

Modellierung von dynamischen Zielen in Agentensystemen mit Ziel/Transitions-Netzen

Dennis Chong¹

Abstract: Die geeignete Darstellung von Zielen ist ein wichtiger Aspekt bei der Modellierung von Agentensystemen. Ziele stellen mentale Zustände eines Agenten dar und können sich zur Laufzeit ändern (dynamische Ziele). Sie sind deshalb als dynamische Elemente auf der Modellebene abzubilden. Für die Modellierung von Agentensystemen mit dynamischen Zielen wird in dieser Arbeit die Klasse der Ziel/Transitions-Netze vorgeschlagen. Der Ansatz beruht auf Prädikat/Transitions-Netzen, einer verbreiteten höheren Petri-Netz-Art.

1 Motivation

Um dem zunehmenden Trend von autonomen Software-Systemen mit verteilten Daten und Lösungen zu begegnen, hat sich in der Softwaretechnik das Agentenkonzept etabliert. Agenten repräsentieren in sich geschlossene Software-Entitäten, die über einen längeren Zeitraum in einer Umgebung persistent sind und die Fähigkeit zu autonomen Aktionen und Reaktionen haben (vgl. [Woo02]). Das Verhalten eines Agenten ist dabei an seine Ziele gekoppelt. Ziele stellen mentale Zustände eines Agenten dar und beschreiben die Beschaffenheit der Umgebung, die er mit seinen Handlungen zu erreichen versucht. Sie können zur Laufzeit Änderungen unterliegen, weil Agenten in der Lage sind, auf Grundlage ihrer Erfahrungen selbstständig Ziele zu definieren, zu verwerfen und zu modifizieren (vgl. [KR91]). Dies kann geschehen, wenn ein Ziel erreicht wurde oder weil es aus anderen Gründen nicht mehr sinnvoll erscheint, es in der ursprünglichen Form beizubehalten. In diesem Zusammenhang bedeutet Dynamik, dass Ziele einerseits das Verhalten des Systems beeinflussen und andererseits vom Verhalten des Systems beeinflusst werden.

Ein verbreiteter Ansatz zur Modellierung von Agenten ist die *Agent UML* [BMO00]. Die Agent UML ist eine Weiterentwicklung der Unified Modeling Language (UML), um die besonderen Aspekte von Agenten zu berücksichtigen. Sie besitzt jedoch keine formale Semantik und kann nicht dazu verwendet werden, formale Aussagen über das Modellverhalten zu machen, weil die Systemabläufe nicht eindeutig festgelegt sind. Für die *formale* Spezifikation von Agenten kommen deshalb überwiegend modal- und temporallogische Ansätze (implizit oder explizit) zum Einsatz. Beispiele sind das BDI-Modell nach Rao und Georgeff [RG95], die intentionale Theorie nach Cohen und Levesque [CL90] und die agentenorientierte Programmiersprache (AOP) nach Shoham [Sho93]. Im Gegensatz

¹ Institute of Artificial Intelligence Methods and Information Mining (AIM), Hochschule Bremerhaven, Fachbereich 2 - Informatik, dchong@hs-bremerhaven.de

zur Agent UML besitzen diese Modelle aber keine graphische Repräsentation. Die Verfügbarkeit einer graphischen Repräsentation kann jedoch in der Praxis ein entscheidendes Kriterium für ein Modell sein, wie der Erfolg der UML zeigt. Formale Ansätze zur Modellierung von Agenten, die gleichzeitig über eine graphische Repräsentation verfügen, basieren in der Regel auf Graphtransformationen oder Petri-Netzen. Beispiele sind [DHK01] und [HKHK⁺07] für Graphtransformationen und [XVIY03] und [MW97] für (höhere) Petri-Netze. In den entsprechenden Ansätzen ist es auf unterschiedliche Weise möglich, Ziele eines Agenten im Modell abzubilden. Die Dynamik von Zielen wird allerdings nicht unterstützt. Aus diesem Grund soll in dieser Arbeit ein neuer, formaler Ansatz vorgeschlagen werden, der über eine graphische Repräsentation verfügt und gleichzeitig dynamische Ziele eines Agenten explizit berücksichtigen kann.

Höhere Petri-Netze besitzen zahlreiche Vorteile, weshalb sie in dieser Arbeit als Ausgangspunkt ausgewählt wurden. Zu diesen Vorteilen zählen: die formale Syntax und Semantik, die graphische Repräsentation, die strukturierte und kompakte Darstellung, die Verbreitung in der Literatur, die Berücksichtigung von Nebenläufigkeit und Nichtdeterminismus und die Ausführbarkeit (Simulation) von Petri-Netzen. In den folgenden Abschnitten soll mit der Einführung von Ziel/Transitions-Netzen die Klasse der Prädikat/Transitions-Netze [Gen87] um die Fähigkeit zur Darstellung von dynamischen Zielen erweitert werden. Zu diesem Zweck wird der weitere Verlauf dieser Arbeit wie folgt gegliedert: In Abschnitt 2 werden die Grundideen von Ziel/Transitions-Netzen informell eingeführt. Die algebraischen Grundlagen werden in Abschnitt 3 behandelt. In Abschnitt 4 wird die Klasse der Ziel/Transitions-Netze formal definiert. Abschnitt 5 diskutiert einige Aspekte zur Mächtigkeit und in Abschnitt 6 wird ein Beispiel für einen Agenten mit dynamischen Zielen präsentiert. Den Abschluss dieser Arbeit bildet der Ausblick in Abschnitt 7.

2 Informelle Einführung

In Prädikat/Transitions-Netzen werden die Netz-Elemente mit algebraischen Ausdrücken beschriftet. Auf diese Weise kann die Zahl der notwendigen Stellen und Transitionen in einem Netz begrenzt werden. Gleichzeitig existiert ein wichtiger Zusammenhang zwischen Prädikat/Transitions-Netzen und logischen Formeln. Die Stellen eines Netzes können als *variable Prädikate* interpretiert werden, so dass sich mit Prädikat/Transitions-Netzen prädikatenlogische Formeln repräsentieren lassen (vgl. [GL81]). Die Extension eines variablen Prädikates ist dabei eindeutig durch die Markierung der entsprechenden Stelle festgelegt. Das Prädikat ist bei einer Markierung genau dann erfüllt, wenn die Stelle eine Marke enthält, die die Argumente des Prädikates in Form eines Tupels enthält. Umgekehrt lassen sich mit logischen Formeln, die variable Prädikate enthalten, auch Markierungen eines Netzes beschreiben. Die Idee besteht nun darin, den Zusammenhang von Stellen und Prädikaten bzw. Markierungen und Formeln auszunutzen, um Aussagen über Zustände des Modells als dynamische Elemente auf der Modell-Ebene abzubilden. Voraussetzung ist ein gesonderter Typ von Stellen, der nicht mit Tupeln, sondern mit Formeln markiert wird.

