Einschub: Reguläre Ausdrücke

Reguläre Sprachen ("Typ 3 - Sprachen") werden durch reguläre Grammatiken beschrieben ("generiert"). Das ist i.A. ziemlich umständlich! Bsp.: Ganze Zahlen

$S \to \phi$
$S \to 1$
$\vdots$
$S \to 9$
$S \to \phi S$
$S \to 1 S$
$\vdots$
$S \to 9 S$

Deswegen benutzt man "reguläre Ausdrücke" zum Beschreiben von regulären Sprachen. Diese können vollständig durch reguläre Grammatiken ausgedrückt werden: keine Veränderung der prinzipiellen Ausdrucksstärke! Aber es ist viel bequemer:

[ϕ-9]+

Bestandteile von regulären Ausdrücken:

einzelne Zeichen	a	
Alternative	$\beta \mid \gamma$	
Konkatenation	$\beta\gamma$	
Epsilon	ε	(leerer String)
Kleene'sche Abschluß	$\beta*$	$(= \varepsilon \mid \beta \mid \beta\beta \mid \ldots)$

Meist ebenfalls zugelassen:

Characterklasse	[abcd]	(= a|b|c|d)
	[b-e]	(= b|c|d|e)
Optionaler Ausdruck	$\beta?$	$(= \beta \mid \varepsilon)$
Positiv Abschluß	$\beta+$	$(= \beta\beta*)$

Irgendein Zeichen (außer Zeilenumbruch) .

Aufgaben

1. Binärzahlen, die Vielfache von 2 sind:
   $(0|1)*0$

2. Strings aus a und b, die mind. einmal zwei a's hintereinander enthalten:
   $(a|b)*aa(a|b)*$

3. Strings aus a und b, die keine zwei a's hintereinander enthalten:
   $b*(abb*)*(a|\varepsilon)$

4. Hexadezimalzahlen
   $0x[0-9a-fA-F]+$

5. Ein einzelnes Plus-Zeichen:
   $\backslash +$

6. Identifier in der Prog. Sprache C:
   $[a-zA-Z\_][a-zA-Z0-9\_]*$

   Hier sieht man die Nützlichkeit von "regulären Abkürzungen" (= Namen für reguläre Ausdrücke):

   Definiere $\{L\} = [a-zA-Z\_]$ ("Letter")

   $\{D\} = [0-9]$ ("Digit")

   Dann kann man den rg. Ausdruck für "Identifier" so schreiben:
   $\{L\}(\{L\}|\{D\})*$

Einschub: Kontextfreie Grammatiken

Kontextfreie Sprachen ("Typ 2-Sprachen") werden durch kontextfreie Grammatiken beschrieben ("generiert").

Bestandteile:

- Menge von Terminalsymbolen T (die "Tokens")
- Menge von Nichtterminalsymbolen N (die "Variablen")
- Menge von Produktionen der Form $l \to r$, wobei $l \in N$ und $r \in (T \cup N)^*$ ("String v. Grammatiksymb.")
- Ein ausgezeichnetes $S \in N$, das "Startsymbol".

Bsp.: Korrekte Klammerung

$T = \{a, l, r\}$, $N = \{S\}$

P: $S \to a$

$S \to l\,S\,r$

Abkürzung f. die Produktionen: $S \to a \mid l\,S\,r$

Bem.: Jede reguläre Sprache ist eine kontextfreie Sprache.

a	$S \to a$
ε	$S \to \varepsilon$
$\beta \mid \gamma$	$S \to \beta \mid \gamma$
$\beta\gamma$	$S \to \beta\gamma$
β^*	$S \to \varepsilon \mid \beta S$ ("rechtsrekursiv")
	alternativ: $S \to \varepsilon \mid S\beta$ ("linksrekursiv")

Also: Reguläre Sprachen $\subsetneq$ Kontextfreie Sprachen

Warum gibt es dann überhaupt Scanner?

Gründe: Effizienz, Einfachheit!

