Priv.-Doz. Dr. T. Noll, C. Jansen

Aufgabe 1 (Lexical Analysis): (3+7+4+2 = 16 Points)

Assume you would like to automatically tag your audio files with information extracted from their filenames. You decide to categorise the extracted information into three classes:

- $TYPE \in \{AB, MU\}$, tells whether your file contains an audiobook (AB) or a piece of music (MU)
- $DESC$, an arbitrary non-empty string of letters and numbers, which contains the description of the file, i.e. title, artist etc.
- $EXT \in \{MP3, AAC\}$ indicating the audio format of the file

Example: The file "`MP3-Compiler-AB.MP3`" should be tagged with $TYPE = AB$, $DESC = Compiler$, $EXT = MP3$.

a) Construct regular expressions α_{TYPE}, α_{DESC}, α_{EXT} and give corresponding DFAs $\mathfrak{A}_{TYPE}$, $\mathfrak{A}_{DESC}$, $\mathfrak{A}_{EXT}$ such that $\mathfrak{L}(\mathfrak{A}_{TYPE}) = [\![\alpha_{TYPE}]\!]$, $\mathfrak{L}(\mathfrak{A}_{DESC}) = [\![\alpha_{DESC}]\!]$, $\mathfrak{L}(\mathfrak{A}_{EXT}) = [\![\alpha_{EXT}]\!]$.

b) Construct the backtracking automaton $\mathfrak{A}$ accepting

$$\alpha_1 := \alpha_{TYPE}, \alpha_2 := \alpha_{EXT}, \alpha_3 := \alpha_{DESC}$$

and additionally

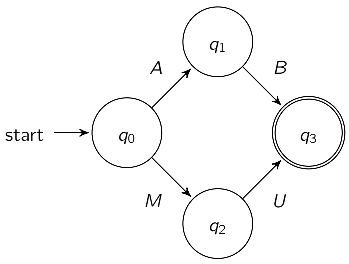
$$\alpha_4 := \alpha_{SEP} = -, \alpha_5 := \alpha_{DOT} = .$$

c) Compute the run of $\mathfrak{A}$ for the input word $w = MU - MUH.MP3$.

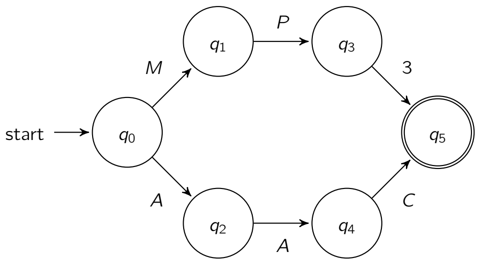
d) What is the worst case time complexity of a backtracking DFA?

Lösung:

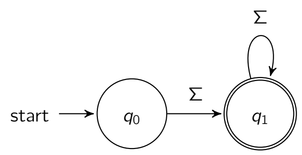
a) $\alpha_{TYPE} = AB + MU$



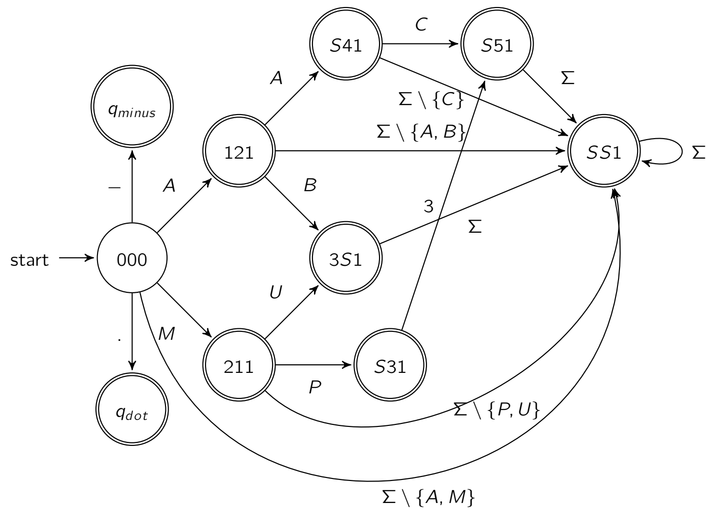
$\alpha_{EXT} = MP3 + AAC$



$\alpha_{DESC} = \Sigma^+$ with $\Sigma = \{a, \ldots, z, A, \ldots, Z, 0, \ldots 9\}$



b) Product automaton over alphabet $\Delta := \Sigma \cup \{., -\}$:



For all remaining $\delta \in \Delta$ edge to sink state.

