

# Synthese von S/T-Netzen aus unendlichen partiellen Sprachen

Robin Bergenthum, Robert Lorenz, Sebastian Mauser

Lehrstuhl für Angewandte Informatik  
Katholische Universität Eichstätt-Ingolstadt, Eichstätt, Germany  
e-mail: vorname.name@ku-eichstaett.de

## Zusammenfassung

In dieser Arbeit diskutieren Term-basierte endliche Repräsentationen eingeschränkter Klassen unendlicher partieller Sprachen, aus denen sich effektiv mit Hilfe einer Regionen-Theorie S/T-Netze synthetisieren lassen. Die Algorithmen basieren auf einem kürzlich in [Lor06] vorgestellten Konzept für Regionen partieller Sprachen und einem daraus abgeleiteten Synthesealgorithmus für endliche partielle Sprachen.

## 1 Einführung

In dieser Arbeit werden wir Erweiterungen bekannter Konzepte zur Synthese von System-Modellen aus Verhaltensmodellen untersuchen. Als System-Modelle betrachten wir hierbei Stellen/Transitions-Netze (S/T-Netze). Das Verhalten von S/T-Netzen lässt sich bezüglich verschiedener Semantiken angeben, deshalb formuliert man das sog. *Synthese-Problem* allgemein wie folgt immer bezüglich einer vorgegebenen Semantik:

**Gegeben:** Ein Modell des Verhaltens.

**Gesucht:** Ein S/T-Netz, dessen Verhalten bzgl. der vorgegebenen Semantik dem gegebenen Verhaltensmodell entspricht.

Für (Sprachen- und Graphen-basierte) Verhaltensmodelle bzgl. der sog. *sequentiellen Semantik* und *Schrittsemantik* [BD96] gibt es bereits effiziente (polynomiale) Algorithmen zur Lösung dieses Synthese-Problems. In [LJ06, Lor06] wurden dies Techniken für die *Halbordnungsemantik* erweitert. Hierbei wurden in einem Sprachen-basierten Ansatz *partielle Sprachen* als Verhaltensmodell bzgl. der Halbordnungsemantik betrachtet. Partielle Sprachen sind (im allgemeinen unendliche) Mengen von sog. *beschrifteten partiellen Ordnungen (BPOs)*, d.h. Mengen partiell geordneter Ereignisse, die mit Aktionsnamen beschriftet sind. In einer BPO werden geordnete Ereignisse als kausal abhängig interpretiert und ungeordnete Ereignisse als kausal unabhängig. Zwei Ereignisse bezeichnen wir hierbei als kausal unabhängig, falls sie in jeder beliebigen Reihenfolge und auch gleichzeitig eintreten können.

Abbildung 1 zeigt eine partielle Sprache, die aus zwei BPOs besteht. Die zweite BPO (von rechts) lässt sich folgendermaßen interpretieren: Im initialen Zustand kann eine Aktion  $a$  eintreten, und danach können die Aktionen  $a$  und  $b$  unabhängig voneinander eintreten. Im Fall partieller Sprachen lautet das Synthese-Problem dann wie folgt:

**Gegeben:** Eine partielle Sprache  $L$  über einer endlichen Menge von Beschriftungen.

**Gesucht:** Ein S/T-Netz, dessen Menge partiell geordneter Ausführungen der gegebenen Sprache  $L$  entspricht.

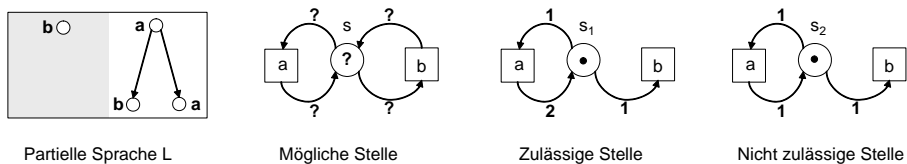


Abbildung 1: Eine partielle Sprache mit Beispielen von zulässigen und nicht zulässigen Stellen.

Eine *partiell geordnete Ausführung* eines S/T-Netzes ist eine BPO, deren Ereignisse mit Transitionsnamen des S/T-Netzes beschriftet sind, s.d. alle Transitionen in der angegebenen kausalen Ordnung schalten können. Die Menge der (partiell geordneten) Ausführungen eines S/T-Netzes lässt sich formal definieren als die Menge aller Sequentialisierungen von Abläufen des Netzes. Ein Ablauf entsteht aus einem Prozess des Netzes, falls man die Bedingungen des Prozesses weglässt und nur noch die partielle Ordnung zwischen dessen Ereignissen betrachtet.

