

Formal Languages-Course 5.

Géraud Sénizergues

Bordeaux university

18/05/2020

Master computer-science MINF19, IEI, 2019/20

contents

- 1 Context-free grammars : ambiguity
- 2 Context-free languages :closure properties
- 3 Context-free grammars :transformations
 - Cleaning context-free grammars
 - A simple transformation
 - Quadratic normal form

Context-free grammars : ambiguity

context-free grammars : ambiguity

Definition

A context-free grammar $G = \langle A, N, R, \sigma \rangle$ is called **ambiguous** iff there exist **derivations trees** T_1, T_2 , with root σ , such that $T_1 \neq T_2$ and $\text{fr}(T_1) = \text{fr}(T_2) \in A^*$.

- 1- The grammar G is called **non-ambiguous** iff it is not ambiguous.
- 2- In words : G is ambiguous iff there exist two **different** derivation-trees T_1, T_2 with the same frontier $w \in A^*$.
- 3- Equivalently, G is ambiguous iff one of the following is true :
 - there are two different **leftmost-derivations** D_1, D_2 that end in the same terminal word w
 - there are two different **rightmost-derivations** D_1, D_2 that end in the same terminal word w
 - there are two different **construction-trees** that have the same yield w .

context-free grammars : ambiguity

Example

Let $G = \langle A, N, R, S \rangle$ with

$$A = \{a, b\}, \quad N = \{S\},$$

$$R : S \rightarrow aS, \quad S \rightarrow aSb, \quad S \rightarrow ab.$$

$$L = L(G, S) = \{a^n b^m \mid n \geq m \geq 1\}.$$

$$D_1 : S \xrightarrow{\ell} aS \xrightarrow{\ell} aaSb \xrightarrow{\ell} aaabb$$

$$D_2 : S \xrightarrow{\ell} aSb \xrightarrow{\ell} aaSb \xrightarrow{\ell} aaabb$$

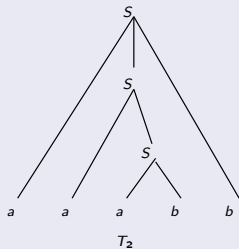
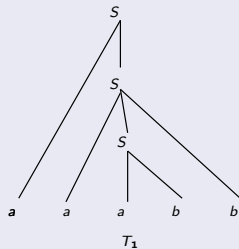
context-free grammars : ambiguity

Example

Let $G = \langle A, N, R, S \rangle$ with $A = \{a, b\}$, $N = \{S\}$,

$R: S \rightarrow aS, S \rightarrow aSb, S \rightarrow ab.$

$L = L(G, S) = \{a^n b^m \mid n \geq m \geq 1\}.$



context-free grammars : ambiguity

Example

Let $H = \langle A, N', R', S \rangle$ with

$$A = \{a, b\}, \quad N' = \{S, T\},$$

$$R' : S \rightarrow aS, \quad S \rightarrow T.$$

$$T \rightarrow aTb, \quad T \rightarrow ab.$$

$$L = L(H, S) = \{a^n b^m \mid n \geq m \geq 1\}.$$

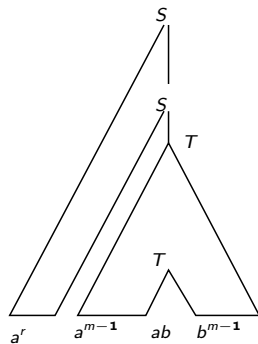
This new grammar for the same language L is *non-ambiguous*.

Let $w = a^n b^m$ with $n \geq m \geq 1$.

The only leftmost derivation for $a^{m+r} b^m$ is $D(r, m)$:

$$S \xrightarrow{r}_{R'} a^r S \xrightarrow{1}_{R'} a^r T \xrightarrow{m-1}_{R'} a^r a^{m-1} T b^{m-1} \xrightarrow{1}_{R'} a^r a^{m-1} a b b^{m-1} = a^{m+r} b^m.$$

context-free gramars : ambiguity



context-free grammars : ambiguity

Example

Let $H = \langle A, N', R', S \rangle$ be the previous grammar. Let us name the rules :