Partitioning of final states:

$F_1 = \{3S1\}$, $F_2 = \{S51\}$, $F_3 = \{121, 211, S31, S41, SS1\}$, $F_4 = \{q-\}$, $F_5 = \{q.\}$

c) $(N, 000MU - MUH.MP3, \epsilon)$
$\vdash (T_3, 211U - MUH.MP3, \epsilon)$
$\vdash (T_1, 3S1 - MUH.MP3, \epsilon)$
$\vdash (N, 000 - MU, T_1)$
$\vdash (T_4, q_{minus}MUH.MP3, T_1)$
$\vdash (N, 000MUH.MP3, T_1T_4)$
$\vdash (T_3, 211UH.MP3, T_1T_4)$
$\vdash (T_1, 3S1H.MP3, T_1T_4)$
$\vdash (T_3, SS1.MP3, T_1T_4)$
$\vdash (N, 000.MP3, T_1T_4T_3)$
$\vdash (T_5, q_{dot}MP3, T_1T_4T_3)$
$\vdash (N, 000MP3, T_1T_4T_3T_5)$
$\vdash (T_3, 211P3, T_1T_4T_3T_5)$
$\vdash (T_3, S313, T_1T_4T_3T_5)$
$\vdash (T_2, S51, T_1T_4T_3T_5)$

output: $T_1T_4T_3T_5T_2$

d) $\mathcal{O}(|\omega|^2)$, ω input word

Aufgabe 2 (Grammar Transformation): (4+4+8+4 = 20 Points)

Consider the following grammar G:

$$S \quad \rightarrow \quad S + S \mid S* \mid a$$

a) Apply left-factorisation to the original grammar G.

b) Remove left-recursion from the original grammar G.

c) Specify fi- and fo-sets for all nonterminals in both resulting grammars.

d) Do the resulting grammars satisfy the $LL(1)$ property? Justify your answer!

Lösung:

a) Left-factorisation: G'

$$\begin{array}{lcl} S & \rightarrow & SA \mid a \\ A & \rightarrow & +S \mid * \end{array}$$

b) Left-Recursion: G''

$$\begin{array}{lcl} S & \rightarrow & aA \\ A & \rightarrow & +SA \mid *A \mid \epsilon \end{array}$$

c) G':

$fi(S) = \{a\}, fi(A) = \{+, *\}$

$fo(S) = \{\epsilon, +, *\} = fo(A)$

G'':

$fi(S) = \{a\}, fi(A) = \{+, *, \epsilon\}$

$fo(S) = \{\epsilon, +, *\} = fo(A)$

d) G': left-recursive OR

$la(S \rightarrow SA) = fi(SA \cdot fo(S)) = \{a\}$

$la(S \rightarrow a) = \{a\}$

$\rightarrow la(S \rightarrow SA) \cap la(S \rightarrow a) \neq \emptyset$

$\rightarrow G' \notin LL(1)$

G'':

$la(A \rightarrow +SA) = \{+\}$

$la(A \rightarrow *A) = \{*\}$

$la(A \rightarrow \epsilon) = fo(A) = \{\epsilon, +, *\}$

$\rightarrow la(A \rightarrow +SA) \cap la(A \rightarrow \epsilon) \neq \emptyset$

$\rightarrow G'' \notin LL(1)$

Aufgabe 3 (Top-Down Analysis): (10 Points)

Show that for every context-free grammar G the following holds:

$$G \text{ ambiguous } \Rightarrow G \notin LL(1)$$

Lösung:

$G = (N, \Sigma, P, S)$ ambiguous

$\Rightarrow \exists \omega \in \Sigma^*$ with at least two *left*-derivations $S \Rightarrow_l^* vA\alpha \begin{array}{l} \Rightarrow_l v\beta\alpha \Rightarrow_l^* vv' = \omega \\ \Rightarrow_l v\gamma\alpha \Rightarrow_l^* vv' = \omega \end{array}$

We assume this splitting to be the first difference appearing in the derivations.