Es ist klar, dass durch die endliche Menge der Beschriftungen von  $L$  bereits die Transitionsmenge des zu synthetisierenden S/T-Netzes gegeben ist. Gleich beschriftete Ereignisse von BPOs in  $L$  stehen für das Schalten ein und derselben Transition in verschiedenen Zuständen des Systems. Betrachtet man das S/T-Netz mit leerer Stellenmenge, das nur aus diesen Transitionen besteht, so ist offenbar jede BPO in  $L$  eine partiell geordnete Ausführung dieses Netzes. Das liegt daran, dass zwischen den Transitionen keine kausale Abhängigkeiten (in Form von Stellen) existieren und sie beliebig oft in beliebiger Reihenfolge schalten können. Deshalb wird dieses Netz auch sehr viele partiell geordnete Ausführungen haben, die nicht in  $L$  spezifiziert wurden. Also versucht man, das Verhalten dieses Netzes einzuschränken, indem man durch zusätzliche Stellen kausale Abhängigkeiten zwischen den Transitionen erzeugt (siehe Abbildung 1). Eine Stelle lässt sich hierbei definieren durch ihre initiale Markierung und die Kantengewichte ihrer Verbindungs-Flusskanten zu allen Transitionen (ein Kantengewicht von 0 soll hierbei bedeuten, dass keine entsprechende Flusskante existiert). Es gibt grundsätzlich zwei Sorten möglicher zusätzlicher Stellen:

- **Nicht zulässige Stellen:** Es kann vorkommen, dass eine BPO aus  $L$  keine partiell geordnete Ausführung des zugehörigen ("ein-stelligen") S/T-Netzes ist, das nur diese eine zusätzliche Stelle hat. In diesem Fall schränkt die Stelle das Verhalten offenbar zu stark ein und wir nennen sie *nicht zulässig*.
- **Zulässige Stellen:** Wenn alle BPOs aus  $L$  partiell geordnete Ausführungen des zugehörigen "ein-stelligen" S/T-Netzes ist, so schränkt die Stelle das Verhalten offenbar nicht zu stark ein und wir nennen sie *zulässig*.

Abbildung 1 zeigt eine zulässige Stelle  $s_1$  und eine nicht zulässige Stelle  $s_2$ . Die Stelle  $s_1$  ist zulässig, da offenbar beide BPOs in der angegebenen partiellen Sprache  $L$  Ausführungen bzgl. dieser Stelle sind: Anfangs können  $a$  oder  $b$  schalten und nach dem Schalten von  $a$  liegen zwei Marken in der Stelle  $s_1$ , s.d. dann  $a$  und  $b$  unabhängig voneinander schalten können. Die Stelle  $s_2$  ist nicht zulässig, da zwar anfangs  $a$  und  $b$  schalten können, aber nach dem Schalten von  $a$  nur eine Marke in der Stelle  $s_2$  liegt, womit dann  $a$  und  $b$  nicht gleichzeitig und damit auch nicht unabhängig voneinander aktiviert sind. Die grundsätzliche Idee ist nun, alle zulässigen Stellen zum zu synthetisierenden S/T-Netz zu addieren. Dieses Netz, das wir von nun an *zulässiges S/T-Netz* nennen wollen, hat dann offenbar folgende Eigenschaften:

- (A) Jede BPO in  $L$  ist eine partiell geordnete Ausführung des zulässigen S/T-Netzes (denn das gilt für jede einzelne Stelle dieses Netzes).

- (B) Das zulässige S/T-Netz kann zwar mehr partiell geordnete Ausführungen haben als in  $L$  angegeben, aber es gibt kein S/T-Netz mit einer kleineren Menge partiell geordneter Ausführungen, die noch  $L$  enthält (da wir *alle* zulässigen, d.h. alle möglichen, Stellen addiert haben). Das Verhalten des zulässigen S/T-Netzes ist also *minimal* mit Eigenschaft (A).

Daraus ergibt sich sofort das folgende allgemeine **Theorem**: *Entweder die Menge der partiell geordneten Ausführungen des zulässigen S/T-Netzes entspricht  $L$ , oder es gibt kein S/T-Netz, welches das Synthese-Problem löst.* Natürlich möchte man die Menge der zulässigen Stellen effektiv aus der gegebenen partiellen Sprache  $L$  berechnen können. Das ist möglich mit Hilfe einer Definition von sog. *Regionen* einer partiellen Sprache. Für Verhaltensmodelle bzgl. der sequentiellen Semantik und der Schrittsemantik wurde erfolgreich ein Konzept für Regionen entworfen [BD96], welches in [LJ06, Lor06] verallgemeinert wurde für partielle Sprachen. Dort wurde gezeigt, dass jede Region von  $L$  eine Stelle definiert, s.d. (1) *jede durch eine Region von  $L$  definierte Stelle zulässig ist*, und (2) *für jede zulässige Stelle eine Region von  $L$  existiert, die diese Stelle definiert*. Damit lässt sich das zulässige S/T-Netz direkt aus der Menge aller Regionen berechnen. Für eine konkrete algorithmische Umsetzung bleibt dabei jedoch noch ein Problem zu lösen: Die Menge aller zulässigen Stellen bzw. die Menge aller Regionen ist (im Allgemeinen) unendlich. Man benötigt also eine effektiv berechenbare Repräsentation der Menge aller Regionen durch eine endliche Menge von sog. *Basis-Regionen*. Dieses Problem wurde für Verhaltensmodelle bzgl. der sequentiellen Semantik und der Schrittsemantik in [BD96] und für *endliche* partielle Sprachen in [Lor06] gelöst. Im nächsten Paragraphen wollen wir die konkrete Definition von Regionen von partiellen Sprachen vorstellen, und skizzieren wie sich die Menge aller Regionen einer endlichen partiellen Sprache durch eine endliche Menge von Basis-Regionen repräsentieren lässt. Im darauffolgenden Paragraph wollen wir diese Theorie auf unendliche partielle Sprachen erweitern. Hierbei steht man vor dem algorithmischen Problem, dass eine unendliche partielle Sprache durch eine unendliche Menge von BPOs gegeben ist, die man erstmal durch einen endlichen Formalismus repräsentieren muss. Dafür wählen wir in dieser Arbeit einen Term-basierten Ansatz.