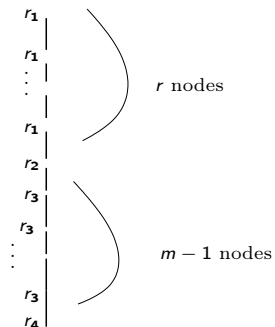
$$r1 : S \longrightarrow aS, \quad r2 : S \longrightarrow T.$$

$$r3 : T \longrightarrow aTb, \quad r4 : T \longrightarrow ab.$$

The only construction-tree for $a^{m+r}b^m$ is :

$$r_1^r \cdot r_2 \cdot r_3^{m-1} \cdot r_4.$$

context-free grammars : ambiguity



$$\text{yd}(T) = a^r a^m b^m$$

context-free grammars : ambiguity

Example

$H = \langle A, N, P \rangle$ with set of rules :

$$C \longrightarrow CsC \mid bCe \mid wC \mid i$$

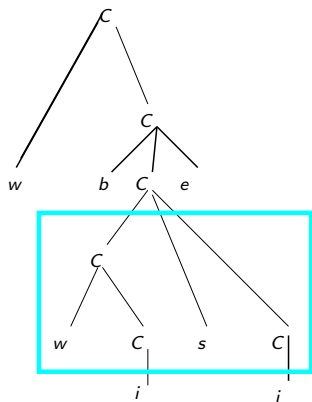
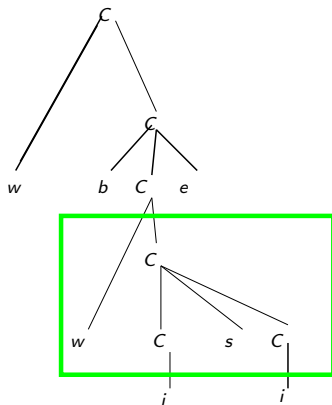
We already considered :

D_1 : $C \xrightarrow{\ell} wC \xrightarrow{\ell} wbCe \xrightarrow{\ell} wbCsCe \xrightarrow{\ell} wbwCsCe$
 $\xrightarrow{\ell} wbwisCe \xrightarrow{\ell} wbwisie$ But *there is also* :

D_2 : $C \xrightarrow{\ell} wC \xrightarrow{\ell} wbCe \xrightarrow{\ell} wbwCe \xrightarrow{\ell} wbwCsCe \xrightarrow{\ell}$
 $wbwisCe \xrightarrow{\ell} wbwisie$

context-free grammars : ambiguity

The two corresponding derivation-trees are :

 T_1  T_2

context-free gramars : ambiguity

wbwsie can be seen, according to T_1 as :

```
while cond
--begin
--while cond
----i
--;
```

--i
--end or, according to T_2 as :

```
while cond
--begin
--while cond
----i
----;
```

```
----i
--end
```

context-free grammars : ambiguity

Example

$H = \langle A, \{C\}, P, C \rangle$ with set of rules :

$$C \longrightarrow CsC \mid bCe \mid wC \mid i$$

Another grammar (for the **same** language) could be :

$K = \langle A, \{C, D\}, R, C \rangle$ with set of rules :

$$C \longrightarrow DsC \mid D \quad D \longrightarrow wD \mid bCe \mid i$$

Intuition : D generates the “**prime**” instructions (those which cannot be factored as usv with u, v instructions)

C generates **all** instructions : $c = d_0 s d_1 \cdots s d_{n-1}$
 where each d_j is a prime instruction.

context-free gramars : ambiguity

$K = \langle A, \{C, D\}, R, C \rangle$ with set of rules :

$$C \longrightarrow DsC \mid D \quad D \longrightarrow wD \mid bCe \mid i$$

The leftmost derivation of $wbwisie$ in this grammar is :

$$C \xrightarrow{3} wbCe, \quad wbCe \xrightarrow{2} wbDsDe \xrightarrow{3} wbwisie,$$

producing the structure :

```
while cond
--begin
--while cond
----i
--;
--i
--end
```

Context-free languages : closure properties

Closure properties

Theorem

Let A be a finite alphabet.

1- The set of context-free languages over A^* is closed under the operations : *union, product, star, cross*.

2- If $\text{Card}(A) \geq 2$, then the set of context-free languages over A^* is *not closed* under complement ; it is also *not closed* under intersection.

We give a *constructive* proof of point 1. Hence, from c.f. grammars generating languages L_1, L_2 , one can construct c.f. grammars for the languages $L_1 \cup L_2, L_1 \cdot L_2, L_1^*$.

