzstruktur wird aus dem Potenzmengen-Verband über die Variablen des CSP abgeleitet und kann mit Hilfe einer konstruktiven Beschreibung automatisch erzeugt werden. Die Stellenmarken repräsentieren die jeweils möglichen Belegungskombinationen und die entgegen der Kantenrichtung fließenden Transitionsmarken modellieren die Constraints. Begegnen sich Belegungen und Constraints, beeinflussen diese einander. Mit alternierenden Netzaufbau- und Simulationsphasen wird das CSP gelöst, wobei die Simulationsprozesse die jeweiligen Belegungskombinationen bestimmen und die sich ergebenden Constraints propagieren. Der Lösungsprozeß reduziert das CSP auf das minimale Pendant und die abschließende Markierung zeigt alle Lösungen des zugrundeliegenden CSPs. Der vorgestellte Ansatz gehört zur Klasse der Ergebniserzeugungs-Strategien und basiert auf einem bekannten Constraint-Propagation-Verfahren.

Aus allgemeiner Sicht wird gezeigt, wie man mit Petri-Netzen Vorgänge modellieren kann, die sich auf ungerichtete Graphen beziehen oder auf gerichteten Graphen, bei denen sich die Kantenrichtung ändert. Gleichzeitig bietet die Constraint Satisfaction ein weiteres Anwendungsgebiet für höhere Netze mit Stellen- und Transitionsmarken.

### Literatur

- [1] Freuder, E. C. *Synthesizing constraint expressions*. Communications ACM, November 1978, Vol 21, No 11, 958-966.
- [2] Kruse, R. J., Lautenbach, K. *Diagnostik mit Petri-Netzen*. In: 4. Workshop Algorithmen und Werkzeuge für Petrinetze, Informatik-Bericht Nr. 85, Humboldt-Universität zu Berlin, Berlin, 1997.
- [3] Kruse, R. J., Lautenbach, K. *Bayessche Petri-Netze*. In: 5. Workshop Algorithmen und Werkzeuge für Petrinetze, Forschungsbericht Nr. 694, Universität Dortmund, Dortmund, 1998.
- [4] Lautenbach, K. *Simple Marked-graph-like Predicate Transition Nets*. Arbeitspapiere der GMD 41, St. Augustin, 1983.
- [5] Lautenbach, K., Pagnioni, A. *Liveness and Duality in Marked-graph-like Predicate Transition Nets*. Springer LNCS 188, pp. 331-352, 1985.
- [6] Lautenbach, K., Pagnioni, A. *Invariants and Duality in Predicate Transition Nets and in Coloured Nets*. Arbeitspapiere der GMD 132, St. Augustin, 1985.
- [7] Tsang, E. *Foundations of Constraint Satisfaction*. Academic Press Inc., London, 1993.