Baumautomaten

Präsentiert von: Lars Nitzke
Proseminar am Lehrstuhl für Softwaretechnik

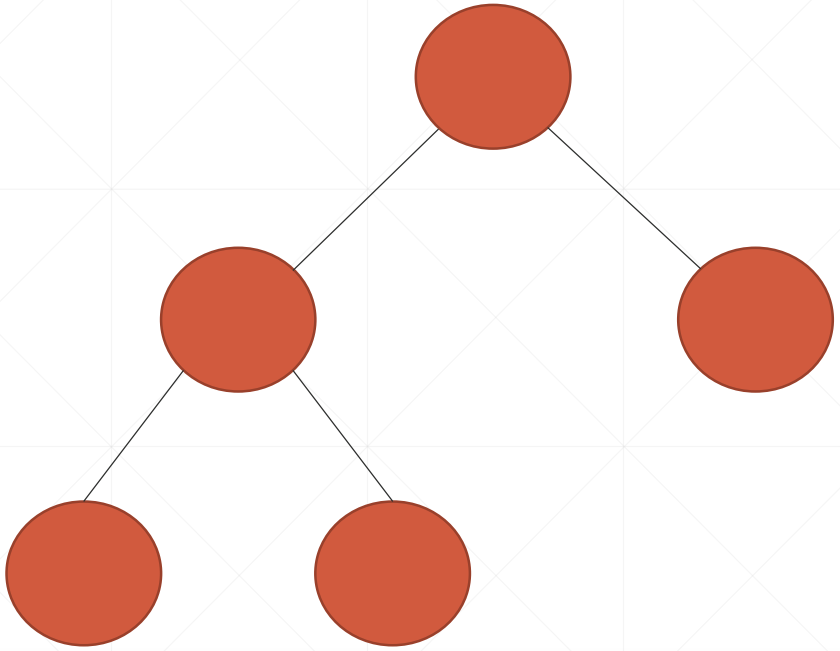
Gliederung

- Bäume
- Baumsprachen
- Baumautomaten

Bäume

Was sind Bäume?

- Datenstrukturen
- Können verschiedene Dinge darstellen
  - Verzeichnisse
  - Terme
  - Usw.



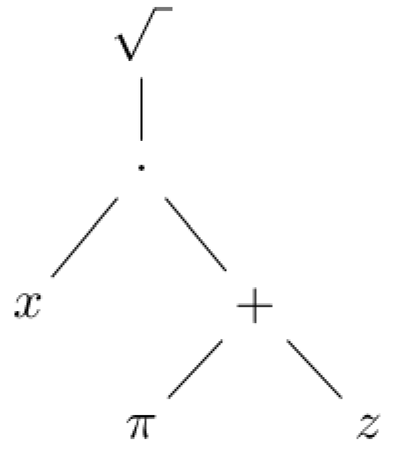
Dabei sind  die Knoten oder innere Knoten und die Übergänge die Kanten.
Knoten die keine ausgehenden Kanten haben, nennt man Blätter.

Beispiele

XML-Dokument

```
<Something>
    <SomethingElse>
        ...
    </SomethingElse>
    <Name> Max Mustermann </Name>
</Something>
```

Der Term $\sqrt{x \cdot (\pi + z)}$ als Baum



Definition Rangalphabet

Ein Rangalphabet ist eine nichtleere endliche Menge Σ von Symbolen.

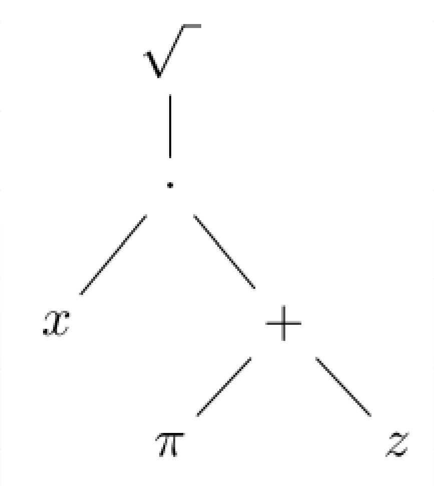
$$\Sigma_i := \{a \mid a \in \Sigma\}$$
$$\Sigma = \Sigma_0 \cup \cdots \cup \Sigma_n$$

Dabei müssen Σ_i nicht disjunkt sein.