Closure properties

sketch of proof of point 1

Let $G_1 = \langle A, N_1, R_1, S_1 \rangle$, $G_2 = \langle A, N_2, R_2, S_2 \rangle$. We assume that $N_1 \cap N_2 = \emptyset$.

union :

$$H = \langle A, N_1 \cup N_2 \cup \{\sigma\}, R, \sigma \rangle$$

where $R = R_1 \cup R_2 \cup \{\sigma \rightarrow S_1, \sigma \rightarrow S_2\}$

$$L(H, \sigma) = L(G_1, S_1) \cup L(G_2, S_2).$$

Closure properties

Let $G_1 = \langle A, N_1, R_1, S_1 \rangle$, $G_2 = \langle A, N_2, R_2, S_2 \rangle$.

product :

$$K = \langle A, N_1 \cup N_2 \cup \{\sigma\}, R, \sigma \rangle$$

where $R = R_1 \cup R_2 \cup \{\sigma \rightarrow S_1 S_2\}$

$$L(K, \sigma) = L(G_1, S_1) \cdot L(G_2, S_2).$$

star :

$$K_* = \langle A, N_1 \cup \{\sigma\}, R, \sigma \rangle$$

where $R = R_1 \cup \{\sigma \rightarrow \sigma S_1, \sigma \rightarrow \varepsilon\}$

$$L(K_*, \sigma) = L(G_1, S_1)^*.$$

□

Closure properties

point 2 :Let

$$L = \{a^n b^n c^n \mid n \geq 1\}.$$

L is **not context-free**.

Intuitive idea : if it was, there would be a repetition on some branch of a derivation-tree for $w = a^N b^N c^N$, leading, by pumping, to a larger derivation-tree with frontier $w' = a^{N+k} b^{N+\ell} c^N \notin L$ or $w'' = a^N b^{N+k} c^{N+\ell} \notin L$.

Let $L_1 = \{a^p b^q c^r \mid p, q, r \in \mathbb{N} \setminus \{0\}, p = q\}$, $L_2 = \{a^p b^q c^r \mid p, q, r \in \mathbb{N} \setminus \{0\}, q = r\}$.

L_1, L_2 are **context-free** (exercise).

$L = L_1 \cap L_2$ is **not context-free**.

Closure properties

point 2 :Let

$$L = \{a^n b^n c^n \mid n \geq 1\}.$$

L is **not** context-free.

$$L_3 = \{a^p b^q c^r \mid p, q, r \in \mathbb{N} \setminus \{0\}, p = q\},$$

$$L_4 = \{a^p b^q c^r \mid p, q, r \in \mathbb{N} \setminus \{0\}, q = r\}.$$

L_3, L_4 are **context-free** (exercise).

$$L = L_3 \cap L_4.$$

This shows that $CF(\{a, b, c\}^*)$ is **not closed** under intersection.

$$L = C(CL_3 \cup CL_4).$$

This shows that $CF(\{a, b, c\}^*)$ is **not closed** under complement.

Closure properties

A corollary of the above theorem, point 1, is

Proposition

Let A be a finite alphabet. Then every *regular* subset of A^* is also *context-free*.

One can also give a direct construction of a c.f. grammar for every finite automaton : let $\mathcal{A} = \langle Q, A, \delta, q_0, F \rangle$ be some nfa .
Let $G = \langle A, N, R, S_{q_0} \rangle$ with

$$N = \{S_q \mid q \in Q\},$$

$$R : \quad \begin{array}{ll} S_p \longrightarrow xS_q, & \text{for every } (p, x, q) \in \delta \\ S_q \longrightarrow \varepsilon & \text{for every } p \in F. \end{array}$$

Closure properties

One can prove, by induction on $|u|$ that :

$$S_p \xrightarrow{*}_R u \cdot S_q \Leftrightarrow p \xrightarrow{u}_{\mathcal{A}} q,$$

showing that

$$L(G, S_{q_0}) = L(\mathcal{A}).$$

Closure properties

Theorem

Let A be a finite alphabet.

For every languages $L \in \text{CF}(A^)$ and $R \in \text{REG}(A^*)$, $L \cap R$ is context-free.*

Closure properties

proof of the theorem :

Let $G = \langle A, N, P, S \rangle$ be a c.f. grammar and $\mathcal{A} = \langle Q, A, \delta, q_0, F \rangle$ be some nfa .