## 2 Formale Grundlagen: Regionen und endliche partielle Sprachen

Ein **markiertes S/T-Netz**  $N = (S, T, F, W, m_0)$  besteht aus einer Stellenmenge  $S$ , einer Transitionsmenge  $T$  ( $T \cap S = \emptyset$ ), einer Flussrelation  $F \subseteq (S \times T) \cup (T \times S)$ , einer Gewichtsfunktion  $W : (S \times T) \cup (T \times S) \rightarrow \mathbb{N}_0$  mit  $W(e) > 0 \Leftrightarrow e \in F$  und einer initialen Markierung der Stellen  $m_0 : S \rightarrow \mathbb{N}_0$ . **BPOs**  $\text{bpo} = (V, <, l)$  werden formal dargestellt durch eine Knotenmenge  $V$ , eine irreflexive und transitive Kantenmenge  $< \subseteq V \times V$  und eine Beschriftungsfunktion  $l$  auf  $V$ . Sind zwei Ereignisse  $x$  und  $y$  bzgl. einer BPO  $\text{bpo} = (V, <, l) \in L$  geordnet – d.h. gilt  $x < y$  – so bedeutet das, dass die zugehörigen Transitionen  $l(x)$  und  $l(y)$  kausal voneinander abhängig sind. Eine solche kausale Abhängigkeit entsteht genau dann, wenn die eine Transition  $l(x)$  durch ihr Schalten Marken in einer Stelle produziert, welche von der anderen Transition  $l(y)$  durch deren Schalten aus dieser Stelle konsumiert werden. Eine solche Stelle wollen wir nun definieren. Dazu spezifizieren wir für jede Kante  $(x, y)$  einer BPO in  $L$  durch eine natürliche Zahl *die Anzahl der Marken, welche von der Transition  $l(x)$  in der zu definierenden Stelle produziert werden und die von der Transition  $l(y)$  aus derselben Stelle konsumiert werden*. Auf diese Weise können wir aber noch nicht angeben,

- wieviele Marken eine Transition aus der Anfangsmarkierung der zu definierenden Stelle konsumiert (da diese Marken von keiner Transition produziert werden),

- wieviele Marken von einer Transition in der zu definierenden Stelle produziert werden, die von keiner weiteren Transition mehr konsumiert werden.

Deshalb erweitern wir jede BPO  $\text{bpo} = (V, <, l)$  zu einer BPO  $\text{bpo}^* = (V^*, <^*, l^*)$  um

- ein initiales Ereignis  $v_0$  ( $\forall v \in V : v_0 <^* v$ ), welches für eine Transition steht, die die initiale Markierung der zu definierenden Stelle erzeugt,
- ein maximales Ereignis  $v_{max}$  ( $\forall v \in V : v <^* v_{max}$ ), welches für eine Transition steht, die alle nicht von den Ereignissen der BPO konsumierten Marken aus der zu definierenden Stelle konsumiert.

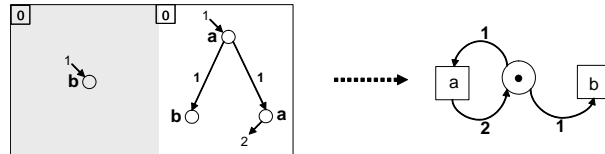


Abbildung 2: Links: Eine Region einer partiellen Sprache. Rechts: Die durch diese Region definierte Stelle.

