

# NP Zusammenfassung

Lukas Schaller

April 25, 2019

# Contents

<b>1</b>	<b>A1</b>	<b>2</b>
1.1	Aktionen . . . . .	2
1.2	Trainingsblatt . . . . .	3
1.3	Nichtdeterminismus . . . . .	3
1.3.1	Annahmen nebenläufiger Prozesse . . . . .	3
1.4	Labled Transitionsystem . . . . .	4
<b>2</b>	<b>B1 - CSS</b>	<b>5</b>
2.1	$CSS_0$ . . . . .	5
2.1.1	Syntax . . . . .	5
2.1.2	Semantik . . . . .	5
2.2	$CCS_0^\omega$ . . . . .	6
2.2.1	Semantik . . . . .	6
2.2.2	Geschützte Ausdrücke . . . . .	6
2.3	Sequentielles $CSS_0^\omega$ . . . . .	6
2.3.1	Semantik . . . . .	6
2.3.2	Synchronisation . . . . .	7
2.3.3	Restriktions-Operator . . . . .	7
2.4	Volle CCS-Power . . . . .	7
2.4.1	Syntax . . . . .	7
2.4.2	Semantik . . . . .	8
2.4.3	Regulärer Ausdruck . . . . .	8

# Chapter 1

## A1

### 1.1 Aktionen

**Aktionen** Seien  $Kom$  eine Menge von Kommunikationsaktionen und  $Int$  eine davon disjunkte Menge von internen Aktionen. Dann ist

$$Act = Kom \cup Int$$

die Menge aller Aktionen. Dabei gelten folgende Konventionen:

$$\alpha, \beta, \gamma, \dots \in Act \quad a, b, c, \dots \in Int \quad [a], [b], [c], \dots \in Kom$$

**Transitionssystem (LTS)** Ein beschrites Transitionssystem TS ist ein Tripel  $(S, \longrightarrow, s_0)$ , wobei

- $S$  die Zustandsmenge
- $\longrightarrow \subseteq S \times Act \times S$  die Transitionsrelation,
- $s_0 \in S$  der initiale Zustand ist

**Nachfolger** Sei ein LTS  $TS = (S, \rightarrow, s_0)$  gegeben. Sei  $s \in S$ ,  $C \subseteq S$ ,  $\alpha \in Act$ , und  $A \subseteq Act$ .

$$Post(s, \alpha) = \{s' \in S \mid s \xrightarrow{\alpha} s'\},$$

$$Post(s, A) = \bigcup_{\alpha \in A} Post(s, \alpha)$$

$$Post(C, \alpha) = \bigcup_{s \in C} Post(s, \alpha),$$

$$Post(C, A) = \bigcup_{\alpha \in A} Post(C, \alpha)$$

Aktionen die in Zustand  $s$  als nächstes möglich sind:

$$Act(s) = \{\alpha \in Act \mid \exists s' : s \xrightarrow{\alpha} s'\}$$

Aktionen, die in Zustand  $s$  als nächstes beobachtet werden können:

$$Kom(s) = \{\alpha \in Kom \mid \exists s' : s \xrightarrow{\alpha} s'\}$$

**Erreichbarkeit**  $Reach(s)$  ist die Menge aller von  $s$  erreichbaren Zustände in  $TS$

$$Reach(s) = \bigcup_{n \in \mathbb{N}} Post^n(s)$$

wobei  $Post^0(s) = s$  und  $Post^{n+1}(s) = Post(Post^n(s), Act)$

## 1.2 Trainingsblatt

### Vorgänger

$$Pre(s, \alpha) = \{s' \in S \mid s' \xrightarrow{\alpha} s\},$$

$$Pre(s, A) = \bigcup_{\alpha \in A} Pre(s, \alpha)$$

$$Pre(C, \alpha) = \bigcup_{s \in C} Pre(s, \alpha),$$

$$Pre(C, A) = \bigcup_{\alpha \in A} Pre(C, \alpha)$$

**terminaler Zustand** Ein terminaler Zustand ist ein Zustand ohne Nachfolger. Ein Zustand ist genau dann terminal wenn  $Act(s) = \emptyset$ .