As obviously $fi(v') = fi(v')$ and therefore $fi(\beta\alpha) \cap fi(\gamma\alpha) \neq \emptyset$

$\Rightarrow G \notin LL(1)$

Aufgabe 4 (Bottom-Up Analysis): (10+8+2 = 20 Points)

Transform the following grammar G into an equivalent start-separated grammar G':

$$\begin{array}{lll} S & \to & AS \mid b \quad (1,2) \\ A & \to & SA \mid a \quad (3,4) \end{array}$$

a) Specify the corresponding nondeterministic bottom-up analysis automaton $\mathfrak{A}_{G'}$ and give its run on the input word $w = abab$.

b) Compute all $LR(0)$ sets of G'.

c) Is $G' \in LR(0)$? Justify your answer!

Lösung:

$S' \to S \quad (0)$

a) Transitions:

Shift

$(a\omega, \alpha, z) \vdash (\omega, \alpha a, z)$
$(b\omega, \alpha, z) \vdash (\omega, \alpha b, z)$

Reduce

$(\omega, \alpha S, z) \vdash (\omega, \alpha S', z0)$
$(\omega, \alpha AS, z) \vdash (\omega, \alpha S, z1)$
$(\omega, \alpha b, z) \vdash (\omega, \alpha S, z2)$
$(\omega, \alpha SA, z) \vdash (\omega, \alpha A, z3)$
$(\omega, \alpha a, z) \vdash (\omega, \alpha A, z4)$

Run:

$(abab, \epsilon, \epsilon)$
$\vdash (bab, a, \epsilon)$
$\vdash (bab, A, 4)$
$\vdash (ab, Ab, 4)$
$\vdash (ab, AS, 42)$
$\vdash (b, ASa, 42)$
$\vdash (b, ASA, 424)$
$\vdash (b, AA, 4243)$
$\vdash (\epsilon, AAb, 4243)$
$\vdash (\epsilon, AAS, 42432)$
$\vdash (\epsilon, AS, 424321)$
$\vdash (\epsilon, S, 4243211)$
$\vdash (\epsilon, S', 42432110)$

b) Abbreviation: $F := [S \to \cdot AS], [S \to \cdot b][A \to \cdot SA][A \to \cdot a]$

$LR(0)(\epsilon) = [S' \to \cdot S]F$
$LR(0)(S) = [S' \to S\cdot][A \to S \cdot A]F$
$LR(0)(A) = [S \to A \cdot S]F = LR(0)(AA) = LR(0)(SAA) = LR(0)(\ldots AA)$
$LR(0)(b) = [S \to b\cdot] = LR(0)(\ldots b)$
$LR(0)(a) = [A \to a\cdot] = LR(0)(\ldots a)$
$LR(0)(SA) = [A \to SA\cdot][S \to A \cdot S]F = LR(0)(SSA) = LR(0)(\ldots SA)$

$LR(0)(SS) = [A \to S \cdot A]F = LR(0)(SSS) = LR(0)(\ldots SS)$
$LR(0)(AS) = [S \to AS\cdot][A \to S \cdot A]F = LR(0)(SAS) = LR(0)(\ldots AS)$

c) $G' \notin LR(0)$ as $LR(0)(S)$ contains a shift/reduce conflict.