Abbildung 2 zeigt links  $\star$ -Erweiterungen der zwei BPOs der partiellen Sprache aus Abbildung 1, wobei wir aus Platzgründen die Existenz der initialen und maximalen Ereignisse nur durch zusätzliche ein- und ausgehende Kanten angedeutet haben, wo ein positiver Markenfluss vorhanden ist. Jetzt ist es möglich, eine Stelle zu definieren, indem man in jeder BPO den Markenfluss  $r(x, y)$  bzgl. dieser Stelle zwischen kausal abhängigen Ereignissen  $x$  und  $y$  angibt. Dabei ist klar, dass gleich (mit derselben Transition) beschriftete Ereignisse insgesamt jeweils gleich viele Marken konsumieren und produzieren sollen (bzgl. der zu definierenden Stelle). Die insgesamt von einem Ereignis  $x$  einer BPO  $\text{bpo} = (V, <, l)$  produzierten Marken nennen wir den *Markenabfluss von  $x$*  (bzgl.  $\text{bpo}$ ), er entspricht der Summe  $\sum_{x <^* y} r(x, y)$  der Markenflüsse aller von  $x$  ausgehenden Kanten. Die insgesamt von einem Ereignis  $x$  konsumierten Marken nennen wir den *Markenzufluss von  $x$*  (bzgl.  $\text{bpo}$ ), er entspricht der Summe  $\sum_{y <^* x} r(y, x)$  der Markenflüsse aller in  $x$  eingehenden Kanten. Abbildung 2 zeigt links einen Markenfluss  $r$  der partiellen Sprache aus Abbildung 1, wobei die Anzahl der vom initialen Ereignis produzierten und der vom maximalen Ereignis konsumierten Marken an den Enden der zusätzlichen Kanten angeschrieben ist (existiert keine entsprechende zusätzliche Kante, so ist die zugehörige Anzahl der Marken 0). Die Zahl im Kästchen einer BPO links oben gibt die Anzahl der initialen Marken an, die nicht konsumiert werden. Offenbar haben alle mit  $a$  beschrifteten Ereignisse einen Markenabfluss von 2 und einen Markenzufluss von 1. Die mit  $b$  beschrifteten Ereignisse haben einen Markenabfluss von 0 und einen Markenzufluss von 1. Außerdem müssen alle BPOs in  $L$  partiell geordnete Ausführungen bzgl. der zu definierenden Stelle im selben Anfangszustand, d.h. in derselben Anfangsmarkierung dieser Stelle sein. Diese ist per Konstruktion gegeben durch den Markenabfluss  $\sum_{v_0 <^* y} r(v_0, y)$  des initialen Ereignisses  $v_0$  einer BPO, welchen wir auch den *initialen Markenfluss von  $\text{bpo}$*  nennen. In Abbildung 2 haben beide BPOs einen initialen Markenfluss von 1. Eine **Region einer partiellen Sprache**  $L$  ist nun als ein Markenfluss  $r : E_L \rightarrow \mathbb{N}_0$  auf der Menge Kanten  $E_L = \bigcup_{(V, <, l) \in L} <^*$  definiert, der die eben hergeleiteten Eigenschaften erfüllt:

- Gleich beschriftete Ereignisse haben gleichen Markenabfluss und Markenzufluss bzgl.  $r$ .
- Alle BPOs in  $L$  haben den gleichen initialen Markenfluss bzgl.  $r$ .

Eine **Region**  $r$  **definiert eine Stelle**  $s_r$  durch folgende Festlegungen: Die initiale Markierung von  $s_r$  ist definiert als der initiale Markenfluss der BPOs in  $L$  und die Kantengewichte der Flusskanten zwischen  $s_r$  und den Transitionen wird durch die Markenab- und zuflüsse bestimmt: Das Kantengewicht  $W(s_r, l(x))$  ist definiert als der Markenzufluss von  $x$ , das Kantengewicht  $W(l(x), s_r)$  ist definiert als der Markenabfluss von  $x$ . Damit ist der in Abbildung 2 links gezeigte Markenfluss eine Region, die wie angegeben eine Stelle im zu synthetisierenden S/T-Netz durch die zugehörigen Markenab- und zuflüsse der Ereignisse definiert. Ist  $L$  endlich, so lassen sich die Eigenschaften (i) und (ii) wie folgt durch ein System homogener linearer Gleichungen beschreiben. Sei dazu  $E_L = \{e_1, \dots, e_n\}$ . Dann kann man jede Funktion  $r : E_L \rightarrow \mathbb{N}_0$  als nicht-negativen ganzzahligen Vektor  $\mathbf{r} = (r_1, \dots, r_n)$  schreiben. Es bezeichne für Ereignisse  $x$  einer BPO  $(V, <, l) \in L$ :

- $\bullet x = \{e \in E_L \mid e = (y, x), y <^* x\}$  die Menge der in  $x$  eingehenden Kanten,
- $x^\bullet = \{e \in E_L \mid e = (x, y), x <^* y\}$  die Menge der aus  $x$  ausgehenden Kanten,
- $\text{bpo}^\bullet = \{e \in E_L \mid e = (v_0, y), v_0 <^* y\}$  die Menge der aus  $v_0$  ausgehenden Kanten.