**Alphabet des LTS** Alphabet eines LTS  $TS = \bigcup_{s \in Reach(TS)} Kom(s)$

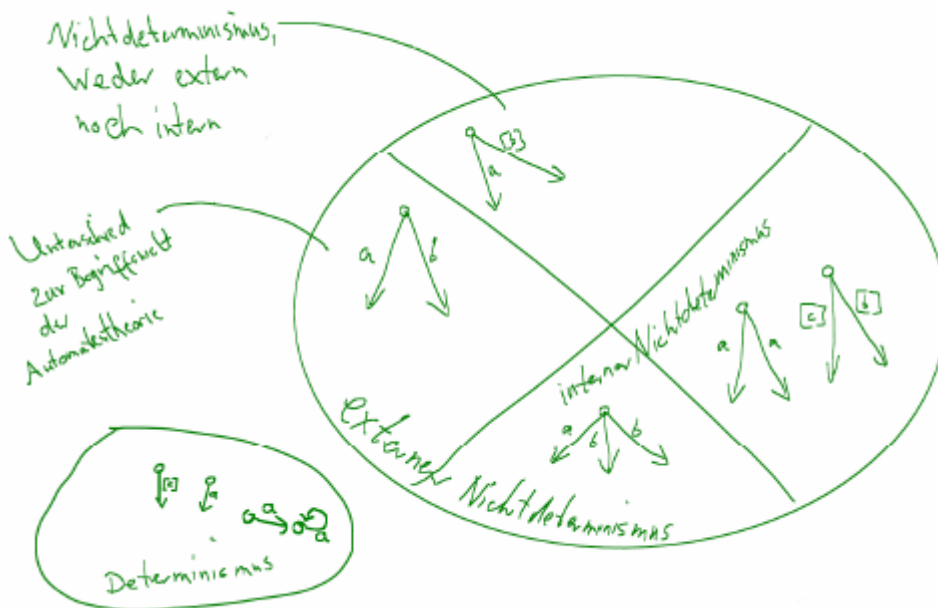
## 1.3 Nichtdeterminismus

**Nichtdeterminismus** Sei  $Post(s) = Post(s, Act)$  und es sei ein LTS  $TS = (S, \rightarrow, s_0)$  gegeben.  $TS$  ist deterministisch genau dann wenn für alle  $s \in S$ ,

$$|Post(s)| \leq 1 \text{ und } |Act(s)| \leq 1$$

Andernfalls heißt das  $TS$  nichtdeterministisch!

### Interner und Externer Nichtdeterminismus

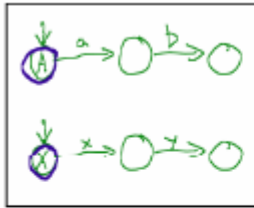


### 1.3.1 Annahmen nebenläufiger Prozesse

- Zeit wird nur bezüglich relativen Geschwindigkeit der Prozesse untereinander betrachtet, nicht absolut. Jeder Prozess kann gerade beliebig schnell oder langsam voranschreiten.
- Aktionen sind unteilbar und zeitlos.
- Nebenläufige Prozesse agieren komplett voneinander unabhängig und unbeeinflusst, es sei denn, sie kommunizieren explizit miteinander.