Abbildung 1 zeigt ein einfaches Beispiel für ein Ziel/Transitions-Netz, das die Verwendung von variablen Prädikaten verdeutlichen soll. In der Anfangsmarkierung sei die Stelle

A mit dem Tupel (0) markiert. Mit dem Schalten der Transition t_1 wird die entsprechende Marke entfernt und eine neue Marke auf A erzeugt. Das hier dargestellte Netz setzt keine Bindung der Variablen im Nachbereich einer Transition durch den Vorbereich voraus, so dass beim Schalten von t_1 die Variable y mit einer beliebigen natürlichen Zahl belegt werden kann. Die durch einen doppelten Rand hervorgehobene Stelle Z kann mit Formeln, d.h. mit Zielen, markiert werden. Für diese Stelle sei die Anfangsmarkierung durch die Formel $A(1)$ gegeben, d.h. das Ziel besteht zunächst darin, das Tupel (1) auf der Stelle A zu erzeugen. A stellt dabei ein variables Prädikat dar, das mit der Stelle A assoziiert wird. Die Transition t_1 ist durch eine Schleife mit der Stelle Z verbunden, die mit der Variable F gewichtet ist. Über die Variable F , die mit beliebigen Formeln belegt werden kann, erhält t_1 Zugriff auf das aktuelle Ziel und verwendet es gleichzeitig im Wächter als Negation. Auf diese Weise kann t_1 nur dann aktiviert werden, wenn das aktuelle Ziel nicht erfüllt ist, d.h. wenn eine entsprechende Marke nicht auf A vorhanden ist. Tritt durch fortlaufendes Schalten schließlich der Fall ein, dass auf A tatsächlich die auf Z geforderte Marke vorhanden ist, bleibt t_1 inaktiv. Stattdessen ist in diesem Fall die Transition t_2 aktiviert, weil nur dann das variable Prädikat $A(z)$ erfüllt ist und die Formel im Wächter von t_2 gilt. Die Aktivierung von t_1 und t_2 hängt also von der Beschaffenheit der Markierung von A ab, obwohl keine direkte Verbindung zwischen t_2 und A existiert. Schaltet t_2 , wird das aktuelle Ziel von Z entfernt und ein neues Ziel (der Nachfolger von z) erzeugt. Daraufhin kann t_1 wieder fortlaufend schalten, bis die gewünschte Marke auf A vorliegt.

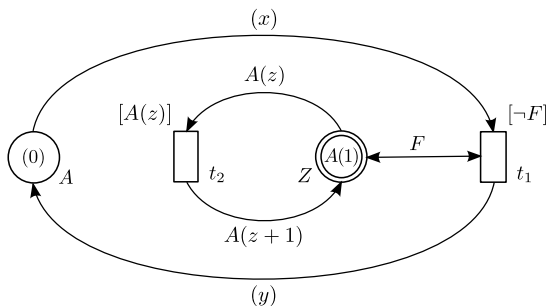


Abb. 1: Ein Ziel/Transitions-Netz mit einer gesonderten Stelle Z für dynamische Ziele.

Die Verwendung von variablen Prädikaten in Ziel/Transitions-Netzen wirkt sich auch auf die Ausdrucksmöglichkeit aus. Ein häufiges Problem bei der Modellierung mit Petri-Netzen ist der Wunsch, die Aktiviertheit einer Transition mit einem so genannten Nulltest zu koppeln. Dabei soll eine Transition nur dann schalten können, wenn eine oder mehrere gegebene Stellen *sauber* sind, d.h. keine Marken enthalten. In beschränkten Netzen kann ein solcher Nulltest durch die Einführung von *Komplementstellen* erfolgen (siehe z.B. [PW03]), aber für unbeschränkte Netze ist dies im Allgemeinen nicht möglich. In Ziel/Transitions-Netzen kann ein solcher Nulltest sehr einfach formuliert werden. Abbildung 2 zeigt zwei Beispiele.

Die Beispiele zeigen, wie sich Systemzustände in Ziel/Transitions-Netzen mit Formeln beschreiben lassen und wie entsprechende Formeln auf der Modell-Ebene mit dem Modellverhalten gekoppelt werden können. Die Formel-Marken können dabei wie gewöhnliche

Marken im Netz konsumiert und produziert werden. Im Folgenden sollen diese Ideen formal definiert werden.

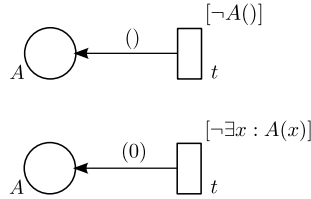


Abb. 2: Oben: Ein Ziel/Transitions-Netz mit einem Nulltest für das 0-stellige Prädikat A . Unten: Nulltest eines 1-stelligen Prädikates. Die Kanten sind für den Nulltest nicht relevant.

3 Algebraische Grundlagen

In diesem Abschnitt werden die algebraischen Grundlagen, die für eine formale Definition von Ziel/Transitions-Netzen notwendig sind, eingeführt.

$\mathbb{N} = \{0, 1, 2, \dots\}$ bezeichne die Menge der natürlichen Zahlen und $\mathbb{B} = \{\top, \perp\}$ die Menge der Wahrheitswerte. Mit R^+ sei die transitive Hülle und mit R^* die reflexiv-transitive Hülle einer (zweistelligen) Relation $R : A \times B$ bezeichnet. Ein Wort $a_1 \dots a_n$ bzw. ein Tupel (a_1, \dots, a_n) über A ist eine endliche geordnete Folge von Elementen aus A , so dass $a_i \in A$ für alle $i \in \{1..n\}$. ε bezeichne das leere Wort, A^* bezeichne die Menge aller Wörter über A und $A^+ = A^* \setminus \{\varepsilon\}$ die Menge aller Wörter über A ohne das leere Wort. Für eine beliebige Menge I bezeichne $(A_i)_{i \in I}$ eine Familie von Mengen, so dass A_i für alle $i \in I$ eine Menge ist und $A_i \cap A_j = \emptyset$ für alle $i, j \in I$ mit $i \neq j$. Für eine Familie $A = (A_i)_{i \in I}$ bezeichne A ebenso die Vereinigung $\bigcup_{i \in I} A_i$.

Signaturen und Strukturen

In Ziel/Transitions-Netzen werden die Netz-Elemente mit algebraischen Ausdrücken beschriftet. Aus diesem Grund müssen zunächst die notwendigen Begriffe wie Signatur und Struktur formal eingeführt werden, bevor die entsprechenden syntaktischen und semantischen Konzepte definiert werden können:

Eine *Signatur* $\Sigma = (S, \Omega, \Pi)$ besteht aus einer endlichen Menge S von *Sorten*, einer Familie $\Omega = (\Omega_w)_{w \in S^+}$ von *Operationssymbolen* und einer Familie $\Pi = (\Pi_w)_{w \in S^*}$ von *Prädikatensymbolen*. Eine Σ -*Struktur* A_Σ für eine Signatur Σ besteht aus einer Menge A_s für jede Sorte $s \in S$, deren Elemente *Individuen* genannt werden, einer Funktion $f_\omega : A_{s_1} \times \dots \times A_{s_n} \rightarrow A_{s_{n+1}}$ für jedes Operationssymbol $\omega \in \Omega_{s_1 \dots s_n s_{n+1}}$ und einer Relation $r_\pi : A_{s_1} \times \dots \times A_{s_n}$ für jedes Prädikatensymbol $\pi \in \Pi_{s_1 \dots s_n}$. Ein Operationssymbol $\omega \in \Omega_s$ mit $s \in S$ heißt *Konstantensymbol* der Sorte s , die entsprechende Funktion $f_\omega : A_s$ heißt *Konstante*.

Die Signatur ist syntaktischer Natur und legt Bezeichner für Sorten, Operationen und Prädikate fest. Auf ihr können Ausdrücke wie Terme und Formeln erklärt werden. Die Struktur ist semantischer Natur und assoziiert die Bezeichner in der Signatur mit konkreten

Mengen, Funktionen und Relationen. Auf diese Weise können die entsprechenden Ausdrücke eindeutig ausgewertet werden. Aus Gründen der Flexibilität werden bei der Modellierung mit Ziel/Transitions-Netzen grundsätzlich mehrsortige Signaturen bzw. Strukturen berücksichtigt.