Betrachte nun eine beliebige Funktion  $r : E_L \rightarrow \mathbb{N}$ . Dann haben gleich beschriftete Ereignisse  $x_1$  und  $x_2$  in (u.U. verschiedenen) BPOs in  $L$  genau dann den gleichen Markenabfluss bzgl.  $r$ , wenn  $\mathbf{r} \cdot \mathbf{a} = 0$  für den Vektor  $\mathbf{a} = (a_1, \dots, a_n)$  mit  $a_i = 1 \Leftrightarrow e_i \in x_1^\bullet$ ,  $a_i = -1 \Leftrightarrow e_i \in x_2^\bullet$  und  $a_i = 0$  sonst ( $\mathbf{x} \cdot \mathbf{y}$  bezeichne hierbei das Skalarprodukt von zwei Vektoren  $\mathbf{x}$  und  $\mathbf{y}$ ). Entsprechend haben  $x_1$  und  $x_2$  genau dann den gleichen Markenzufluss, wenn  $\mathbf{r} \cdot \mathbf{b} = 0$  für den Vektor  $\mathbf{b} = (b_1, \dots, b_n)$  mit  $b_i = 1 \Leftrightarrow e_i \in \bullet x_1$ ,  $b_i = -1 \Leftrightarrow e_i \in \bullet x_2$  und  $b_i = 0$  sonst. Schließlich haben verschiedene BPOs  $\text{bpo}_1$  und  $\text{bpo}_2$  in  $L$  genau dann den gleichen initialen Markenfluss, wenn  $\mathbf{r} \cdot \mathbf{c} = 0$  für den Vektor  $\mathbf{c} = (c_1, \dots, c_n)$  mit  $c_i = 1 \Leftrightarrow e_i \in \text{bpo}_1^\bullet$ ,  $c_i = -1 \Leftrightarrow e_i \in \text{bpo}_2^\bullet$  und  $c_i = 0$  sonst. Damit ist die Menge aller Regionen einer endlichen Sprache  $L$  gleich der Menge aller nicht-negativen ganzzahligen Lösungen eines endlichen homogenen linearen (Un-)Gleichungssystems  $\mathbf{A}_L \cdot \mathbf{x} = 0$ . Es ist bekannt, dass es dann eine endliche Menge von (effektiv berechenbaren) Basisvektoren, sog. *Basis-Regionen*,  $\mathbf{r}_1, \dots, \mathbf{r}_m$  gibt, s.d. jede Lösung  $\mathbf{r}$  von  $\mathbf{A}_L \cdot \mathbf{x} = 0$  darstellbar ist als eine nicht-negative Linearkombination  $\mathbf{r} = \sum_{i=1}^m \lambda_i \cdot \mathbf{r}_i$  von Basisvektoren ( $\lambda_1, \dots, \lambda_m \in \mathbb{R}^+$ ). In [Lor06] wurde gezeigt, dass das sog. *Basis-S/T-Netz*, das genau die durch Basis-Regionen definierten Stellen hat, genau dieselbe Menge von (partiell geordneten) Ausführungen hat wie das zulässige S/T-Netz. Also lässt sich das zulässige S/T-Netz für endliche partielle Sprachen durch ein effektiv berechenbares endliches S/T-Netz repräsentieren. Im folgenden Paragraph wollen wir dieses Ergebnis auf endliche Repräsentationen bestimmter Klassen unendlicher partieller Sprachen übertragen. Hierbei verzichten wir auf formale Beweise und leiten nur die jeweiligen Synthese-Algorithmen her.

### 3 Reguläre partielle Sprachen und Erweiterungen

Eine Möglichkeit, unendliche partielle Sprachen endlich darzustellen, sind reguläre Ausdrücke über einem endlichen Alphabet  $\mathcal{A}$  von BPOs. Für  $A \in \mathcal{A}$  schreiben wir dazu  $A = (V_A, <_A, l_A)$ . Außerdem sei  $\lambda = (\emptyset, \emptyset, \emptyset)$  die leere BPO. Als elementare reguläre Ausdrücke betrachten wir die Buchstaben aus  $\mathcal{A}$ , sowie die Zeichen  $\lambda$  und  $\emptyset$ . Ein regulärer Ausdruck  $\alpha$  entsteht dann durch *Hintereinanderschalten*  $\alpha = \alpha_1; \alpha_2$ , *Vereinigen*  $\alpha = \alpha_1 + \alpha_2$  oder *Iterieren*  $\alpha = (\alpha_1)^*$  anderer regulärer Ausdrücke  $\alpha_1$  und  $\alpha_2$ . Jedem regulären Ausdruck  $\alpha$  weisen wir eine Semantik in Form einer partiellen Sprache  $L(\alpha)$  zu. Dazu definieren wir zuerst die Hintereinanderschaltung von BPOs  $A, B \in \mathcal{A}$ :  $AB = (V_A \cup V_B, <_A \cup <_B \cup (V_A \times V_B), l_A \cup l_B)$  ist diejenige aus  $A$  und  $B$  zusammengesetzte BPO, in der alle Ereignisse in  $A$  vor allen Ereignissen in  $B$  eintreten müssen. Weiterhin bezeichnen wir  $A^1 = A$  und  $A^n = A^{n-1}A$  für  $n \in \mathbb{N}$ . Da die partielle Sprache der

Ausführungen eines S/T-Netzes abgeschlossen unter Sequentialisierung und Präfixbildung ist, wird auch  $L(\alpha)$  als abgeschlossen unter Sequentialisierung und Präfixbildung definiert. Dazu definieren wir  $L(\lambda) = \{\lambda\}$ ,  $L(\emptyset) = \emptyset$ , sowie  $L(A)$  als die Menge aller Sequentialisierungen von Präfixen von  $A$  für  $A \in \mathcal{A}$ , und weiter induktiv für reguläre Ausdrücke  $\alpha_1$  und  $\alpha_2$ :