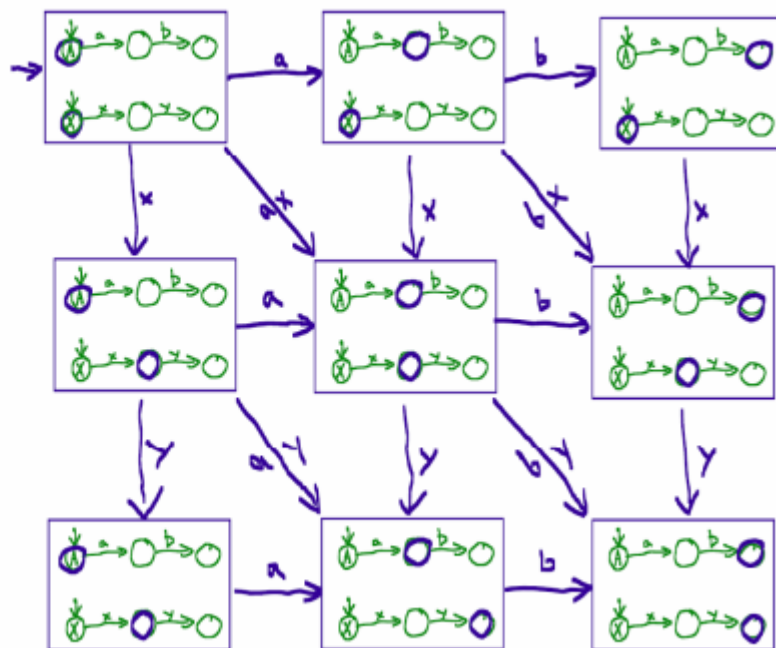
## 1.4 Labeled Transitionsystem

Ein kleines Beispiel:



Das ist ein  
Transitionssystem!

Allerdings mit  
Mengen von  
Aktionen  
beschriftet.



**Hinweis:** Die diagonalen Kanten können wegen der Zeitlosigkeit der Aktionen auch weggelassen werden.

# Chapter 2

## B1 - CSS

### 2.1 $CSS_0$

#### 2.1.1 Syntax

Gegeben sei die Menge aller Aktionen  $Act$ . Dann ist die Menge aller Ausdrücke in  $CSS_0$  gegeben durch:

$$P ::= 0 \mid P + P \mid \alpha.P$$

wobei  $\alpha \in Act$ .

#### Klammersparregeln

- '+' klammert links:  $P + Q + R \rightsquigarrow (P + Q) + R$
- '.' klammert rechts:  $\alpha.\beta.P \rightsquigarrow \alpha.(\beta.P)$
- Punkt vor Strich:  $\alpha.P + Q \rightsquigarrow (\alpha.P) + Q$

#### 2.1.2 Semantik

Die Semantik einer Sprache beschreibt, welches mathematische Objekt mit einem Ausdruck der Sprache assoziiert werden soll. Die Semantik des Ausdrucks  $P$  aus  $CSS_0$  ist ein LTS  $\llbracket P \rrbracket = (S, \rightarrow, s_0)$ .

- Zustandsmenge  $S$  ist die Menge aller  $CSS_0$ -Ausdrücke
- $s_0 = P$
- $\rightarrow$  ist eine Teilmenge von  $(CSS_0 \times Act \times CSS_0)$

#### Inferenzregeln

Hinweis:  $CSS_0$  beinhaltet noch nicht die Inferenzregeln  $rec$

$\text{choice\_l} \quad \frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'}$	$\text{prefix} \quad \frac{}{a.P \xrightarrow{\alpha} P}$
$\text{choice\_r} \quad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$	$\text{rec} \quad \frac{\Gamma(X)=P \quad P \xrightarrow{\alpha} P'}{X \xrightarrow{\alpha} P'}$

#### Isomorphie

Zwei Transitionssysteme  $TS = (S, \rightarrow, s_0)$  und  $TS' = (S', \rightarrow', s'_0)$  sind isomorph ( $TS \sim_{iso} TS'$ ), wenn es eine Bijektion  $f$  gibt mit  $f : Reach(TS) \rightarrow Reach(TS')$ , so dass  $f(s_0) = s'_0$  und für alle  $s_1, s_2 \in Reach(TS)$  und alle  $\alpha \in Act$  gilt:  $s_1 \xrightarrow{\alpha} s_2$  genau dann wenn  $f(s_1) \xrightarrow{\alpha'} f(s_2)$ .

**Satz** Für jedes endliche azyklische LTS  $TS$  gibt es einen Ausdruck  $P \in CSS_0$ , so dass  $TS \sim_{iso} \llbracket P \rrbracket$ .