Im Folgenden soll davon ausgegangen werden, dass für jedes Individuum $a \in A_s$ einer Sorte $s \in S$ genau ein Konstantensymbol $\omega \in \Omega_s$ mit $f_\omega = a$ existiert, d.h. jedes Individuum kann syntaktisch durch genau ein Konstantensymbol der Signatur und semantisch durch genau eine Konstante der Struktur repräsentiert werden. Darüber hinaus wird für die Beispiele von Ziel/Transitions-Netzen in dieser Arbeit die Signatur und Struktur der natürlichen Zahlen mit den gängigen Operationen vorausgesetzt.

Terme und Formeln

Die syntaktischen Regeln zur Bildung von Ausdrücken in Ziel/Transitions-Netzen sind durch Terme und Formeln gegeben:

Sei $\Sigma = (S, \Omega, \Pi)$ eine Signatur. Dann ist $X = (X_w)_{w \in S}$ eine Familie von *Variablen* für Σ mit einer Variablenmenge X_s für jede Sorte $s \in S$.

1. Die Menge der *Terme* einer Sorte $s \in S$ ist wie folgt gegeben:

- (a) Jede Variable $x \in X_s$ ist ein Term der Sorte s .
- (b) Wenn $\omega \in \Omega_{s_1 \dots s_n s}$ ein Operationssymbol ist und t_i für alle $i \in \{1..n\}$ ein Term der Sorte s_i ist, dann ist $\omega(t_1, \dots, t_n)$ ein Term der Sorte s .

2. Die Menge der *Formeln* ist wie folgt gegeben:

- (a) Wenn t_1 und t_2 Terme sind, dann ist $(t_1 = t_2)$ eine Formel.
- (b) Wenn $\pi \in \Pi_{s_1 \dots s_n}$ ein Prädikatensymbol ist und t_i für alle $i \in \{1..n\}$ ein Term der Sorte s_i ist, dann ist $\pi(t_1, \dots, t_n)$ eine Formel.
- (c) Wenn F_1 und F_2 Formeln sind und $x \in X$ eine Variable ist, dann sind auch $(\neg F_1)$, $(F_1 \wedge F_2)$, $(F_1 \vee F_2)$, $(F_1 \rightarrow F_2)$, $(F_1 \leftrightarrow F_2)$, $(\exists x : F_1)$ und $(\forall x : F_1)$ Formeln.

Ein Vorkommen einer Variablen x in einer Formel F heißt genau dann *frei*, wenn x in keiner Teilformel von F der Form $\forall x : G$ oder $\exists x : G$ vorkommt. Jedes Vorkommen einer Variablen x in einem Term t heißt *frei*. Ein Ausdruck (Term oder Formel) heißt *geschlossen*, wenn keine freien Variablen darin vorkommen, sonst *offen*.

Es ist möglich, auf Klammern in Ausdrücken zu verzichten, wo sie nicht unbedingt notwendig sind. Darüber hinaus ist neben der Präfix-Notation auch die Infix-Notation zulässig, wenn sie die Lesbarkeit unterstützt, z.B. in mathematischen Ausdrücken.

Auswertung und Vereinfachung

Die Interpretation eines Ausdrucks hängt von der gewählten Struktur ab und ist im Folgenden nur für geschlossene Ausdrücke erklärt. Die freien Variablen in offenen Termen und Formeln müssen somit zunächst geeignet substituiert werden, bevor die Ausdrücke ausgewertet werden können:

Sei Σ eine Signatur und $X = (X_w)_{w \in S}$ eine Familie von Variablen für Σ . Eine Abbildung $\alpha = (x_1 \mapsto c_1, \dots, x_n \mapsto c_n)$, die Variablen x_i auf Konstantensymbole c_i der selben Sorte abbildet, heißt *Belegung*. Für einen beliebigen Ausdruck e bezeichne $e : (x_1 \mapsto c_1, \dots, x_n \mapsto c_n)$ den Ausdruck, der durch die Ersetzung jeder freien Variable x_i in e durch c_i entsteht. Die *Auswertung* A_Σ von Ausdrücken ist eine Abbildung, die Terme auf Individuen und Formeln auf Wahrheitswerte abbildet:

1. Für einen geschlossenen Term $t = \omega(t_1, \dots, t_n)$ mit $\omega \in \Omega$ ist $A_\Sigma(t) = f_\omega(A_\Sigma(t_1), \dots, A_\Sigma(t_n))$.
2. Für eine geschlossene Formel F ist $A_\Sigma(F)$ wie folgt festgelegt:
 - (a) Für $F = (t_1 = t_2)$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(t_1) = A_\Sigma(t_2)$, sonst $A_\Sigma(F) = \perp$.
 - (b) Für $F = \pi(t_1, \dots, t_n)$ mit $\pi \in \Pi$ ist $A_\Sigma(F) = \top$, wenn $(A_\Sigma(t_1), \dots, A_\Sigma(t_n)) \in r_\pi$, sonst $A_\Sigma(F) = \perp$.
 - (c) Für $F = \neg F_1$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(F_1) = \perp$, sonst $A_\Sigma(F) = \perp$.
 - (d) Für $F = F_1 \wedge F_2$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(F_1) = A_\Sigma(F_2) = \top$, sonst $A_\Sigma(F) = \perp$.
 - (e) Für $F = F_1 \vee F_2$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(\neg((\neg F_1) \wedge (\neg F_2))) = \top$, sonst $A_\Sigma(F) = \perp$.
 - (f) Für $F = F_1 \rightarrow F_2$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(\neg(F_1 \vee F_2)) = \top$, sonst $A_\Sigma(F) = \perp$.
 - (g) Für $F = F_1 \leftrightarrow F_2$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(F_1 \rightarrow F_2) = A_\Sigma(F_2 \rightarrow F_1) = \top$, sonst $A_\Sigma(F) = \perp$.
 - (h) Für $F = \exists x : F_1$ mit $x \in X_s$ und $s \in S$ ist $A_\Sigma(F) = \top$, wenn ein Konstantensymbol a der Sorte s existiert, so dass $A_\Sigma(F_1 : (x \mapsto a)) = \top$, sonst $A_\Sigma(F) = \perp$.
 - (i) Für $F = \forall x : F_1$ mit $x \in X_s$ und $s \in S$ ist $A_\Sigma(F) = \top$, wenn $A_\Sigma(\neg \exists x : \neg F_1) = \top$, sonst $A_\Sigma(F) = \perp$.

Neben der Auswertung von Ausdrücken wird auch der Begriff der Vereinfachung für Ziel/Transitions-Netze benötigt. Die Vereinfachung von Ausdrücken bezeichnet die syntaktische Ersetzung von Termen durch semantisch äquivalente Konstantensymbole:

Sei t ein geschlossener Term und F eine geschlossene Formel. Die Vereinfachung von t ist gegeben durch $\widehat{A}_\Sigma(t) = c$ für ein Konstantensymbol c mit $A_\Sigma(t) = A_\Sigma(c)$. Mit $\widehat{A}_\Sigma(F)$ sei die Vereinfachung der Formel F bezeichnet, in der alle Terme vereinfacht wurden.

Da vorausgesetzt wird, dass zu jedem Individuum genau ein Konstantensymbol existiert, ist die Vereinfachung von Ausdrücken für alle Terme und Formeln eindeutig.

