Section 4

Context-Free Languages

⟨assignment⟩ ::= ⟨variable⟩ = ⟨expression⟩

⟨statement⟩ ::= ⟨assignment⟩ |
      ⟨compound-statement⟩ |
      ⟨if-statement⟩ |
      ⟨while-statement⟩ | . . .

⟨if-statement⟩ ::=
      if ⟨test⟩ then ⟨statement⟩ |
      if ⟨test⟩ then ⟨statement⟩ else ⟨statement⟩

⟨while-statement⟩ ::=
      while ⟨test⟩ do ⟨statement⟩

Definition (well-formed formulas)

. . . by using the construction rules below, and only those, finitely many times:

– every propositional atom $p, q, r, \ldots$ is a wff

– if ϕ is a wff, then so is $(\neg \phi)$

– if ϕ and ψ are wff, then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$,

BNF   Backus Naur form

$\psi ::= p \mid (\neg \psi) \mid (\psi \wedge \psi) \mid (\psi \vee \psi) \mid (\psi \rightarrow \psi)$

M.Huet & M.Ryan, Logic in Computer Science

$AnBn \subseteq \{a, b\}^*$

Example

– $\Lambda \in AnBn$
– for every $x \in AnBn$, also $axb \in AnBn$

$Pal \subseteq \{a, b\}^*$

Example

– $\Lambda, a, b \in Pal$
– for every $x \in Pal$, also $axa, bxb \in Pal$

[M] E 1.18

Balanced $\subseteq \{(,)\}^*$

Example

– $\Lambda \in$ *Balanced*
– for every $x, y \in$ *Balanced*, also $xy \in$ *Balanced*
– for every $x \in$ *Balanced*, also $(x) \in$ *Balanced*

Expr $\subseteq \{a, +, *, (,)\}$

Example

– $a \in$ *Expr*
– for every $x, y \in$ *Expr*, also $x + y \in$ *Expr* and $x * y \in$ *Expr*
– for every $x \in$ *Expr*, als $(x) \in$ *Expr*

[M] E 1.19

Example

– $\Lambda \in AnBn$ (basis)

– for every $x \in AnBn$, also $axb \in AnBn$ (induction)

$S \to \Lambda$

$S \to aSb$

$$S \;\Rightarrow\; aSb \;\Rightarrow\; aaSbb \;\Rightarrow\; aa\,bb$$
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaa\,bbb$$

if $S \Rightarrow^* x$ then also $S \Rightarrow^* axb$

Example

– Λ, a, $b \in Pal$
– for every $x \in Pal$, also axa, $bxb \in Pal$

$S \rightarrow \Lambda \mid a \mid b$
$S \rightarrow aSa$
$S \rightarrow bSb$

$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aababaa$

$AnBn = \{\, a^n b^n \mid n \geqslant 0 \,\}$

variants

$\{\, a^n b^{n+1} \mid n \geqslant 0 \,\}$

$\quad S \to b \quad$ (end with extra b)

$\quad S \to aSb$

$\{\, a^i b^j \mid i \leqslant j \,\}$

$\quad S \to \Lambda$

$\quad S \to aSb \mid Sb \quad$ (free b's)

$\{\, a^i b^j \mid i \neq j \,\}$

$\quad S \to A \mid B \quad$ (choice!)

$\quad A \to aAb \mid aA \mid a \quad (i > j)$

$\quad B \to aBb \mid Bb \mid b \quad (i < j)$

Examples: recursion

Example

– $a \in Expr$
– for every $x, y \in Expr$, also $x + y \in Expr$ and $x * y \in Expr$
– for every $x \in Expr$, als $(x) \in Expr$

$S \rightarrow a \mid S + S \mid S * S \mid (S)$
derivation(s) for $a + (a * a)$ and $a + a * a$...
ambiguity
[M] E 4.2

$NonPal \subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

[M] E 4.3

Examples: recursion

NonPal $\subseteq \{a, b\}^*$
$x = abbbaaba \in$ *NonPal*

Example

– for every $A \in \{a, b\}^*$, *aAb* and *bAa* are elements of *NonPal*
– for every S in *NonPal*, *aSa* and *bSb* are in *NonPal*

$A \rightarrow \Lambda \mid aA \mid bA$
$S \rightarrow aAb \mid bAa \mid aSa \mid bSb$
[M] E 4.3

NonPal $\subseteq \{a, b\}^*$
$x = abbbaaba \in NonPal$

Example

– for every $A \in \{a, b\}^*$, *aAb* and *bAa* are elements of *NonPal*
– for every S in *NonPal*, *aSa*, *bSb*, *aSb* and *bSa* are in *NonPal*