## 2.2 $CCS_0^\omega$

### 2.2.1 Semantik

Gegeben sei die Menge aller Aktionen  $Act$  und eine Menge von Rekursionsvariablen  $Var$ . Dann ist die Menge der Ausdrücke in  $CCS_0^\omega$  wie folgt:

$$P ::= o \mid X \mid P + P \mid \alpha.P, \text{ wobei } \alpha \in Act \text{ und } X \in Var$$

#### Beispiele

$$\begin{aligned} X &:= a.b.Y \\ Y &:= b.Z + a.Y \\ Z &:= a.Y \end{aligned}$$

Eine Menge solcher Gleichungen definiert offensichtlich eine partielle Funktion:  $\Gamma \in Var \rightarrow CCS_0^\omega$ . In diesem Beispiel ergibt sich demnach:  $\Gamma = \{(X, a.b.Y), (Y, b.Z + a.Y), (Z, a.Y)\}$

Es soll gelten:

- $(\alpha.P, \alpha, P) \in \rightarrow$ ;
- $(P + Q, \alpha, P') \in \rightarrow$ , wenn immer  $(P, \alpha, P') \in \rightarrow$ ;
- $(P + Q, \alpha, Q') \in \rightarrow$ , wenn immer  $(Q, \alpha, Q') \in \rightarrow$ ;
- $(X, \alpha, P') \in \rightarrow$ , wenn immer  $\Gamma(x) = P$  und  $(P, \alpha, P') \in \rightarrow$ ;
- nichts sonst ist Element von  $\rightarrow_\Gamma$

### 2.2.2 Geschützte Ausdrücke

Eine Variable  $X$  ist geschützt in einem Ausdruck  $P$ , wenn jedes Auftreten von  $X$  in  $P$  in einem Teilausdruck von  $P$  der Form  $\alpha.Q$  enthalten ist. Andernfalls heißt  $X$  ungeschützt.

Ein Ausdruck  $P$  heißt geschützt, wenn alle darin vorgekommenen Variablen in  $P$  geschützt sind. Andernfalls heißt  $P$  ungeschützt.

#### Beispiele

ungeschützte Ausdrücke:  $X, \tau.X + Y, (a.X) + X$       geschützte Ausdrücke:  $\alpha.X, \tau.(X + Y), \alpha.(X + b.X)$

## 2.3 Sequentielles $CCS_0^\omega$

Für eine Bindung  $\Gamma : Var \rightarrow CCS_0^\omega$  sei  $\rightarrow_\Gamma \subseteq CCS_0^\Gamma \times Act \times CCS_0^\Gamma$  die kleinste Relation  $\rightarrow$ , die den Inferenzregeln genügen.

### 2.3.1 Semantik

Sei  $LTS_0^\omega = \{(CCS_0^\omega, T, s) \mid T \subseteq CCS_0^\omega \times Act \times CCS_0^\omega, s \in CCS_0^\omega\}$ . Die Semantik von  $CCS_0^\omega$  ist eine (kaskadierte) Funktion:

$$\begin{aligned} \llbracket - \rrbracket &: (Var \rightarrow CCS_0^\omega) \rightarrow CCS_0^\omega \rightarrow LTS_0^\omega \\ \llbracket - \rrbracket \Gamma P &= (CCS_0^\omega, \rightarrow_\Gamma, P) \end{aligned}$$

Wir schreiben  $\llbracket P \rrbracket_\Gamma$  für  $\llbracket - \rrbracket \Gamma P$ , oder auch nur  $\llbracket P \rrbracket$ , sofern  $\Gamma$  aus dem Kontext heraus klar ist.

**Satz** Für jedes endliche LTS  $TS$  gibt es einen Ausdruck  $P \in CCS_0^\omega$  und eine endliche Bindung  $\Gamma$ , sodass  $TS \sim_{iso} \llbracket P \rrbracket_\Gamma$

### Zusätzliche Inferenzregeln

TODO: Hier Bild von `par.l` und `par.r` einfügen

### 2.3.2 Synchronisation

Synchronisation bietet die Grundlage um unterschiedliche Prozesse auf Aktionen reagieren zu lassen. So kann zum Beispiel die 'light'-Aktion des Feuerzeug-Prozesses in einem Prozess eines Chinaböllers die passive Aktion 'light' hervorrufen. In CSS machen wir dazu die Kommunikationsaktionen entweder 'aktiv' oder 'passiv'. Statt einer beliebigen Menge  $Kom$  von Markierungen verwenden wir eine Menge, die mehr Struktur aufweist:  $Kom = A^! \cup A^?$ . Aktionen mit '!' als Output-Aktionen (aktiv) und Aktionen mit '?' als Input-Aktionen interpretiert werden.