Multimengen

Von grundlegender Bedeutung für die Definition von Ziel/Transitions-Netzen ist auch der Begriff der Multimenge, der sowohl für Kantengewichte als auch für Markierungen verwendet wird. Multimengen sind besondere Mengen, in denen das mehrfache Vorkommen eines Elementes möglich ist, d.h. jedem Element wird eine Häufigkeit zugeordnet:

Sei A eine beliebige nichtleere Menge. Eine *Multimenge* über A ist eine Abbildung $m : A \rightarrow \mathbb{N}$, die jedem Element aus A eine *Häufigkeit* zuordnet. \mathbb{N}^A bezeichne die Menge aller Multimengen über A . Für ein $a \in A$ gilt $a \in m$ genau dann, wenn $m(a) > 0$. m heißt endlich genau dann, wenn $m(a) > 0$ nur für endlich viele $a \in A$ gilt. 0 bezeichnet die leere Multimenge mit $0(a) = 0$ für alle $a \in A$. Für zwei Multimengen $m_1, m_2 \in \mathbb{N}^A$ gilt $m_2 \leq m_1$ genau dann, wenn $m_2(a) \leq m_1(a)$ für alle $a \in A$. Die Addition ist gegeben durch $(m_1 + m_2)(a) = m_1(a) + m_2(a)$ für alle $a \in A$. Für $m_2 \leq m_1$ ist die Subtraktion gegeben durch $(m_1 - m_2)(a) = m_1(a) - m_2(a)$ für alle $a \in A$. Die Multiplikation einer natürlichen Zahl $n \in \mathbb{N}$ mit einer Multimenge $m \in \mathbb{N}^A$ ist gegeben durch $(n \cdot m)(a) = n \cdot m(a)$ für alle $a \in A$.

Für die Schreibweise von Multimengen wird die Mengen- oder die Produktsammennotation verwendet. So ist für die Multimenge $m = (a_1 \mapsto 2, a_2 \mapsto 1, a_3 \mapsto 0, a_4 \mapsto 2)$ über der Menge $A = \{a_1, a_2, a_3, a_4\}$ sowohl $\{a_1, a_1, a_2, a_4, a_4\}$ als auch $2a_1 + a_2 + 2a_4$ zulässig. Eine Funktion f , die auf den Elementen einer Menge A definiert ist, wird bei Bedarf auf Tupel, Mengen und Multimengen über A erweitert, indem f auf jedes Element des Tupels, der Menge oder der Multimenge angewandt wird.

4 Ziel/Transitions-Netze

Bevor die Klasse der Ziel/Transitions-Netze eingeführt werden kann, ist der Begriff der Signatur um eine Familie von so genannten *variablen* Prädikatensymbolen zu erweitern. Im Gegensatz zu den in der Signatur Σ definierten Prädikatensymbolen, deren Extension durch eine Relation in einer Σ -Struktur festgelegt wird, werden variable Prädikatensymbole anhand von Markierungen interpretiert. Die Extension eines variablen Prädikates ist gegenüber der Menge der möglichen Markierungen eines Ziel/Transitions-Netzes also variabel.

Definition 1 (Erweiterte Signatur, Erweiterte Formel) Sei $\Sigma = (S, \Omega, \Pi)$ eine Signatur und $X = (X_w)_{w \in S}$ eine Variablenmenge für Σ . Die Erweiterte Signatur von Σ besteht außerdem aus einer Familie $\widehat{\Pi} = (\widehat{\Pi}_a)_{a \in S^*}$ von variablen Prädikatensymbolen mit $\widehat{\Pi} \cap \Pi = \emptyset$

und einer Variablenmenge \widehat{X} mit $\widehat{X} \cap X = \emptyset$. Eine erweiterte Formel ist eine Formel, in der variable Prädikatsymbole aus $\widehat{\Pi}$ und Variablen aus \widehat{X} als Teilformeln enthalten sein können. Für ein variables Prädikat $\widehat{\pi} \in \widehat{\Pi}$ und eine erweiterte Formel \widehat{F} gilt $\widehat{\pi} \in \widehat{F}$ genau dann, wenn $\widehat{\pi}$ in \widehat{F} enthalten ist.

Da eine Variable $\widehat{x} \in \widehat{X}$ nicht durch Quantoren gebunden werden kann, heißt jedes Vorkommen von \widehat{x} in einer erweiterten Formel frei. Darüber hinaus kann \widehat{x} mit erweiterten Formeln substituiert werden. Im Folgenden sei mit dem Begriff Formel stets eine erweiterte Formel gemeint.

Definition 2 (Ziel/Transitions-Netz) Ein Ziel/Transitions-Netz $ZT = (N, A_N, M_0)$ besteht aus folgenden Bestandteilen:

1. $N = (P_N, T_N, F_N)$ ist ein Netz mit:

- (a) P_N ist eine nichtleere, endliche Menge von Stellen. $\widehat{P}_N \subseteq P_N$ bezeichnet die Menge der Ziel-Stellen und $P_N \setminus \widehat{P}_N$ die Menge der Prädikat-Stellen.
- (b) T_N ist eine nichtleere, endliche Menge von Transitionen, so dass $T_N \cap P_N = \emptyset$ und $T_N \cup P_N \neq \emptyset$.
- (c) $F_N \subseteq (P_N \times T_N) \cup (T_N \times P_N)$ ist eine Menge von Kanten.

Der Vorbereich einer Stelle bzw. einer Transition $x \in P_N \cup T_N$ ist durch $F_N x = \{y \mid (y, x) \in F_N\}$, der Nachbereich durch $x F_N = \{y \mid (x, y) \in F_N\}$ gegeben. Für eine Kante $f = (p, t)$ oder $f = (t, p)$ mit $p \in P_N$ und $t \in T_N$ bezeichne $P_N(f) = p$ die Stelle und $T_N(f) = t$ die Transition der verbindenden Kante.

2. $A_N = (A_\Sigma, A_P, A_T, A_F)$ ist eine Beschriftung mit:

- (a) A_Σ ist eine Σ -Struktur für eine erweiterte Signatur Σ .
- (b) A_P ist eine bijektive Abbildung, die jeder Prädikat-Stelle ein variables Prädikatsymbol zuordnet.
- (c) A_T ist eine Abbildung, die jeder Transition eine erweiterte Formel, genannt Wächter, zuordnet.
- (d) A_F ist eine Abbildung, die jeder Kante f mit $P_N(f) \in P_N \setminus \widehat{P}_N$ eine endliche, nichtleere Multimenge über Tupel von Termen und jeder Kante f mit $P_N(f) \in \widehat{P}_N$ eine endliche, nichtleere Multimenge über Formeln zuordnet. Für eine Kante f mit $P_N(f) \in P_N \setminus \widehat{P}_N$ und ein Tupel $(t_1, \dots, t_n) \in A_F(f)$ sind t_1, \dots, t_n genau dann Terme der Sorten s_1, \dots, s_n , wenn $A_P(P_N(f)) \in \Pi_{s_1, \dots, s_n}$. Für ein $f \notin F_N$ gilt $A_F(f) = \emptyset$.

3. M_0 ist eine (Anfangs-) Markierung, die jeder Prädikat-Stelle $p \in P_N \setminus \widehat{P}_N$ eine endliche Multimenge über Tupel von Individuen und jeder Ziel-Stelle $\widehat{p} \in \widehat{P}_N$ eine endliche Multimenge über geschlossenen Formeln zuordnet. Für eine Prädikat-Stelle $p \in P_N \setminus \widehat{P}_N$ und ein Tupel $(d_1, \dots, d_n) \in M_0(p)$ sind d_1, \dots, d_n genau dann Individuen aus A_{s_1}, \dots, A_{s_n} , wenn $A_p(p) \in \Pi_{s_1, \dots, s_n}$.