$A \rightarrow \Lambda \mid aA \mid bA$
$S \rightarrow aAb \mid bAa \mid aSa \mid bSb \mid aSb \mid bSa$
[M] E 4.3

alphabet $\{\, 1, 2, 5, = \,\}$

$\{\, x{=}y \mid x \in \{1,2\}^*, y \in \{5\}^*, n_1(x) + 2n_2(x) = 5n_5(y) \,\}$

$n_\sigma(x)$ number of σ occurrences in x

$212{=}5 \qquad 22222{=}55 \qquad 12(122)^3 2 = 5^4$

The problem with most solutions is that when read from left to right the initial string over $\{1, 2\}$ cannot always be chopped into part with exact value 5, without chopping the symbol 2.

The solution is like a finite automaton, which reads $1, 2$ and 'saves' the values until the value 5 is reached, then we write a 5 to the right.

$\Sigma = \{\, 1, 2, 5, = \,\}$

variables S_i, $0 \leqslant i \leqslant 4$

axiom S_0

productions

$S_0 \rightarrow 1 S_1 \mid 2 S_2$
$S_1 \rightarrow 1 S_2 \mid 2 S_3$
$S_2 \rightarrow 1 S_3 \mid 2 S_4$
$S_3 \rightarrow 1 S_4 \mid 2 S_0 5$
$S_4 \rightarrow 1 S_0 5 \mid 2 S_1 5$
$S_0 \rightarrow =$

Definition

context-free grammar (CFG)   4-tuple $G = (V, \Sigma, S, P)$
– V alphabet   *variables* / *nonterminals*
– Σ alphabet   *terminals*   disjoint $V \cap \Sigma = \varnothing$
– $S \in V$ *axiom*, *start symbol*
– P finite set   rules, *productions*
  of the form $A \to \alpha, \quad A \in V, \alpha \in (V \cup \Sigma)^*$

derivation step $\alpha = \alpha_1 A \alpha_2 \Rightarrow_G \alpha_1 \gamma \alpha_2 = \beta$ for $A \to \gamma \in P$

Definition

language generated by G
$L(G) = \{\, x \in \Sigma^* \mid S \Rightarrow_G^* x \,\}$

[M] Def 4.6 & 4.7

NonPal, its grammar components

$$A \rightarrow \Lambda \mid aA \mid bA$$
$$S \rightarrow aAb \mid bAa \mid aSa \mid bSb$$

variables $V = \{ S, A \}$
terminals $\Sigma = \{ a, b \}$
axiom S
productions
$P = \{A \rightarrow \Lambda, A \rightarrow aA, A \rightarrow bA, S \rightarrow aAb, S \rightarrow bAa, S \rightarrow aSa, S \rightarrow bSb\}$

$\Rightarrow_G^*$ is the *transitive and reflexive closure* of $\Rightarrow_G$

zero, one or more steps

general case $\quad \alpha = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n = \beta$

$\alpha \Rightarrow_G^* \beta$ iff there are strings $\alpha_0, \alpha_1, \ldots, \alpha_n$ such that
$-\ \alpha_0 = \alpha$
$-\ \alpha_n = \beta$
$-\ \alpha_i \Rightarrow \alpha_{i+1} \quad$ for $0 \leqslant i < n$.

special case $\quad n = 0 \quad \alpha = \alpha_0 = \beta$

Variables can be rewritten regardless of context

Lemma

If $u_1 \Rightarrow^* v_1$ and $u_2 \Rightarrow^* v_2$, then $u_1 u_2 \Rightarrow^* v_1 v_2$.

Lemma

If $u \Rightarrow^* v_1 v v_2$ and $v \Rightarrow^* w$, then $u \Rightarrow^* v_1 w v_2$.

Lemma

If $u \Rightarrow^* v$ and $u = u_1 u_2$,
then $v = v_1 v_2$ such that $u_1 \Rightarrow^* v_1$ and $u_2 \Rightarrow^* v_2$.

$AeqB = \{\, x \in \{a, b\}^* \mid n_a(x) = n_b(x) \,\}$

$aaabbb$, $ababab$, $aababb$, ...