- $L(\alpha_1 + \alpha_2) = L(\alpha_1) \cup L(\alpha_2)$ ,
- $L(\alpha_1; \alpha_2) = L(\alpha_1)L(\alpha_2) = \bigcup_{A_1 \in L(\alpha_1), A_2 \in L(\alpha_2)} L(A_1 A_2)$ ,
- $L((\alpha_1)^*) = L(\alpha_1)^* = \{\lambda\} \cup \bigcup_{A_1, \dots, A_n \in L(\alpha_1), n \in \mathbb{N}} L(A_1 \dots A_n)$ .

Nur durch die Iteration lassen sich hierbei *unendliche* partielle Sprachen spezifizieren. Eine BPO  $A$  kann dabei genau dann beliebig oft ausgehend von einer bestimmten Markierung hintereinander – also iteriert – schalten, wenn sie in jeder Stelle höchstens gleich viele Marken konsumiert wie produziert (damit kann ein Schalten von  $A$  die Anzahl der Marken in keiner Stelle verringern). Definiert man also eine Stelle durch einen Markenfluss  $\mathbf{r}$  auf den Kanten von  $A^*$ , so kann  $A$  bzgl. dieser Stelle genau dann iteriert schalten, wenn der zugehörige initiale Markenfluss von  $A$  höchstens gleich dem zugehörigen sog. *finalen Markenfluss* von  $A$  ist. Der finale Markenfluss einer BPO ist hierbei gegeben als die Anzahl nicht konsumierter Marken, also als der Markenzufluss des maximalen Ereignisses der BPO. Dies lässt sich offenbar durch eine lineare homogene Ungleichung der Form  $\mathbf{r} \cdot \mathbf{d} \geq \mathbf{0}$  ausdrücken für den Vektor  $\mathbf{d} = (d_1, \dots, d_n)$  mit  $d_i = 1 \Leftrightarrow e_i \in \bullet A$ ,  $d_i = -1 \Leftrightarrow e_i \in A \bullet$  und  $d_i = 0$  sonst, wobei  $\bullet A = \{e \in \langle A \rangle \mid e = (y, v_{max})\}$ . Mehrmals iterierte BPOs  $A^n$  mit  $n \geq 2$  müssen also nicht zur Berechnung zulässiger Stellen herangezogen werden. Schaltet man wie in  $A^*B$  eine andere BPO  $B$  hinter eine iterierte BPO  $A$ , so muss  $B$  nach beliebig häufigem und insbesondere auch keinmaligem Schalten von  $A$  schalten können. Es ist dabei unmittelbar klar, dass  $B$  nach jedem  $A^n$  für  $n \geq 2$  schalten kann, wenn  $B$  nach  $A$  schalten kann (da ein Schalten von  $A$  die Anzahl der Marken in keiner Stelle verringern kann). Man muss also Stellen durch Regionen nur so definieren, dass (i)  $A$  iteriert schalten kann, und (ii)  $B$  nach ein- und keinmaligem Schalten von  $A$  schalten kann. Eigenschaft (i) stellt man wie oben skizziert durch lineare Ungleichungen über der Kantenmenge von iterierten BPOs sicher, wozu man induktiv zu einem regulären Ausdruck  $\alpha$  die Menge  $I(\alpha)$  der in ihm iterierten BPOs berechnet. Eigenschaft (ii) stellt man durch lineare Ungleichungen über der Kantenmenge einer endlichen Menge  $R(\alpha)$  von Repräsentanten von  $L(\alpha)$  sicher, welche iterierte BPOs in allen Kombinationen ein- bzw. einmal enthält.

- $R(\lambda) = \{\lambda\}$  und  $R(\emptyset) = I(\emptyset) = I(\lambda) = \emptyset$ ,  $R(A) = \{A\}$  und  $I(A) = \emptyset$ ,
- $R(\alpha_1 + \alpha_2) = R(\alpha_1) \cup R(\alpha_2)$  und  $I(\alpha_1 + \alpha_2) = I(\alpha_1) \cup I(\alpha_2)$ ,
- $R(\alpha_1; \alpha_2) = \{A_1 A_2 \mid A_1 \in R(\alpha_1), A_2 \in R(\alpha_2)\}$  und  $I(\alpha_1; \alpha_2) = I(\alpha_1) \cup I(\alpha_2)$ ,
- $R((\alpha_1)^*) = R(\alpha_1) \cup \{\lambda\}$  und  $I((\alpha_1)^*) = I(\alpha_1) \cup R(\alpha_1)$ .