Die Struktur eines Ziel/Transitions-Netzes wird durch ein Netz aus Stellen, Transitionen und Kanten festgelegt. Dabei wird die Menge der Stellen in eine Menge von Ziel-Stellen und in eine Menge von Prädikat-Stellen unterteilt. Während Prädikat-Stellen wie in Prädikat/Transitions-Netzen Tupel von Individuen als Marken enthalten, werden Ziel-Stellen mit geschlossenen Formeln markiert. Jede Prädikat-Stelle wird durch eine Bijektion mit einem variablen Prädikat assoziiert und jedes Tupel, das die Stelle markiert, muss die Stelligkeit des entsprechenden Prädikates respektieren. Kanten, die mit Prädikat-Stellen verbunden sind, werden mit nichtleeren Multimengen über Tupel von Termen gewichtet. Sie berücksichtigen ebenfalls die Stelligkeiten des entsprechenden Prädikates. Kanten, die mit Ziel-Stellen verbunden sind, werden mit nichtleeren Multimengen von Formeln gewichtet. Transitionen können ebenfalls Formeln (Wächter) zugeordnet werden. Sie beschreiben zusätzliche Restriktionen für die Aktiviertheit einer Transitioneninstanz (siehe Definition 3).

Die graphische Repräsentation von Ziel/Transitions-Netzen lehnt sich an die übliche Notation für Petri-Netze an: Stellen werden als Kreise, Transitionen als Rechtecke und Kanten als gerichtete Pfeile gezeichnet. Die Prädikat-Stellen werden mit den variablen Prädikatsymbolen beschriftet, während die Ziel-Stellen zur Unterscheidung durch einen doppelten Rand hervorgehoben werden. Der Wächter einer Transition wird in eckigen Klammern notiert. Wenn er lediglich von einer Tautologie repräsentiert wird (der Wächter ist immer erfüllt), kann auf eine Darstellung verzichtet werden.

Für die Schaltregel von Ziel/Transitions-Netzen müssen die Regeln zur Auswertung der variablen Prädikatsymbole festgelegt werden. Wie bereits angedeutet, hängt die Extension eines variablen Prädikates vom Zustand des Ziel/Transitions-Netzes ab und nicht von einer in der Struktur zugeordneten Relation: Ein variables Prädikat ist genau dann erfüllt, wenn die Argumente des Prädikates in Form eines Tupels die assoziierte Stelle markieren. Umgekehrt bedeutet dies, dass die Negation des Prädikates genau dann erfüllt ist, wenn eine entsprechende Marke nicht vorhanden ist.

Wie in Prädikat/Transitions-Netzen reicht es darüber hinaus nicht aus, die Begriffe Aktiviertheit und Folgemarkierung auf einer Transition zu erklären. Für die Auswertung von Kantengewichten und Wächtern wird eine Belegung der Variablen vorausgesetzt, um etwaige Terme und Formeln zu schließen. Aus diesem Grund wird der für Prädikat/Transitions-Netze verwendete Begriff der *Transitioneninstanz* auf Ziel/Transitions-Netze übertragen. Eine Transitioneninstanz wird unter einer Markierung für aktiviert erklärt, wenn der Wächter der Transition erfüllt ist (für die gegebene Markierung und Belegung) und genug Marken im Vorbereich vorhanden sind, um allen ausgewerteten Kantengewichten zu genügen. Das Schalten einer aktivierten Transitioneninstanz konsumiert die identifizierten Tupel und Formeln von den Stellen des Vorbereichs und produziert neue Tupel und

Formeln auf den Stellen des Nachbereichs. Dabei ist zu beachten, dass die Gewichte von Kanten an Ziel-Stellen im Gegensatz zu den Gewichten von Kanten an Prädikat-Stellen nicht vollständig durch die gegebene Struktur ausgewertet werden. Zunächst werden die freien Variablen in den Formeln substituiert. Anschließend werden alle Terme vereinfacht.

Definition 3 (Folgemarkierung) Sei $ZT = (N, A_N, M_0)$ ein Ziel/Transitions-Netz, $t \in T_N$ eine Transition und M eine Markierung von ZT .

1. Sei F eine erweiterte Formel. Für die Auswertung von F bei M wird $A_\Sigma(F)$ um folgenden Fall erweitert: Für $F = \widehat{\pi}(t_1, \dots, t_n)$ mit $\widehat{\pi} \in \widehat{\Pi}$ ist $A_\Sigma(F) = \top$, wenn $(A_\Sigma(t_1), \dots, A_\Sigma(t_n)) \in M(A_{\widehat{P}}^{-1}(\widehat{\pi}))$, sonst $A_\Sigma(F) = \perp$.
2. Sei $\chi(t)$ die Menge aller freien Variablen, die im Wächter von t und in allen Kanten $f = (p, t)$ und $f = (t, p)$ für beliebige $p \in P_N$ vorkommen. Darüber hinaus sei α eine Belegung, die jeder Variablen $x \in X \cap \chi(t)$ einer Sorte $s \in S$ ein Konstantensymbol der Sorte s und jeder Variablen $\widehat{x} \in \widehat{X} \cap \chi(t)$ eine geschlossene Formel zuordnet. Dann heißt $t : \alpha$ Transitioneninstanz.
3. Sei $t : \alpha$ eine Transitioneninstanz. $t : \alpha$ heißt aktiviert unter M und kann schalten genau dann, wenn $A_\Sigma(A_T(t) : \alpha) = \top$ bei M und

$$(a) A_\Sigma(A_F(p, t) : \alpha) \leq M(p) \text{ für alle } p \in P_N \setminus \widehat{P}_N,$$

$$(b) \widehat{A}_\Sigma(A_F(\widehat{p}, t) : \alpha) \leq M(\widehat{p}) \text{ für alle } \widehat{p} \in \widehat{P}_N.$$

4. Sei $t : \alpha$ eine aktivierte Transitioneninstanz in M . Das Schalten von $t : \alpha$ in M erzeugt eine Folgemarkierung M' nach folgender Vorschrift:

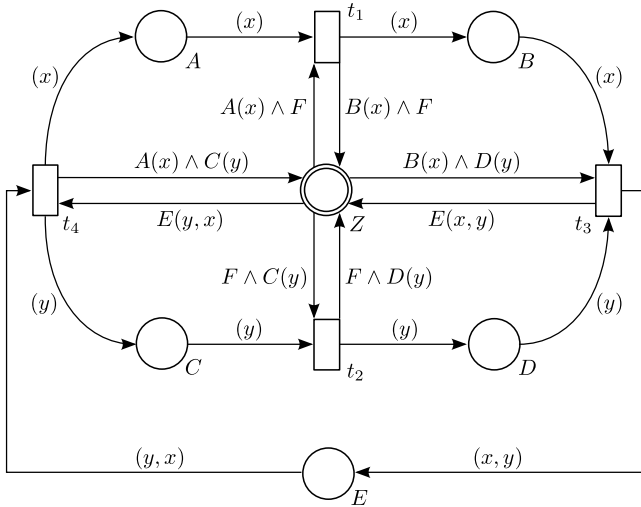
$$(a) M'(p) = M(p) - A_\Sigma(A_F(p, t) : \alpha) + A_\Sigma(A_F(t, p) : \alpha) \text{ für alle } p \in P_N \setminus \widehat{P}_N.$$

$$(b) M'(\widehat{p}) = M(\widehat{p}) - \widehat{A}_\Sigma(A_F(\widehat{p}, t) : \alpha) + \widehat{A}_\Sigma(A_F(t, \widehat{p}) : \alpha) \text{ für alle } \widehat{p} \in \widehat{P}_N.$$

Das Schalten einer Transitioneninstanz $t : \alpha$ in einer Markierung M zu einer Folgemarkierung M' wird mit $M \xrightarrow{t:\alpha} M'$ oder kurz $M \xrightarrow{t} M'$ notiert. R_{ZT} ist die Menge aller erreichbaren Markierungen von ZT .

