

Operator Precedence Sprachen und ihre Automaten

Jonas Kremer

7. Februar 2019

Inhaltsverzeichnis

1	Motivation	3
2	Vorbereitung und Definitionen	3
2.1	Operatorpräzedenzgrammatik	3
2.2	Operatorpräzedenzautomaten	5
2.3	Äquivalenz von OPG und OPA	5
3	(Abschluss-) Eigenschaften von OPLs	5
3.1	Deterministischer vs. Nondeterministischer OPA	5
3.2	Abgeschlossenheit	5
3.3	Visibly Pushdown Sprache als Teilklasse	5
4	Implementation of OPLs	5
5	Conclusion	5

1 Motivation

2 Vorbereitung und Definitionen

In diesem Abschnitt geht es um die Definitionen und Namenskonventionen, die für Operatorpräzedenzsprachen benötigt werden. Zunächst kommt eine kurze Definition von kontextfreien Grammatiken, gefolgt von Definitionen und Einschränkungen für Operatorpräzedenzgrammatiken (OPG). Im Anschluss wird die Klasse der Operatorpräzedenzautomaten (OPA) eingeführt, sowohl deterministisch als auch nichtdeterministisch, sowie deren Äquivalenz zu OPGs bewiesen.

2.1 Operatorpräzedenzgrammatik

Eine kontextfreie Grammatik (kfG) ist ein 4-tupel $G = (N, \Sigma, P, S)$, wobei N die Menge der Nichtterminalsymbole, Σ die Menge der Terminalsymbole, P die Menge der Produktionsregeln und S das Startsymbol bezeichnet. Folgende Namenskonventionen werden im weiteren Verlauf verwendet: Kleine lateinische Buchstaben am Anfang des Alphabets a, b, \dots bezeichnen einzelne Terminalsymbole; Spätere, kleine lateinische Buchstaben u, v, \dots bezeichnen Terminalstrings; Große lateinische Buchstaben A, B, \dots stehen für Nichtterminalsymbole und griechische Buchstaben α, β, \dots für beliebige Strings über $N \cup \Sigma$. Sofern nicht explizit eingeschränkt können Strings auch leer sein.

Weiterhin haben Produktionsregeln die Form $A \rightarrow \alpha$ mit der *leeren Regel* $A \rightarrow \epsilon$. *Umbenennende* Regeln haben nur ein Nichtterminalsymbol als rechte Seite. Eine *direkte Ableitung* wird mit \Rightarrow beschrieben, eine beliebige Anzahl von Ableitungen mit \Rightarrow^* .

Eine Grammatik heisst *reduziert*, wenn jede Regel aus P benutzt werden kann um einen String aus Σ^* zu erzeugen. Sie ist *invertierbar*, wenn keine zwei Regeln identische rechten Seiten haben.

Eine Regel ist in *Operatorform*, wenn ihre rechte Seite keine benachbarten Nichtterminale hat. Entsprechend heisst eine Grammatik, die nur solche Regeln beinhaltet *Operatorgrammatik* (OG). Jede kfG $G = (N, \Sigma, P, S)$ kann in eine äquivalente Operatorgrammatik $G' = (N', \Sigma, P', S)$ umgewandelt werden [siam 25, 38]. Die folgenden Definitionen gelten für Operatorpräzedenzgrammatiken (OPGs).

Definition 2.1 Für eine OG G und ein Nichtterminal A sind die linken und rechten Terminalmengen definiert als

$$\mathcal{L}_G(A) = \{a \in \Sigma \mid A \xRightarrow{*} Ba\alpha\}, \quad \mathcal{R}_G(A) = \{a \in \Sigma \mid A \xRightarrow{*} \alpha aB\}$$

mit $B \in N \cup \{\epsilon\}$

Auf den Namen der Grammatik G kann verzichtet werden, wenn der Kontext klar ist. Eines der wichtigsten Merkmale von Operatorpräzedenzgrammatiken ist die Definition von drei binären Operatorpräzedenzrelationen.

Definition 2.2 (Präzedenzrelationen)

Gleiche Präzedenz: $a \doteq b \Leftrightarrow \exists A \rightarrow \alpha a B b \beta, B \in N \cup \{\epsilon\}$

Übernimmt Präzedenz: $a > b \Leftrightarrow \exists A \rightarrow \alpha D b \beta, D \in N \text{ and } a \in \mathcal{R}_G(D)$

Gibt Präzedenz ab: $a < b \Leftrightarrow \exists A \rightarrow \alpha a D \beta, D \in N \text{ and } b \in \mathcal{L}_G(D)$

Es muss beachtet werden, dass diese Relationen im Gegensatz zu ähnlichen arithmetischen Relationen ($<, >, =$) keine transitiven, symmetrischen oder reflexiven Eigenschaften vorliegen. Weiterhin schließt die Gültigkeit einer Relation die andere nicht aus, sodass zB. sowohl $a < b$ als auch $a \doteq b$ gelten kann. Für eine OG G kann eine Operatorpräzedenzmatrix (OPM) $M = OPM(G)$ als $|\Sigma| \times |\Sigma|$ erstellt werden, die für jedes geordnete Paar (a, b) die Menge M_{ab} der Operatorpräzedenzrelationen beinhaltet. Für solche Matrizen sind Inklusion und Vereinigung natürlich definiert.

Definition 2.3 Eine OG G ist eine OPG oder auch FG gdw. $M = OPM(G)$ eine konfliktfreie Matrix ist.

Also: $\forall a, b, |M_{ab}| \leq 1$

Eine OPL ist eine Sprache, die durch eine OPG gebildet wird.

Für solche OPMs werden nun weitere Namenskonventionen vereinbart. Zwei Matrizen sind *kompatibel*, wenn ihre Vereinigung konfliktfrei ist. Eine Matrix heisst *total*, wenn gilt $\forall a, b : M_{ab} \neq \emptyset$. Für eine OPG wird die *Fischer Normalform (FNF)* definiert.

Definition 2.4 (Fischernormalform) Eine OPG G ist in Fischer Normalform gdw. gilt:

- G ist invertierbar
- G hat keine leeren Regeln außer dem Startsymbol, sofern dies nicht weiter verwendet wird
- G hat keine umbenennenden Regeln

Zu jeder OPG kann eine äquivalente Grammatik in FNF gebildet werden. [ASDFGH] In Ausblick auf die Operatorpräzedenzautomaten erweitern wir die OPM um ein Symbol $\# \notin \Sigma$, welches Start und Ende eines Strings bezeichnet. Dieses Symbol beeinflusst die Grammatik nicht weiter, da für jedes Terminal gilt: $\forall a \in \Sigma : \# < a$ und $a > \#$. Ferner gilt: $M_{\# \#} = \{\doteq\}$.

Definition 2.5 Ein OP Alphabet ist ein Paar (Σ, M) mit:

Σ ist ein Alphabet

M ist eine konfliktfreie OPM, erweitert zu einem $|\Sigma \cup \{\#\}|^2$ Array, das zu jedem geordneten Paar (a, b) höchstens eine OP Relation enthält.

Zum Schluss sollte noch eine Einschränkung bezüglich der \doteq -Relation getroffen werden. Aus

- 2.2 Operatorpräzedenzautomaten
- 2.3 Äquivalenz von OPG und OPA
- 3 (Abschluss-) Eigenschaften von OPLs
 - 3.1 Deterministischer vs. Nondeterministischer OPA
 - 3.2 Abgeschlossenheit
 - 3.3 Visibly Pushdown Sprache als Teilklasse
- 4 Implementation of OPLs
- 5 Conclusion