We build the c.f. grammar : $H = \langle A, N', P', \sigma \rangle$ with

$$N' = \{(p, v, q) \mid p, q \in Q, v \in N\}.$$

For every rule $U \rightarrow w_1 U_1 \cdot w_2 U_2 \cdots w_k U_k w_{k+1}$ in P we insert in P' the set of all rules

$$(p, U, q) \rightarrow w_1(p_1, U_1, q_1)w_2(p_2, U_2, q_2) \cdots w_k(p_k, U_k, q_k)w_{k+1}$$

where

$$p \xrightarrow{w_1}_{\mathcal{A}} p_1, \quad q_1 \xrightarrow{w_2}_{\mathcal{A}} p_2 \cdots q_{k-1} \xrightarrow{w_k}_{\mathcal{A}} p_k, \quad q_k \xrightarrow{w_{k+1}}_{\mathcal{A}} q.$$

Closure properties

We add in P' all the rules

$$\sigma \longrightarrow (q_0, S, q)$$

for $q \in F$. One can prove that, for every $v \in N, p, q \in Q$

$$L(H, (p, v, q)) = L(G, v) \cap \{u \in A^* \mid p \xrightarrow{u}_A q\}.$$

It follows that :

$$L(H, \sigma) = L(G, S) \cap L(\mathcal{A}).$$

Closure properties

Exercise

Is $CF(A^*)$ closed under *reversal* ?

Is $CF(A^*)$ closed under *regular substitution* ?

Is $CF(A^*)$ closed under *context-free substitution* ? (i.e. substitutions σ such that, for every $x \in A$, $\sigma(\{x\}) \in CF(A^*)$)

Context-free grammars : transformations

clean grammars

Let $G = \langle A, N, R, \sigma \rangle$ be a context-free grammar. A non-terminal $S \in N$ is said **productive** iff

$$\exists u \in A^*, S \xrightarrow{*}_R u.$$

A non-terminal S is said **useful** iff

$$\exists \alpha, \beta, u \in A^*, \sigma \xrightarrow{*}_R \alpha S \beta \text{ and } S \xrightarrow{*}_R u.$$

clean grammars

Proposition

Let $G = \langle A, N, R, \sigma \rangle$ be a context-free grammar. The set N' of *productive non-terminals* of G can be computed.

1- If $\sigma \notin N'$, then $L(G, \sigma) = \emptyset$.

2- If $\sigma \in N'$, then the grammar $G' = \langle A, N', R', \sigma \rangle$ with $R' = R \cap N' \times (A \cup N')^*$ has *productive non-terminals only* and generates the same language as G .

clean grammars

Let $G = \langle A, N, R, \sigma \rangle$.

Point 1 :Let

$$N_1 = \{S \in N \mid \exists w \in A^*, S \xrightarrow{1}_R w\}.$$

$$N_{i+1} = \{S \in N \mid \exists w \in (A \cup N_i)^*, S \xrightarrow{1}_R w\} \cup N_i.$$

One can show that

$$N' = \bigcup_{i \geq 1} N_i.$$

The sequence N_i is increasing and included in the finite set N , hence there exists $k \leq \text{Card}(N)$ such that

$$N_1 \subset N_2 \subset \dots \subset N_k = N_{k+1}.$$

It follows that

$$N' = \bigcup_{i \geq 1} N_i = N_k$$

clean grammars

Point 2 : If $\sigma \xrightarrow{*}_{R'} u$, since $R' \subseteq R$, also $\sigma \xrightarrow{*}_R u$.

If $\sigma \xrightarrow{*}_R u$, every non-terminal used in this derivation is productive, hence $\sigma \xrightarrow{*}_{R'} u$.

clean grammars :example

Consider the context-free grammar $G := \langle A, N, R, S_1 \rangle$ where $A = \{a, b, c\}$, $N = \{S_1, S_2, S_3, S_4, S_5, S_6\}$ and R consists of the following 15 rules :

$$\begin{array}{lll}
 S_1 \rightarrow aS_3S_5 & S_1 \rightarrow bS_2S_2 & S_1 \rightarrow aS_4 \\
 S_2 \rightarrow aS_3S_3 & S_2 \rightarrow bS_6c & \\
 S_3 \rightarrow bS_2 & S_3 \rightarrow aS_3S_6 & S_3 \rightarrow cS_6 \\
 S_4 \rightarrow bS_3S_4 & S_4 \rightarrow aS_4S_4 & S_4 \rightarrow b \\
 S_5 \rightarrow aS_4S_5 & S_5 \rightarrow bS_1 & S_5 \rightarrow aS_5. \\
 S_6 \rightarrow aS_3.
 \end{array}$$