Rangalphabet

Das Rangalphabet des Terms $\sqrt{x \cdot (\pi + z)}$ besteht aus $\Sigma = \{\sqrt{\ }, +, \cdot, x, \pi, z\}$ mit

- $\Sigma_2 = \{+, \cdot\}$
- $\Sigma_1 = \{\sqrt{\ }\}$
- $\Sigma_0 = \{x, \pi, z\}$



Definition Baumsprache

Sei $\Sigma_0 \cup \cdots \cup \Sigma_n$ ein Rangalphabet. Die Menge T_Σ von Bäumen über Σ ist induktiv definiert durch:

- Jedes Symbol $a \in \Sigma_0$ ist ein Baum $a \in T_\Sigma$
- Für $f \in \Sigma_k$ und $t_1 \ldots t_k \in T_\Sigma$, ist $f(t_1 \ldots t_k)$ ein Baum aus T_Σ, der wie folgt dargestellt wird:

Eine Menge $T \subseteq T_\Sigma$ nennt man Baumsprache

Baumautomaten

Es gibt die

- **deterministischen Baumautomaten** (DTA) und
- **nichtdeterministischen Baumautomaten** (NTA)

Diese Automaten lesen einen Baum von den Blättern aus zur Wurzel hingehend. Außerdem sind sie gleichmächtig.

Im Vergleich dazu, gibt es noch die sogenannten

- **deterministischen top-down Automaten** (↓DTA) und
- **nichtdeterministischen top-down Baumautomaten** (↓NTA),

die einen Baum von der Wurzel aus zu den Blättern hingehend lesen. Während ein ↓NTA gleichmächtig zu einem DTA ist, ist ein ↓DTA schwächer als ein ↓NTA.

Definition DTA

Ein deterministischer Baumautomat ist ein Tupel $\mathcal{A} = (Q, \Sigma, \delta, F)$ mit

- Q eine endliche Menge von Zuständen,
- Σ ein Rangalphabet,
- $F \subseteq Q$ eine endliche Menge von akzeptierenden Zuständen,
- $\delta\colon \bigcup_{i \geq 0} (Q^i \times \Sigma_i) \to Q$ eine Übergangsfunktion.

Dabei ist $Q^0 \times \Sigma_0 \coloneqq \Sigma_0$.
Die erweiterte Übergansfunktion $\delta^*\colon T_\Sigma \to Q$ ist induktiv definiert:

$$\delta^*(a) \coloneqq \delta(a) \qquad a \in \Sigma_0$$

$$\delta^*\big(f(t_1 \dots t_i)\big) \coloneqq \delta(\delta^*(t_1), \dots, \delta^*(t_i), f) \qquad f \in \Sigma_i \text{ und } t_1, \dots, t_i \in \mathrm{T}_\Sigma$$

Ein DTA $\mathcal{A}$ akzeptiert einen Baum $t \in T_\Sigma$, wenn $\delta^*(t) \in F$.
Die Baumsprache die von $\mathcal{A}$ erkannt wird, heißt $T(\mathcal{A}) = \{t \in T_\Sigma \mid \mathcal{A} \text{ akzeptiert t}\}$

T nennt man regulär, falls es einen DTA $\mathcal{A}$ gibt, mit $T = T(A)$

Beispiel DTA

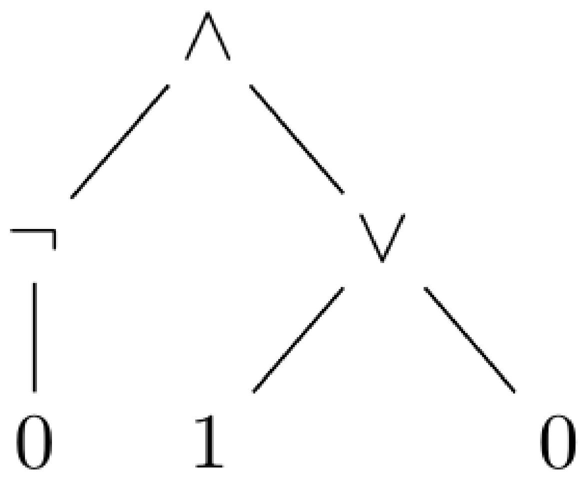
Wir definieren uns den Baumautomaten, der alle wahren Ausdrücke über $\Sigma = \{\wedge, \vee, \neg, 0, 1\}$ akzeptiert. Sei $Q = \{q_0, q_1\}$ und $F = \{q_1\}$.