Es gilt  $R(A^*B) = \{B, AB\}$ ,  $I(A^*B) = \{A\}$ ,  $R((A^*B)^*) = \{\lambda, B, AB\}$  und  $I((A^*B)^*) = \{A, B, AB\}$ . Eine *Region eines regulären Ausdrucks*  $\alpha$  ist nun ein Markenfluss auf den Kanten der BPOs in  $R(\alpha)$  und in  $I(\alpha)$ , der folgende Eigenschaften erfüllt:

- (i) In allen BPOs in  $R(\alpha)$  haben gleich beschriftete Ereignisse gleichen Markenabfluss und -zufluss und alle BPOs in  $R(\alpha)$  haben gleichen initialen Markenfluss.
- (ii) Alle BPOs in  $I(\alpha)$  haben größeren finalen als initialen Markenfluss.
- (iii) Die Markenab- und -zuflüsse von gleich beschrifteten Ereignissen von BPOs in  $R(\alpha)$  und  $I(\alpha)$  stimmen überein.

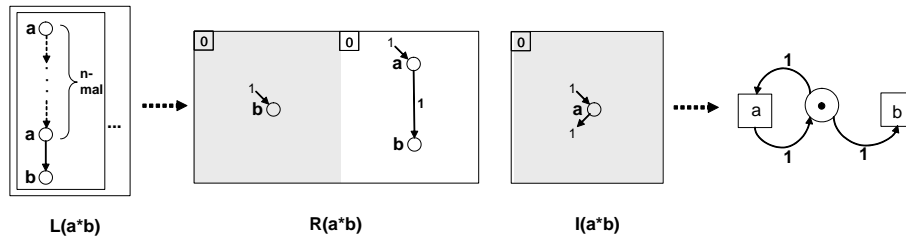


Abbildung 3: Eine durch einen regulären Ausdruck spezifizierte partielle Sprache zusammen mit an die Kanten ihrer Repräsentanten und ihrer iterierten BPOs annotierter Region.

Diese Eigenschaften stellt man leicht durch endlich viele lineare homogene Ungleichungen sicher. Es bleibt formal zu beweisen, dass durch die Menge der Lösungen dieses homogenen linearen Ungleichungssystems genau die Menge der zulässigen Stellen definiert wird. Abbildung 3 zeigt die Mengen  $R(A^*B) = \{B, AB\}$  und  $I(A^*B) = \{A\}$  für die BPOs  $A = (\{a\}, \emptyset, id)$  und  $B = (\{b\}, \emptyset, id)$ , zusammen mit einem an die Kanten annotierten Markenfluss, der diese Eigenschaften erfüllt – und somit eine Region ist, die die rechts gezeigte Stelle definiert.

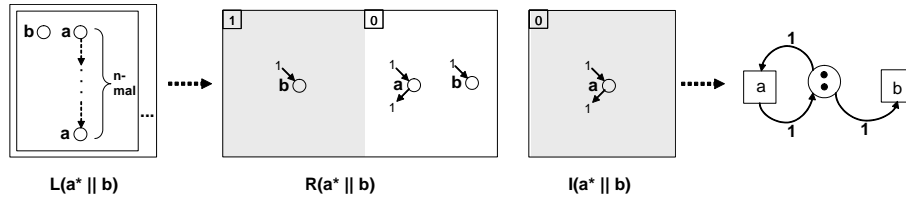


Abbildung 4: Eine durch einen regulären Ausdruck mit  $\parallel$ -Operator spezifizierte partielle Sprache zusammen mit an die Kanten ihrer Repräsentanten und ihrer iterierten BPOs annotierter Region.

Die Klasse partieller Sprachen, die durch reguläre Ausdrücke spezifizierbar sind, ist noch relativ beschränkt. Abbildung 4 zeigt eine partielle Sprache, die man nicht als regulären Ausdruck angeben kann – sie lässt sich aber durch Hinzunahme eines Operators  $\parallel$  zur parallelen Komposition regulärer Ausdrücke durch den Ausdruck  $A^* \parallel B$  repräsentieren. Formal definieren wir für BPOs  $A, B$  und solchermaßen erweiterte reguläre Ausdrücke  $\alpha_1$  und  $\alpha_2$ :

- $A \parallel B = (V_A \cup V_B, <_A \cup <_B, l_A \cup l_B)$ ,  $L(\alpha_1 \parallel \alpha_2) = \bigcup_{A_1 \in L(\alpha_1), A_2 \in L(\alpha_2)} L(A_1 \parallel A_2)$ ,
- $R(\alpha_1 \parallel \alpha_2) = \{A_1 \parallel A_2 \mid A_1 \in R(\alpha_1), A_2 \in R(\alpha_2)\}$ ,  $I(\alpha_1 \parallel \alpha_2) = I(\alpha_1) \cup I(\alpha_2)$ .