[M] E 4.8

From lecture 6:
– Even number of both a and b
two letters together
aa and bb keep both numbers even [odd]
ab and ba switch between even and odd, for both numbers
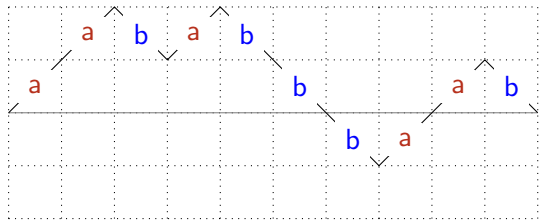
$(\, aa + bb + (ab + ba)(aa + bb)^*(ab + ba) \,)^*$

[M] E 3.4

$AeqB = \{\, x \in \{a, b\}^* \mid n_a(x) = n_b(x) \,\}$

$aaabbb, ababab, aababb, \ldots$

[M] E 4.8

Examples: recursion

$AeqB = \{\, x \in \{a, b\}^* \mid n_a(x) = n_b(x) \,\}$

$AeqB = \{\, x \in \{a, b\}^* \mid n_a(x) = n_b(x) \,\}$

$aaabbb, ababab, aababb, \ldots$

$$S \rightarrow \Lambda \mid aB \mid bA$$
$$A \rightarrow aS \mid bAA \qquad\qquad A \text{ generates } n_a(x) = n_b(x) + 1$$
$$B \rightarrow bS \mid aBB \qquad\qquad B \text{ generates } n_a(x) + 1 = n_b(x)$$

$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow \ldots$ (different options)

(1) $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$
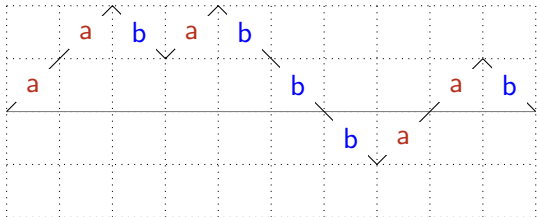
(2) ... (ambiguous, later)

[M] E 4.8

When a string has multiple variables, like *aabSB* in the above example, then we are not forced to rewrite the first variable, we can as well rewrite another one.

Thus we can do $aab\underline{S}B \Rightarrow aabB$, but also $aabS\underline{B} \Rightarrow aabSaBB$, for instance.

$AeqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$

$S \rightarrow \Lambda \mid aSb \mid bSa \mid SS$

$S \Rightarrow SS \Rightarrow a_1 Sb_6 S \Rightarrow a_1 a_2 Sb_3 Sb_6 S \Rightarrow \ldots$

$S \Rightarrow a_1 Sb_{10} \Rightarrow \ldots$

[M] Exercise 1.66

$L_1 = \{\, a^i b^j c^k \mid i = j + k \,\}$ *aaa b cc*

$L_1 = \{\, a^i b^j c^k \mid i = j + k \,\}$ $aaa\,b\,cc$

generate as $a^{k+j}\, b^j\, c^k = \underbrace{a^k\, a^j\, b^j}\, c^k$

$S \to aSc \mid T$

$T \to aTb \mid \Lambda$

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaTcc \Rightarrow aaaTbcc \Rightarrow aaabcc$

$L_1 = \{\, a^i b^j c^k \mid i = j + k \,\}$ $aaa\,b\,cc$

generate as $a^{k+j}\,b^j\,c^k = a^k\,\underbrace{a^j\,b^j}\,c^k$

$S \rightarrow aSc \mid T$

$T \rightarrow aTb \mid \Lambda$

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaTcc \Rightarrow aaaTbcc \Rightarrow aaabcc$

$L_2 = \{\, a^i b^j c^k \mid j = i + k \,\}$ $a\,bbb\,cc$

$L_1 = \{\, a^i b^j c^k \mid i = j + k \,\}$ $aaa\, b\, cc$

generate as $a^{k+j}\, b^j\, c^k = a^k \underbrace{a^j\, b^j}\, c^k$

$S \to aSc \mid T$
$T \to aTb \mid \Lambda$

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaTcc \Rightarrow aaaTbcc \Rightarrow aaabcc$

$L_2 = \{\, a^i b^j c^k \mid j = i + k \,\}$ $a\, bbb\, cc$

generate as $a^i\, b^{i+k}\, c^k = \underbrace{a^i\, b^i}\, \underbrace{b^k\, c^k}$

$S \to XY$ (concatenate)
$X \to aXb \mid \Lambda$
$Y \to bYc \mid \Lambda$

$S \Rightarrow \underline{X}\, Y \Rightarrow a\underline{X}b\, Y \Rightarrow ab\, \underline{Y} \Rightarrow ab\, b\underline{Y}c \Rightarrow ab\, bb\underline{Y}cc \Rightarrow abbbcc$
$S \Rightarrow X\, \underline{Y} \Rightarrow \underline{X}\, bYc \Rightarrow aXb\, b\underline{Y}c \Rightarrow a\underline{X}b\, bbYcc \Rightarrow ab\, bb\underline{Y}cc \Rightarrow$
$abbbcc$

(a priori there is no prescribed order rewriting X or Y)

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

[M] Thm 4.9

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

$G_i = (V_i, \Sigma, S_i, P_i)$, having no variables in common.