5. Der Erreichbarkeitsgraph $G_{ZT} = (V, E)$ von ZT ist gegeben durch die Knotenmenge $V = R_{ZT}$ und die Kantenmenge $E = \{(M, t : \alpha, M') \mid M \xrightarrow{t:\alpha} M'\}$. Ein Tupel $(M, t : \alpha, M')$ beschreibt dabei eine Kante von M nach M' , die mit $t : \alpha$ beschriftet ist.

Grundbegriffe der Petri-Netz-Theorie wie Nebenläufigkeit und Konflikt (von Transitioneninstanzen) lassen sich in gewohnter Weise auf Ziel/Transitions-Netze übertragen.



	Markierung						Folgemarkierungen	Belegung			
	A	B	C	D	E	Z		x	y	F	
M_0	(1)		(0)			$A(1) \wedge C(0)$	$M_0 \xrightarrow{t_1:\alpha_1} M_1, M_0 \xrightarrow{t_2:\alpha_2} M_2$	α_1	1	$C(0)$	
M_1		(1)	(0)			$B(1) \wedge C(0)$	$M_1 \xrightarrow{t_2:\alpha_3} M_3$	α_2	0	$A(1)$	
M_2	(1)			(0)		$A(1) \wedge D(0)$	$M_2 \xrightarrow{t_1:\alpha_4} M_3$	α_3	0	$B(1)$	
M_3		(1)		(0)		$B(1) \wedge D(0)$	$M_3 \xrightarrow{t_3:\alpha_5} M_4$	α_4	1	$D(0)$	
M_4					(1,0)	$E(1,0)$	$M_4 \xrightarrow{t_4:\alpha_6} M_5$	α_5	1	0	
M_5	(0)		(1)			$A(0) \wedge C(1)$	$M_5 \xrightarrow{t_1:\alpha_7} M_6, M_5 \xrightarrow{t_2:\alpha_8} M_7$	α_6	0	1	
M_6		(0)	(1)			$B(0) \wedge C(1)$	$M_6 \xrightarrow{t_2:\alpha_9} M_8$	α_7	0	$C(1)$	
M_7	(0)			(1)		$A(0) \wedge D(1)$	$M_7 \xrightarrow{t_1:\alpha_{10}} M_8$	α_8	1	$A(0)$	
M_8	(0)		(1)			$B(0) \wedge D(1)$	$M_8 \xrightarrow{t_3:\alpha_6} M_9$	α_9	1	$B(0)$	
M_9					(0,1)	$E(0,1)$	$M_9 \xrightarrow{t_4:\alpha_5} M_0$	α_{10}	0	$D(1)$	

Abb. 3: Ein Ziel/Transitions-Netz mit einer vollständigen Darstellung aller erreichbaren Markierungen und Transitioneninstanzen

In Abbildung 3 ist ein weiteres Beispiel für ein einfaches Ziel/Transitions-Netz gegeben, das die Aktivierung von Transitionen und die Berechnung von Folgemarkierungen verdeutlichen soll. In der Anfangsmarkierung sei $M_0(A) = (1)$, $M_0(C) = (0)$ und $M_0(Z) = A(1) \wedge C(0)$. Die restlichen Stellen sind mit der leeren Multimenge markiert, d.h. sie enthalten keine Marken. Sobald die beiden Anfangsmarken von A und C mit dem Schalten von t_1 und t_2 auf B und D abgelegt wurden, ist t_3 aktiviert. Wenn t_3 schaltet, wird eine zusammengesetzte Marke auf E erzeugt, die wiederum t_4 aktiviert. Mit dem Schalten von t_4 werden die Anfangsmarken daraufhin wieder in umgekehrter Reihenfolge auf A und C abgelegt. Anschließend können alle Transitionen erneut schalten, bis die Anfangsmarkierung des Netzes wieder erreicht ist. Darüber hinaus aktualisiert jede Transition die auf Z vorhandene Formel, so dass sie in jedem Zustand die aktuelle Markierung der Prädikatstellen in Form einer Konjunktion repräsentiert. Zu diesem Zweck wird bei Bedarf auf eine Variable F zugegriffen, die mit entsprechenden Teilformeln belegt werden kann. Auf

der Ziel-Stelle Z kann somit stets der aktuelle Zustand des Netzes abgelesen werden. Die erreichbaren Markierungen und die entsprechenden Transitioneninstanzen des Netzes sind ebenfalls in Abbildung 3 dargestellt.

Das Beispiel zeigt, dass Zustände eines Systems sehr einfach durch entsprechende Formeln auf einer Ziel-Stelle beschrieben werden können. Eine derartige Stelle kann für die Modellierung von dynamischen Zielen eines Agenten verwendet werden, indem jedes Ziel durch eine Formel auf der Stelle repräsentiert wird. Durch den üblichen Konsum und die Produktion von Marken beim Schalten des Netzes, können Ziele erstellt, gelöscht und geändert werden. Darüber hinaus lässt sich das Verhalten des Systems durch ihre Verwendung in den Wächtern von Transitionen beeinflussen. Ein konkretes Beispiel für einen einfachen Agenten mit dynamischen Zielen und dem entsprechenden Ziel/Transitions-Netz wird im nächsten Abschnitt dargestellt.

5 Mächtigkeit

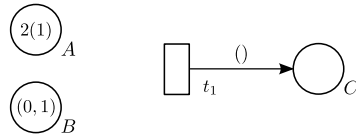
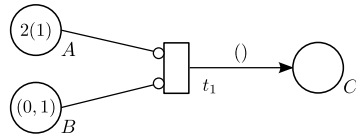
Offensichtlich sind Prädikat/Transitions-Netze ein Spezialfall von Ziel/Transitions-Netzen mit $\widehat{P}_N = \emptyset$. Darüber hinaus werden keine erweiterten Formeln in Wächtern verwendet und alle Tupel in den Kantenbeschriftungen enthalten ausschließlich Variablen. Jedes Prädikat/Transitions-Netz kann somit als Ziel/Transitions-Netz dargesellt werden. Die Umkehrung gilt nicht, wie der folgende Abschnitt zeigt.

Prädikat/Transitions-Netze besitzen nicht die selbe Mächtigkeit wie Turingmaschinen, d.h. mit ihnen können nicht alle berechenbaren Funktionen berechnet werden. Das Problem kann auf die Unfähigkeit von Prädikat/Transitions-Netze zurückgeführt werden, einen Nulltest auf einer Stelle durchzuführen. Es ist zwar möglich, durch Anliegen einer Transition im Nachbereich zu prüfen, ob sich *mindestens* eine Marke auf einer Stelle befindet. Es ist im Allgemeinen aber nicht möglich, zu prüfen, ob sich *keine* Marke auf der Stelle befindet. Eine Transition, die bei einer *sauberen* Stelle aktiviert ist, ist auch bei jeder anderen Anzahl von Marken auf der Stelle aktiviert. Es existieren Erweiterungen wie z.B. Inhibitor-Netze, um diesem Problem zu begegnen. In Inhibitor-Netzen wird durch eine Abbildung $I : T_N \rightarrow 2^{P_N}$ für jede Transition eine Menge von Inhibitoranten festlegt (siehe z.B. [PW03]). Inhibitoranten verlaufen dabei stets von einer Stelle zu einer Transition und werden mit einem Kreis statt mit einer Pfeilspitze gezeichnet. Mit Inhibitoranten wird die Aktiviertheit einer Transition $t \in T_N$ unter einer Markierung M eingeschränkt, indem zusätzlich verlangt wird, dass alle Stellen in $I(t)$ keine Marken enthalten dürfen, d.h. $\forall p \in I(t) : M(p) = 0$.