**Komplementarität** Input- und Output-Aktionen treten als Paare auf. Das Komplement von  $a \in A^! \cup A^?$  ist  $\bar{a}$ . Auch für interne Aktionen  $\tau$  gilt  $\tau = \bar{\tau}$ . Das doppelte Komplement hebt die Wirkung auf:  $\alpha \in Act : \bar{\bar{\alpha}} = \alpha$

**Hinweis** Interne Aktionen können NICHT synchronisieren.

### Zusätzliche Inferenzregel

$$\text{sync} \quad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

### 2.3.3 Restriktions-Operator

Der Restriktions-Operator 'verbindet bestimmte (Paare von) Aktionen eines Prozesses. Intuitiv gesprochen erzwingt er eine Synchronisation.

$P \setminus H$  ist ein zweistelliger Operator, mit  $P$  als CCS-Ausdruck und  $H$  als Menge von Kommunikationsaktionen, die unterbunden werden sollen.

### Inferenzregel

$$\text{res} \quad \frac{P \xrightarrow{\alpha} P' \quad \alpha \notin H}{P \setminus H \xrightarrow{\alpha} P' \setminus H}$$

Annahmen über zulässige Menge  $H$ :

- Die interne Aktion kann nicht unterbunden werden:  $\tau \notin H$
- Aktionen treten in  $H$  paarweise auf:  $a \in H \iff \bar{a} \in H$

## 2.4 Volle CCS-Power

### 2.4.1 Syntax

Gegeben sei die Menge aller Kommunikationsaktionen  $Kom = A^! \cup A^?$ ,  $Act = Kom \cup \{\tau\}$  und eine Menge von Rekursionsvariablen  $Var$ . Dann ist die Menge aller Ausdrücke in  $CCS$  wie folgt gegeben:

$$P ::= 0 \mid X \mid P + P \mid \alpha.P \mid P|P \mid P \setminus H$$

wobei  $\alpha \in Act$ ,  $X \in Var$  und  $H \subseteq Kom$ .



## 2.4.2 Semantik

Die Semantik der Ausdrücke von CCS ist gegeben durch:

$$\llbracket \_ \rrbracket : (Var \rightarrow CCS) \rightarrow CCS \rightarrow LTS^{CCS}$$

$$\llbracket \_ \rrbracket \Gamma P = (CCS, \rightarrow_\Gamma, P)$$

wobei  $LTS^{CCS} = \{(CCS, T, s) | T \subseteq CCS \times Act \times CCS, s \in CCS\}$  und  $\rightarrow_\tau$  die kleinste Relation  $\rightarrow$  ist, die den folgenden Regeln genügt.

### Inferenzregeln

$$\begin{array}{c} \text{prefix} \frac{}{a.P \xrightarrow{\alpha} P} \quad \text{choice\_l} \frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'} \quad \text{choice\_r} \frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'} \quad \text{rec} \frac{\Gamma(X)=P \quad P \xrightarrow{\alpha} P'}{X \xrightarrow{\alpha} P'} \\ \text{par\_l} \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \quad \text{sync} \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \quad \text{par\_r} \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \quad \text{res} \frac{P \xrightarrow{\alpha} P' \quad \alpha \notin H}{P \setminus H \xrightarrow{\alpha} P' \setminus H} \end{array}$$

## 2.4.3 Regulärer Ausdruck

Ein CCS-Asdruck wird regulär genannt, wenn er durch folgende Grammatik gebildet werden kann:

$$\begin{aligned} P &::= 0 \mid X \mid P + P \mid \alpha.P \mid R \\ R &::= 0 \mid R + R \mid \alpha.R \mid R|R \mid R \setminus H \end{aligned}$$

**Satz** Sofern  $\Gamma$  eine endliche Bindung ist bei der alle rechten Seiten regulär sind, ist  $Reach(\llbracket P \rrbracket_\Gamma)$  für jedes  $P \in CCS$  endlich.