

CSS-Grammatik

Anmerkungen

- erhöhtes Anforderungsniveau
- vorgesehene Bearbeitungszeit: 80 min

Aufgabe

Mit Hilfe von Cascading Style Sheets (CSS) können Layout und Inhalt von HTML-Dokumenten getrennt dargestellt werden. Die Syntax von CSS wird vom WWW-Konsortium als Grammatik dargestellt. Dabei werden folgende Metasymbole benutzt:

Metasymbol	Bedeutung
*	muss nicht auftreten, kann aber mehrfach wiederholt werden
+	muss ein- oder mehrfach wiederholt werden
[]	tritt in dieser Kombination auf
" "	zwischen den Anführungszeichen steht ein Terminalsymbol

Ein Metasymbol gehört nicht zur eigentlichen CSS-Sprache, sondern hilft, die Sprache zu definieren (Metaebene).

Die für diese Aufgabe vereinfachten Produktionen der CSS-Grammatik sehen wie folgt aus:

```
Stylesheet → Ruleset+
Ruleset → Selector [", " Selector]* "{" Declaration [";" Declaration]* "}"
Selector → Identifier
Declaration → Property ":" Expression
Property → Identifier
Expression → Number | Identifier | Number "%"
```

Dabei ist *Identifier* ein Bezeichner aus Buchstaben, Ziffern oder dem Bindestrich, der mit einem Buchstaben beginnt und *Number* eine Zahl aus den Ziffern 0 bis 9, die mit einer Ziffer beginnt, optional gefolgt von einem Dezimalpunkt und mindestens einer weiteren Ziffer.

1. Geben Sie die Terminale, die Nichtterminale und das Startsymbol der CSS-Grammatik an, beschreiben Sie die angegebenen Produktionen und ordnen Sie die Grammatik begründet in die Chomsky-Hierarchie ein.
2. Stellen Sie die angegebenen Produktionen als Syntaxdiagramme dar. Syntaxdiagramme für *Identifier* und *Number* brauchen nicht gezeichnet werden.
3. Entwerfen Sie äquivalente Produktionen, die statt mit Schleifen (Metasymbole + und *) mittels Rekursion definiert sind.
4. Geben Sie zum Nachweis, dass `h1{color:blue;line-height:120.5%}` ein Stylesheet ist, eine Ableitung an oder zeichnen Sie einen Ableitungsbaum.
- 5.1 Zeichnen Sie je einen endlichen Automaten für *Identifier* und *Number*.
- 5.2 Modellieren Sie einen deterministischen endlichen Automaten als Akzeptor für die durch die CSS-Grammatik definierte Sprache.

Lösungshinweise

Aufg.	erwartete Leistungen
1	<p>Terminale: {a, b, ..., z, A, B,..., Z, -, 0, 1,...9, ".", " ", ";", ":", "{", "}", "%"}</p> <p>Nichtterminale: {Stylesheet, Ruleset, Selector, Declaration, Property, Expression, Number, Identifier}</p> <p>Startsymbol S: Stylesheet</p> <p>Ein Stylesheet besteht aus einer nichtleeren Reihe von Rulesets. Ein Ruleset beginnt mit einem Selector auf den weitere mit Komma getrennte Selectoren folgen können. In einem Paar geschweifter Klammern folgen ein oder mehrere Declarations, die mit Semikolons zu trennen sind. Ein Selector ist ein Identifier. Ein Declaration besteht aus einer Property, dem Doppelpunkt als Terminal und einem Expression. Ein Property wird zu einem Identifier abgeleitet. Ein Expression ist ein Number, ein Identifier oder ein Number gefolgt von dem Terminal %.</p> <p>Es handelt sich um eine kontextfreie Grammatik (Typ 2), weil die Produktionsköpfe jeweils nur aus einem einzigen Nichtterminal bestehen und die Anzahl der Terminale und Nichtterminale in den Produktionsrümpfen mindestens 1 ist. Sie ist nicht regulär, weil z. B. Property → Identifier keine zulässige Produktion für reguläre Grammatiken ist.</p>
2	<p>The diagram illustrates the hierarchical structure of the grammar. It shows the following components and their relationships:</p> <ul style="list-style-type: none"> Stylesheet derives Ruleset. Ruleset derives a sequence of Selector and Declaration separated by commas (`,`). Selector derives Identifier. Declaration derives Property followed by a colon (`:`) and Expression. Property derives Identifier. Expression derives Number or Identifier. Number derives Expression followed by a percent sign (`%`). The entire Ruleset is enclosed in curly braces `{}`.
3	<p>Stylesheet → Ruleset Ruleset Stylesheet</p> <p>Ruleset → Selectors { Declarations }</p> <p>Selectors → Selector Selector , Selectors</p> <p>Declarations → Declaration Declaration ; Declarations</p> <p>alle anderen Produktionen wie bisher</p>

Aufg.	erwartete Leistungen
4	
5.1	
5.2	

Quelle: Hessisches Kultusministerium, Landesabitur Informatik Leistungskurs, 2010
 Lösungshinweise bearbeitet

Zuordnung zu den Prozess-, Inhalts- und Anforderungsbereichen

Aufg.	Prozessbereiche					Inhaltsbereiche					Bewertungseinheiten in Anforderungsbereichen		
	MI	BB	SV	KK	DI	ID	AL	SA	IS	IMG	I	II	III
1	X	X		X				X			5	1	
2					X			X				6	
3					X			X				4	2
4					X			X			3	3	
5.1					X			X			2		
5.2			X		X			X					4
Summe 30											10	14	6