[M] Thm 4.9

Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

$G_i = (V_i, \Sigma, S_i, P_i)$, having no variables in common.

Construction

$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P)$,   new axiom S
$- P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$ $L(G) = L(G_1) \cup L(G_2)$
$- P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$ $L(G) = L(G_1) L(G_2)$

$G = (V_1 \cup \{S\}, \Sigma, S, P)$,   new axiom S
$- P = P_1 \cup \{S \to S S_1, S \to \Lambda\}$ $L(G) = L(G_1)^*$

[M] Thm 4.9

$L_0 = \{\, a^i b^j c^k \mid j = i + k \,\} = \{\, a^i b^{i+k} c^k \mid j = i + k \,\}$
$ = \{\, \underbrace{a^i b^i}\, \underbrace{b^k c^k} \mid j = i + k \,\}$

$S_0 \rightarrow XY \quad X \rightarrow aXb \mid \Lambda \quad Y \rightarrow bYc \mid \Lambda$

$L = \{\, a^i b^j c^k \mid j \neq i + k \,\}$

[M] E 4.10

$L_0 = \{ a^i b^j c^k \mid j = i + k \} = \{ a^i b^{i+k} c^k \mid j = i + k \}$
$ = \{ \underbrace{a^i b^i}_{} \underbrace{b^k c^k}_{} \mid j = i + k \}$

$\quad S_0 \rightarrow XY \quad X \rightarrow aXb \mid \Lambda \quad Y \rightarrow bYc \mid \Lambda$

$L = \{ a^i b^j c^k \mid j \neq i + k \} = L_1 \cup L_2$
$S \rightarrow S_1 \mid S_2$

$L_1 = \{ a^i b^j c^k \mid j > i + k \}$
$L_2 = \{ a^i b^j c^k \mid j < i + k \}$

[M] E 4.10

$L_0 = \{\, a^i b^j c^k \mid j = i + k \,\} = \{\, a^i b^{i+k} c^k \mid j = i + k \,\}$
$ = \{\, \underbrace{a^i b^i}\, \underbrace{b^k c^k} \mid j = i + k \,\}$

$\quad S_0 \to XY \quad X \to aXb \mid \Lambda \quad Y \to bYc \mid \Lambda$

$L = \{\, a^i b^j c^k \mid j \neq i + k \,\} = L_1 \cup L_2$
$S \to S_1 \mid S_2$

$L_1 = \{\, a^i b^j c^k \mid j > i + k \,\}$
$\quad S_1 \to X_1 b Y_1$
$\quad X_1 \to aX_1 b \mid X_1 b \mid \Lambda$
$\quad Y_1 \to bY_1 c \mid bY_1 \mid \Lambda$

$L_2 = \{\, a^i b^j c^k \mid j < i + k \,\}$

[M] E 4.10

$$L_0 = \{\, a^i b^j c^k \mid j = i + k \,\} = \{\, a^i b^{i+k} c^k \mid j = i + k \,\}$$
$$= \{\, \underbrace{a^i b^i}\, \underbrace{b^k c^k} \mid j = i + k \,\}$$

$$S_0 \rightarrow XY \quad X \rightarrow aXb \mid \Lambda \quad Y \rightarrow bYc \mid \Lambda$$

$L = \{\, a^i b^j c^k \mid j \neq i + k \,\} = L_1 \cup L_2$
$S \rightarrow S_1 \mid S_2$

$L_1 = \{\, a^i b^j c^k \mid j > i + k \,\}$
$\quad S_1 \rightarrow X_1 b Y_1$
$\quad X_1 \rightarrow aX_1 b \mid X_1 b \mid \Lambda$
$\quad Y_1 \rightarrow bY_1 c \mid bY_1 \mid \Lambda$

$L_2 = \{\, a^i b^j c^k \mid j < i + k \,\}$
$\quad S_2 \rightarrow aX_2 Y_2 \mid X_2 Y_2 c$
$\quad X_2 \rightarrow aX_2 b \mid aX_2 \mid \Lambda$
$\quad Y_2 \rightarrow bY_2 c \mid Y_2 c \mid \Lambda$

[M] E 4.10

De uitwerking uit het boek is wat te ingewikkeld, dat hebben we hier wat ingekort.