δ ist wie folgt definiert:

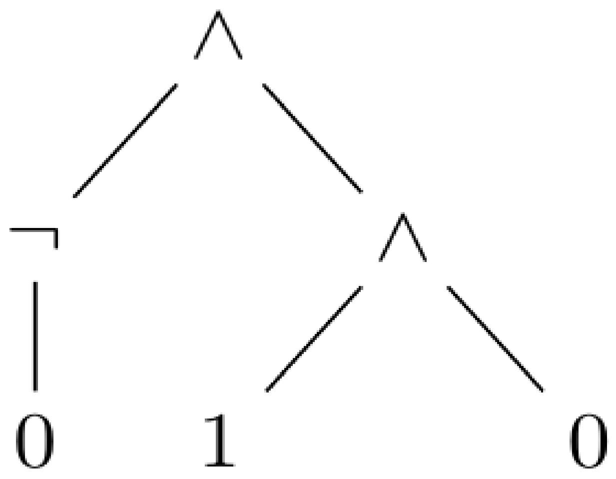
$$\delta(0) = q_0$$
$$\delta(1) = q_1$$
$$\delta(q_i, \neg) = q_{|i-1|}$$
$$\delta(q_i, q_j, \wedge) = q_{\min(i,j)}$$
$$\delta(q_i, q_j, \vee) = q_{\max(i,j)}$$

Lauf durch einen Baum

Positivbeispiel:



Negativbeispiel:



Definition ↓NTA

Ein nichtdeterministischer top-down Baumautomat ist ein Tupel $\mathcal{A} = (Q, \Sigma, Q_0, \Delta)$ mit

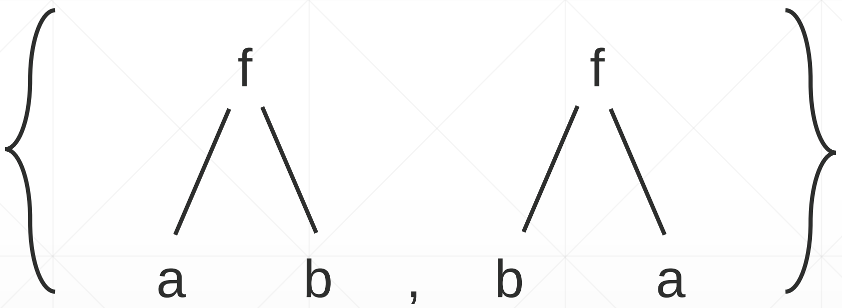
- Q eine endliche Menge von Zuständen,
- Σ ein Rangalphabet,
- $Q_0 \subseteq Q$ eine Menge der Startzustände,
- $\Delta \subseteq \bigcup_{\{i=0\}}^{m} (Q \times \Sigma_i \times Q^i)$ eine Übergangsrelation.

(dabei stellen die Paare in $\Delta \cap (Q \times \Sigma_0)$ die finalen Kombinationen dar)

$\mathcal{A}$ akzeptiert t nur, falls es einen vollständigen Lauft durch t gibt.

Beispiel ↓NTA

- Sei T die Sprache:



- Der folgende ↓NTA erkennt T:

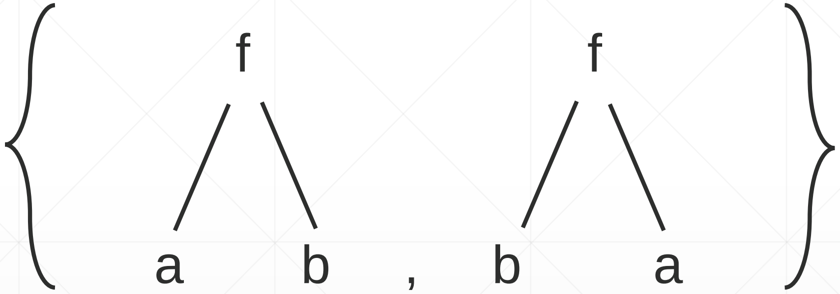
$$Q \coloneqq \{q_0, q_a, q_b\}$$
$$Q_0 \coloneqq \{q_0\}$$
$$\Delta \coloneqq \{(q_0, f, q_a, q_b), (q_0, f, q_b, q_a), (q_a, a), (q_b, b)\}$$

Finalen Kombinationen: $\{(q_a, a), (q_b, b)\}$

Beweis ↓NTA > ↓DTA

- ↓DTA ist ein spezieller ↓NTA
- Besitzt nur einen Startzustand $\{q_0\}$
- Besitzt nur Übergangsfunktionen:
  - $\delta_0 \ldots \delta_n$ mit $\delta_i\colon Q \times \Sigma_i \to Q^i$ anstatt Δ

- Sei T wieder:



Quelle

- Applied Automata Theory by Prof. Dr. Wolfgang Thomas RWTH Achen