Wie bereits in Abschnitt 2 angedeutet, lassen sich Nulltests ebenfalls mit Ziel/Transitions-Netzen realisieren. Dazu werden entsprechende Wächter mit variablen Prädikaten eingeführt, die die selben Restriktionen beschreiben wie Inhibitoranten in Inhibitor-Netzen. Ziel/Transitions-Netze können das Verhalten von Inhibitor-Netzen also simulieren. Ein Beispiel ist in Abbildung 4 dargestellt.

Theorem 1 *Zu jedem Prädikat/Transitions-Netz H mit Inhibitorbogen existiert ein verhaltensgleiches Ziel/Transitions-Netz $ZT(H)$, d.h. H und $ZT(H)$ haben den selben Erreichbarkeitsgraphen.*

Beweis: $ZT(H)$ wird mit dem selben Netz, den selben Beschriftungen und der selben Anfangsmarkierung wie H konstruiert. Für jeden Inhibitorbogen von einer Stelle $p \in I(t)$ zu einer Transition $t \in T_N$ im Inhibitor-Netz H wird eine Formel $G_p = \exists x_1, \dots, \exists x_n : A_p(p)(x_1, \dots, x_n)$ konstruiert, so dass x_1, \dots, x_n genau dann Variablen aus X_{s_1}, \dots, X_{s_n} sind, wenn $A_p(p) \in \Pi_{s_1, \dots, s_n}$. Der Wächter von t im Ziel/Transitions-Netz $ZT(H)$ kann dann durch $A_T(t) = \bigwedge_{p \in I(t)} (\neg G_p)$ erweitert werden. Befindet sich in einer beliebigen Markierung M mindestens ein Tupel auf eine Stelle $p \in I(t)$ im Inhibitor-Netz H , dann ist das Konjunktionsglied $\neg G_p$ im Wächter von t des Ziel/Transitions-Netz $ZT(H)$ nicht erfüllt und t kann nicht aktiviert unter M . Der Wächter $A_T(t)$ in $ZT(H)$ repräsentiert somit genau die Restriktion, die die entsprechenden Inhibitorbogen von t in H beschreiben. Alle Zustände und Zustandsübergänge von H und $ZT(H)$ sind daher identisch. \square



$$A_T(t_1) = [(\neg \exists x : A(x)) \wedge (\neg \exists y : \exists z : B(y, z))]$$

Abb. 4: Oben: Ein Inhibitor-Netz mit zwei Inhibitorbogen. Unten: Ein verhaltensgleiches Ziel/Transitions-Netz ohne Inhibitorbogen. In beiden Fällen ist t_1 nicht aktiviert, da sowohl die Stelle A als auch die Stelle B Marken enthält.

Korollar 1 *Ziel/Transitions-Netze sind turingmächtig.*

Beweis: Folgt aus Theorem 1 und dem Beweis, dass Inhibitor-Netze im Allgemeinen turingmächtig sind (siehe z.B. [PW03, S. 166]). \square

6 Anwendungsbeispiel

In diesem Abschnitt soll ein einfacher Agent mit dynamischen Zielen als Ziel/Transitions-Netz modelliert werden. Als Beispiel dient eine einfache Variante des klassischen BlocksWorld-Szenarios aus [Woo02]. In diesem Szenario existieren unterschiedliche Klötzchen, die aufeinander gestapelt werden können. Die Aufgabe des Agenten besteht darin, durch Auf- und Abstellen von Klötzchen eine vordefinierte Anordnung zu erzeugen. Dabei soll er jedoch das unnötige Umstellen von Klötzchen vermeiden, d.h. die Lösung soll

nicht darin bestehen, zunächst alle vorhandenen Stapel abzubauen. Darüber hinaus wird in diesem Beispiel davon ausgegangen, dass jeder Stapel aus nicht mehr als zwei Klötzchen bestehen kann.

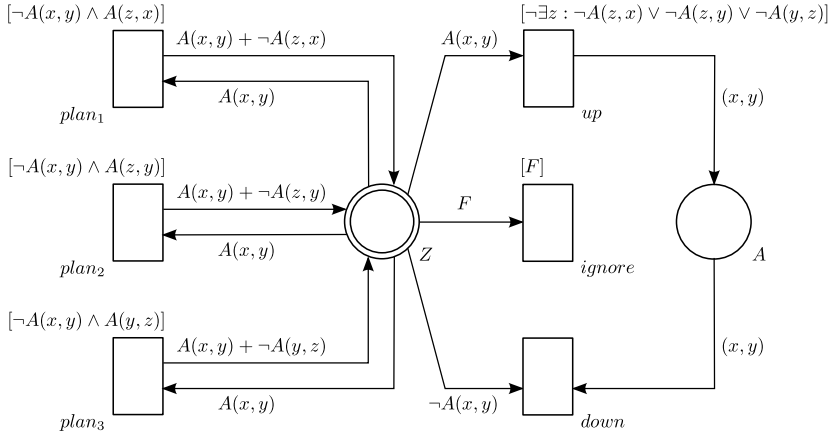


Abb. 5: Ein BlocksWorld-Agent als Ziel/Transitions-Netz.

Abbildung 5 zeigt ein Ziel/Transitions-Netz, das einen BlocksWorld-Agenten und seine Umgebung modelliert. Die Umgebung des Agenten wird durch die Prädikat-Stelle A repräsentiert, in der die aktuelle Anordnung der Klötzchen fixiert ist. Ein Tupel (x, y) bedeutet, dass das Klötzchen x auf dem Klötzchen y gestapelt wurde. Existiert für ein Klötzchen x kein entsprechendes Tupel, dann befindet sich x auf dem Boden. Die (Teil-)Ziele des Agenten werden durch positive und negative Literale auf der Ziel-Stelle Z festgehalten und beschreiben die gewünschte Anordnung der Klötzchen. Positive Literale legen fest, welche Klötzchen aufeinander zu stapeln sind. Negative Literale legen fest, welche Klötzchen nicht aufeinander gestapelt werden dürfen. Als Zielvorgabe sind aber auch unvollständige Konfigurationen zulässig. Das Aufstellen eines Klötzchens findet durch Schalten der Transition up statt. Sie entfernt ein Literal der Form $A(x, y)$ von der Ziel-Stelle Z und stellt das Klötzchen x auf das Klötzchen y , indem das Tupel (x, y) auf A produziert wird. Der Wächter von up garantiert, dass ein Stapelvorgang nur dann durchgeführt werden kann, wenn x und y nicht verstellt sind und y nicht bereits auf ein anderes Klötzchen gestellt wurde. Die Transition $down$ baut einen vorhandenen Stapel (x, y) wieder ab, wenn ein Literal der Form $\neg A(x, y)$ auf Z vorliegt. Dabei wird das Literal mit dem Schalten von $down$ ebenfalls wieder von Z entfernt. Sollte ein beliebiges Ziel auf Z bereits erfüllt sein, dann kann es vom Agenten ignoriert und mit dem Schalten der Transition $ignore$ gelöscht werden. Die Transitionen $plan_1$, $plan_2$ und $plan_3$ repräsentieren mentale Prozesse des Agenten. Sie sind immer dann aktiviert, wenn ein Ziel auf Z einen Stapel (x, y) verlangt, der Stapel aber zunächst nicht erzeugt werden kann. Dies ist der Fall, wenn ein Klötzchen z das Klötzchen x oder y verstellt ($plan_1$ oder $plan_2$) oder wenn das Klötzchen y bereits auf ein Klötzchen z gestellt wurde ($plan_3$). Die Wächter der Transitionen realisieren die notwendigen Fallunterscheidungen. Darüber hinaus setzen sie durch das Konjunktionsglied $\neg A(x, y)$ voraus, dass der geforderte Stapel noch nicht existiert. Wenn $plan_1$, $plan_2$ und $plan_3$ schalten, werden neue Ziele auf der Ziel-Stelle Z erzeugt, um x und y frei zu stellen. Auf diese Weise kann ein BlocksWorld-Agent ausgehend von einer beliebigen