Aufgabe 5 (Attributed Grammars): (2+3+7=12 Points)

Consider the following grammar G generating two 4-digit words separated by "$-$".

$$\begin{array}{lcl} S & \rightarrow & T - T \\ T & \rightarrow & NNNN \\ N & \rightarrow & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid \mathtt{X} \end{array}$$

An International Standard Serial Number (ISSN) consists of two 4-digit words separated by "$-$", i.e. $x_1x_2x_3x_4 - x_5x_6x_7x_8$, where x_1 to x_7 are supposed to be digits, i.e. $x_1, \ldots, x_7 \in \{0, \ldots, 9\}$. x_8 is a check digit with $x_8 \in \{0, \ldots, 9, \mathtt{X}\}$.
An ISSN is *valid*, if the check digit satisfies the following property:
$x_8 = 11 - (8 * x_1 + 7 * x_2 + \cdots + 2 * x_7) MOD\, 11$ where the symbol $\mathtt{X}$ represents the result 10.

Extend G to an attributed grammar which checks whether the generated sequence is a valid ISSN. Therefore specify:

a) the attributes and their domains

b) the association of attributes to symbols

c) the attribute equations for each production

Lösung:

a) $v \triangleq$ integer value of $x_1, \ldots, x_8$, $dom(v) = \{0, \ldots, 10\}$
$v_1, v_2, v_3, v_4 \triangleq$ values of each four-digit word, $dom(v_i) = \{0, \ldots, 10\}$, $i \in [1, 4]$
$valid \triangleq$ specifies if the derived word is a valis ISSN, $dom(valid) = \mathbb{B}$

b) $syn(N) = \{v\}$
$syn(T) = \{v_1, v_2, v_3, v_4\}$
$syn(S) = \{valid\}$
$Inh = \emptyset$

c) $N \rightarrow 0 : v.0 = 0$
$\ldots$
$N \rightarrow 9 : v.0 = 9$
$N \rightarrow X : v.0 = 10$
$T \rightarrow NNNN : v_1.0 = v.1,\ v_2.0 = v.2,\ v_3.0 = v.3, v_4.0 = v.4$

$$S \rightarrow T - T : valid.0 = \begin{cases} true & \text{if } v_1.1, \ldots, v_4.1 < 10 \wedge v_1.3, \ldots, v_3.3 < 10 \wedge \\ & \quad v_4.3 = (8 * v_1.1 + \ldots + 5 * v_4.1 + 4 * v_1.3 + \ldots + v_3.3) mod 11 \\ false & \text{otherwise} \end{cases}$$

Aufgabe 6 (Code Generation): (7+3 Points)

Consider the fragment C of the main program:

```
if ( (0 < y) or (0 < x * y) ) then
   y := x DIV y;
else x := 0;
```

with symbol table $st = st_\emptyset[x/(var, 1, 1), y/(var, 1, 2)]$.

a) Translate C into AM code, starting at address 1.

b) Determine the successor state of $(11, \epsilon, 5 : 4 : 2 : 4 : 2 : 0 : 0 : 0)$.

Lösung:

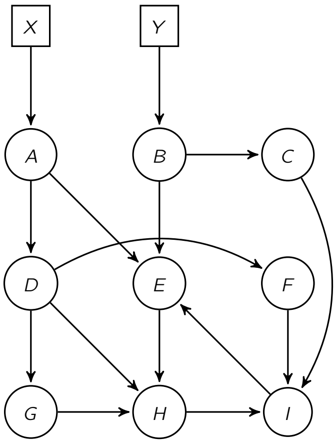
a)

```
1    LIT(0);
2    LOAD(0, 2);
3    LT;
4    LIT(0);
5    LOAD(0, 1);
6    LOAD(0, 2);
7    MULT;
8    LT;
9    OR;
10   JFALSE(16);
11   LOAD(0,1);
12   LOAD(0,2);
13   DIV;
14   STORE(0,2);
15   JMP(18);
16   LIT(0);
17   STORE(0,1);
18   ...
```

b) $(12, 4, 5 : 4 : 2 : 4 : 2 : 0 : 0 : 0)$

Aufgabe 7 (Garbage Collection): (3+5 Points)

Given the following heap configuration:



a) Specify the initial reference counts.

b) What happens to the reference counts when the pointer from A to D is deleted?

Lösung:

a) $A:1, B:1, C:1, D:1, E:3, F:1, G:1, H:3, I:3$

b) $A:1, B:1, C:1, D:0, E:3, F:0, G:0, H:1, I:2$