Aufgaben

(E4)

1. Eine Liste mit bel. vielen a's, einschl. keinem a:

   linksrekursiv: $S \to \varepsilon$

   $S \to Sa$

   rechtsrekursiv: $S \to \varepsilon$

   $S \to aS$

2. Eine Liste mit bel. vielen a's, aber mind. einem a:

   linksrekursiv: $S \to a$

   $S \to Sa$

   rechtsrekursiv: $S \to a$

   $S \to aS$

3. Eine Liste mit bel. vielen a's (einschließlich keinem), getrennt durch jeweils ein b ("Parameterliste"):

   linksrekursiv: $S \to \varepsilon \mid T$

   $T \to a \mid Tba$

   rechtsrekursiv: $S \to \varepsilon \mid T$

   $T \to a \mid abT$

4. Allgemeiner Tipp: Trennung von Wiederholung und der Spezifikation, was wiederholt wird!

   Bsp.: Eine bel. Folge von "Deklarationen" d und "Statements" e:

   $S \to \varepsilon \mid ST$

   $T \to d \mid e$

Einschub: Ableitungen, Parse-Bäume, Mehrdeutigkeit (E5)

Grammatik f. arithmet. Ausdrücke:

$E \to E+E \mid E-E \mid E*E \mid E/E \mid (E) \mid -E \mid id$

Ist $-(id+id)$ ein Satz der Grammatik?

Ja, denn es gibt eine "Ableitung": $E \Rightarrow -E \Rightarrow -(E)$
$\Rightarrow -(E+E) \Rightarrow -(id+E) \Rightarrow -(id+id)$

In jedem Herleitungsschritt zwei Entscheidungen:

a) Welches Nichtterminal wird ersetzt?

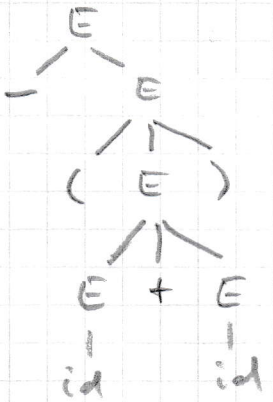
b) Welche Alternative für dieses Nichtterminal wird gewählt?

Links-Ableitungen: das am weitesten links stehende N.T.

Rechts-Ableitungen: " " " rechts " N.T.

Parse-Baum: grafische Darstellung einer Ableitung ohne Berücksichtigung der Reihenfolge.

Bsp: $-(id+id)$

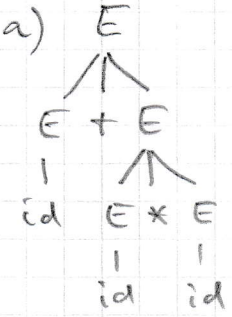


Syntaxanalyse = Finden einer Ableitung = Konstr. des Parse-Baums bei gegeb. Satz

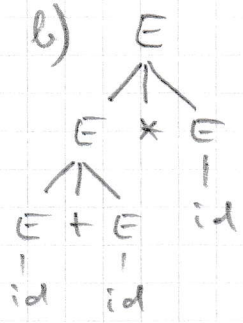
Eine Grammatik, bei der es mind. einen Satz gibt, der durch versch. Parse-Bäume repräsentiert wird, heißt "mehrdeutig".

Bsp.: Grammatik von oben, Satz $id+id*id$

a)



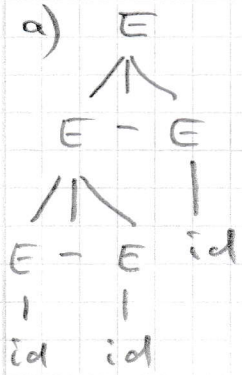
b)



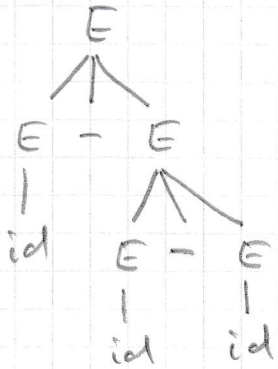
Wir wollen a) haben, wg. "Priorität" der Operatoren

Bsp.: Grammatik von oben, Satz id-id-id

a)



b)



Wir wollen a) haben, wg. "Assoziativität" des Operators

→ Grammatik ist mehrdeutig! Wie wird sie eindeutig, bei gegeb. Prioritäten und Assoziativitäten?

Neue Nichtterminale E ("Expression")
T ("Term")
F ("Faktor")

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T*F \mid T/F \mid F$

$F \rightarrow -F \mid id \mid (E)$

Die Priorität der Operatoren wird durch die "Staffelung" $E/T/F$ erreicht, die (Links-) Assoziativität durch die (Links-) Rekursion der Produktionen.