Anfangs-Konfiguration der Klötzchen in A nacheinander alle notwendigen Schritte durchführen, um die Ziel-Konfiguration in Z zu erreichen. Abbildung 6 zeigt einen möglichen System-Ablauf für die Anfangs-Konfiguration $M_0(G) = (1, 2) + (3, 4) + (5, 6) + (7, 8)$ und die Ziel-Konfiguration $M_0(Z) = A(4, 1) + A(6, 5) + \neg A(3, 2)$.

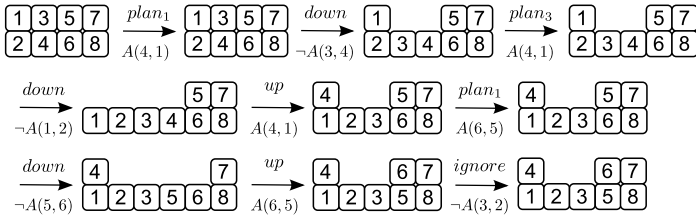


Abb. 6: Ein Ablauf eines BlocksWorld-Agenten. Die gezeigten Klötzchen entsprechen der Markierung der Prädikat-Stelle A . Neben den Transitionen sind auch die Literale der Ziel-Stelle Z dargestellt, die in jedem Schaltvorgang berücksichtigt wurden.

Das Beispiel in diesem Abschnitt zeigt einige Elemente, die den Unterschied zu klassischen Prädikat/Transitions-Netzen verdeutlichen. So ist die Unterscheidung zwischen dem Aktivwerden der Transitionen $plan_1$, $plan_2$, $plan_3$ und up auf entsprechende Wächter zurückzuführen, die die Beschaffenheit der Markierung von A abfragen. Diese Abfragen werden durch die Verwendung von variablen Prädikaten realisiert und sind in dieser Form nicht in Prädikat/Transitions-Netzen möglich. Dies gilt ebenfalls für die Darstellung der positiven und negativen Literale (Ziele). Um diese Mechanismen in Prädikat/Transitions-Netzen zu realisieren und einzubetten, sind komplexere Netz-Strukturen notwendig, d.h. weitere Stellen, Transitionen und Kanten werden benötigt. In Ziel/Transitions-Netze können die entsprechenden Abfragen und Ziele intuitiv und kompakt modelliert werden.

7 Ausblick

In dieser Arbeit wurde mit der Klasse der Ziel/Transitions-Netze ein Ansatz vorgeschlagen, um dynamische Ziele von Agenten mit Petri-Netzen zu modellieren. Auf diese Weise lassen sich die Vorteile von Petri-Netzen auch bei der Modellierung von Agentensystemen nutzen. Für die Darstellung von ziel-orientiertem Verhalten in Agentensystemen soll in den kommenden Arbeiten eine entsprechende Ablaufsemantik für Ziel/Transitions-Netze entwickelt werden, die auf einer kausalen Semantik (Prozesse, siehe z.B. [DFO97]) basiert. Mit der Einführung einer geeigneten Ablaufsemantik wird ein zentraler Vorteil von Ziel/Transitions-Netzen deutlich: Ziele und Zustände können in einem Zusammenhang dargestellt werden, so dass ihre Wechselwirkung untersucht werden kann. Auf diese Weise ist es möglich, ziel-orientiertes Verhalten von Agenten nachzuweisen. Dies ist insbesondere für die Validierung von Agenten-Modellen von Bedeutung.

Literaturverzeichnis

- [BMO00] B. Bauer, J. Müller und J. Odell. Agent UML: A Formalism for Specifying Multiagent Software Systems. In P. Ciancarini und M. J. Wooldridge, Hrsg., *Agent-Oriented Software Engineering: First International Workshop*, Lecture Notes in Computer Science, Volume 1957, Seiten 91–104. Springer, 2000.
- [CL90] P. R. Cohen und H. J. Levesque. Intention Is Choice with Commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.
- [DFO97] J. Desel, T. Freytag und A. Oberweis. Causal-semantic-based simulation and validation of high-level Petri nets. In A. R. Kaylan und A. Lehmann, Hrsg., *Proceedings of 11th European Simulation Multiconference*, Seiten 826–831, 1997.
- [DHK01] R. Depke, R. Heckel und J. Malte Küster. Agent-Oriented Modeling with Graph Transformation. In P. Ciancarini und Michael Wooldridge, Hrsg., *Agent-Oriented Software Engineering: First International Workshop*, Jgg. 1957 of *Lecture Notes in Computer Science*, Seiten 105–120. Springer, 2001.
- [Gen87] H. J. Genrich. Predicate/Transition Nets. In W. Brauer, W. Reisig und G. Rozenberg, Hrsg., *Petri Nets: Central Models and Their Properties. Advances in Petri Nets*, Lecture Notes in Computer Science, Volume 254, Seiten 207–247. Springer, 1987.
- [GL81] H. J. Genrich und K. Lautenbach. System Modelling with High-Level Petri Nets. *Theoretical Computer Science*, 13:109–136, 1981.
- [HKHK⁺07] K. Hölscher, R. Klempien-Hinrichs, P. Knirsch, H.-J. Kreowski und S. Kuske. Autonomous Units: Basic Concepts and Semantic Foundation. In M. Hülsmann und K. Windt, Hrsg., *Understanding Autonomous Cooperation and Control in Logistics - The Impact of Autonomy on Management, Information, Communication and Material Flow*, Seiten 103–120. Springer, 2007.
- [KR91] L.P. Kaelbling und S.J. Rosenschein. Action and planing in embedded agents. In P. Maes, Hrsg., *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, Seiten 35–48. MIT Press, 1991.
- [MW97] D. Moldt und F. Wienberg. Multi-Agent-Systems based on Coloured Petri Nets. In P. Azema und G. Balbo, Hrsg., *Application and Theory of Petri Nets 1997: 18th International Conference*, Jgg. 1248 of *Lecture Notes in Computer Science*, Seiten 241–249. Springer, 1997.
- [PW03] L. Priese und H. Wimmel. *Theoretische Informatik: Petri-Netze*. Springer, 2003.
- [RG95] A. S. Rao und M. P. Georgeff. BDI agents: From theory to practice. In V. Lesser und L. Gasser, Hrsg., *Proceedings of the First International Conference on Multi-Agent Systems*, Seiten 312–319. MIT Press, 1995.
- [Sho93] Y. Shoham. Agent-oriented Programming. *Artificial Intelligence*, 60(1):51–92, 1993.
- [Woo02] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley, 2002.
- [XVIY03] D. Xu, R. Volz, T. Ioerger und J. Yen. Modeling and Analyzing Multi-Agent Behaviors Using Predicate Transition Nets. *International Journal of Software Engineering and Knowledge Engineering*, 13:103–124, 2003.