Für den erweiterten regulären Ausdruck  $A^* \parallel B$  gilt dann  $R(A^* \parallel B) = \{B, A \parallel B\}$  und  $I(A^* \parallel B) = \{A\}$ . Regionen für solche Ausdrücke sind analog zu Regionen von regulären Ausdrücken durch obige Eigenschaften (i) - (iii) definiert, so dass man also wieder die Möglichkeit hat, Regionen als Lösungen eines endlichen linearen homogenen Ungleichungssystems zu schreiben (Abbildung 4 zeigt eine Region von  $A^* \parallel B$  mit zugehöriger Stelle). Auch mit solchen erweiterten regulären Ausdrücken lässt sich ein wesentliches Prinzip von Ausführungen von S/T-Netzen noch nicht spezifizieren: Während bei der Hintereinanderschaltung und der Iteration von BPOs immer *alle* Ereignisse einer BPO hinter *allen* Ereignissen einer anderen BPO eintreten müssen, können in Ausführungen von S/T-Netzen Ereignisse einer BPO auch nur hinter einer Teilmenge der Ereignisse einer anderen BPO eintreten. Abbildung 5 zeigt ein solches S/T-Netz: Hier ist nach jedem Eintreten von  $A$  die Transitionen  $A$  und  $B$  parallel aktiviert. Um auf Term-Ebene ein solches Verhalten auszudrücken, sind geeignet definierte Operatoren zur *lokalen* Hintereinanderschaltung und *lokalen* Iteration notwendig. Hier ist jedoch noch unklar, inwieweit

es sinnvoll und möglich ist, solche Operatoren zur Komposition beliebiger erweiterter regulärer Ausdrücke und nicht nur zur Komposition von einzelnen BPOs zu definieren.

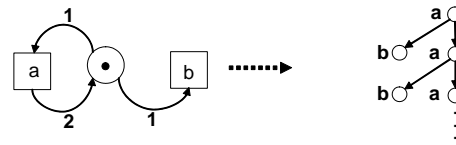


Abbildung 5: Ein S/T-Netz und eine Menge von Ausführungen dieses S/T-Netzes, die sich nicht durch einen erweiterten regulären Ausdruck spezifizieren lässt.

## 4 Ausblick: Effiziente Algorithmen und BPO-Netze

Die vorgestellten Synthese-Methoden sind im Allgemeinen noch nicht effizient, da die Anzahl der Basis-Lösungen eines linearen homogenen Ungleichungssystems exponentiell in der Anzahl der Ungleichungen sein kann. Aus zwei Gründen gehen wir aber davon aus, dass es möglich sein wird, aus den vorgestellten Methoden effiziente Algorithmen abzuleiten: Erstens wurde bis jetzt nicht berücksichtigt, dass mehrere verschiedene Basis-Regionen dieselbe Stelle definieren können (namentlich falls sie nur eine unterschiedliche Markenfluss-Verteilung bzgl. derselben Stelle repräsentieren), zweitens wurde bis dato nicht benutzt, dass manche Basis-Regionen (-Stellen) das Verhalten mehr einschränken als andere Basis-Regionen (-Stellen). Es würde also grundsätzlich genügen, für jede Basis-Stelle nur eine Basis-Region zu berechnen und dabei nur sog. *minimale* Basis-Regionen (-Stellen) zu betrachten, welche das Verhalten des synthetisierenden Netzes unter allen Basis-Regionen maximal einschränken. Ein weiterer Ansatz zur Effizienz-Steigerung wäre, zuerst die Buchstaben aus  $A$  in einem (erweiterten) regulären Ausdruck als einzelne Transitionen/Ereignisse anzusehen und bekannte effiziente Synthese-Algorithmen für die sequentielle Semantik von S/T-Netzen anzuwenden. Damit könnte man effizient Stellen berechnen, die den "Kontrollfluss" zwischen den BPOs eines (erweiterten) regulären Ausdrucks beschreiben. Man erhält so ein (endliches) S/T-Netz, dessen Transitionen BPOs repräsentieren (solche Netze wollen wir *BPO-Netze* nennen). Im nächsten Schritt würde man dann für jede einzelne BPO des regulären Ausdrucks (des BPO-Netzes) die zusätzlich notwendigen Stellen zur Sicherstellung des durch die BPO gegebenen Verhaltens berechnen mit Hilfe der vorgestellten Methoden für endliche partielle Sprachen. Hier ist allerdings noch unklar, inwieweit Regionen verschiedener BPOs miteinander kompatibel sind, da verschiedene BPOs auch gleiche Transitionsbeschriftungen enthalten können. Schließlich könnte man (als nicht Term-basierter Ansatz) auch BPO-Netze selbst zur endlichen Beschreibung unendlicher partieller Sprachen verwenden.

## Literatur

- [BD96] BADOUEL, E. ; DARONDEAU, P.: On the Synthesis of General Petri Nets. / Inria. 1996 (3025). – Forschungsbericht
- [LJ06] LORENZ, R. ; JUHÁS, G.: Towards Synthesis of Petri Nets from Scenarios. In: DONATELLI, S. (Hrsg.) ; THIAGARAJAN, P. S. (Hrsg.): *ICATPN* Bd. 4024, Springer, 2006 (Lecture Notes in Computer Science), S. 302–321
- [Lor06] LORENZ, R.: *Szenario-basierte Verifikation und Synthese von Perinetzen: Theorie und Anwendungen*. Habilitation, 2006