$$N_1 = \{S_4\}, N_2 = \{S_1, S_4\}, N_3 = \{S_1, S_4, S_5\}, N_4 = N_3.$$

Hence the set of productive non-terminals is

clean grammars :example

$N' = \{S_1, S_4, S_5\}$. In grammar G the productions using $N \setminus N'$ must be **removed** :

$$\begin{array}{lll}
 S_1 \rightarrow aS_3S_5 & S_1 \rightarrow bS_2S_2 & S_1 \rightarrow aS_4 \\
 S_2 \rightarrow aS_3S_3 & S_2 \rightarrow bS_6c & \\
 S_3 \rightarrow bS_2 & S_3 \rightarrow aS_3S_6 & S_3 \rightarrow cS_6 \\
 S_4 \rightarrow bS_3S_4 & S_4 \rightarrow aS_4S_4 & S_4 \rightarrow b \\
 S_5 \rightarrow aS_4S_5 & S_5 \rightarrow bS_1 & S_5 \rightarrow aS_5. \\
 S_6 \rightarrow aS_3.
 \end{array}$$

The new grammar G' has the set of productions R' :

$$\begin{array}{lll}
 S_1 \rightarrow aS_4 & S_4 \rightarrow aS_4S_4 & S_4 \rightarrow b \\
 S_5 \rightarrow aS_4S_5 & S_5 \rightarrow bS_1 & S_5 \rightarrow aS_5.
 \end{array}$$

clean grammars

Proposition

Let $G' = \langle A, N', R', \sigma \rangle$ be a context-free grammar with only productive non-terminals.

1- The set N'' of useful non-terminals can be computed.

2- The grammar $G'' = \langle A, N'', R'', \sigma \rangle$ with $R'' = R' \cap (N'' \times (A \cup N'')^*)$ has *useful non-terminals only* and generates the same language as G' .

clean grammars

Let $G' = \langle A, N', R', \sigma \rangle$.

Point 1 :Let

$$M_1 = \{\sigma\}.$$

$$M_{i+1} = \{S \in N' \mid \exists \alpha, \beta \in (A \cup N')^*, \exists T \in M_i, T \xrightarrow{1}_{R'} \alpha S \beta\} \cup M_i.$$

One can show that

$$N'' = \bigcup_{i \geq 1} M_i.$$

The sequence M_i is increasing and included in the finite set N , hence there exists $k \leq \text{Card}(N)$ such that

$$M_1 \subset M_2 \subset \dots \subset M_k = M_{k+1}.$$

It follows that

$$N'' = \bigcup_{i \geq 1} M_i = M_k$$

□

clean grammars

Point 2 : If $\sigma \xrightarrow{*}_{R''} u$, since $R'' \subseteq R'$, also $\sigma \xrightarrow{*}_{R'} u$.

If $\sigma \xrightarrow{*}_{R'} u$, every non-terminal used in this derivation is useful,
hence $\sigma \xrightarrow{*}_{R''} u$.

clean grammars :example

$G' = \langle A, N', R', S_1 \rangle$ where R' is the set of rules :

$$\begin{aligned} S_1 &\rightarrow aS_4 & S_4 &\rightarrow aS_4S_4 & S_4 &\rightarrow b \\ S_5 &\rightarrow aS_4S_5 & S_5 &\rightarrow bS_1 & S_5 &\rightarrow aS_5. \end{aligned}$$

$$M_1 = \{S_1\}, M_2 = \{S_1, S_4\}, M_3 = M_2$$

$$N'' = \{S_1, S_4\}$$

$G'' = \langle A, N'', R'', S_1 \rangle$ where R'' is the set of rules :

$$S_1 \rightarrow aS_4 \quad S_4 \rightarrow aS_4S_4 \quad S_4 \rightarrow b$$

A simple transformation

Proposition

Let $G = \langle A, N, R \rangle$ be a context-free grammar and $U \in N$ such that U does *not occur in the right-hand side* of the rules with left-hand side U .

One can transform G into a grammar $G' = \langle A, N \setminus \{U\}, R' \rangle$ such that, for every $S \in N \setminus \{U\}$, $L(G', S) = L(G, S)$.

A simple transformation

Notation : given languages L_1, L_2 we note $L_1 + L_2$ the union $L_1 \cup L_2$. A set of grammar rules with same left-hand side :

$$S \longrightarrow w_1, S \longrightarrow w_2, \dots, S \longrightarrow w_n$$

is denoted by :

$$S \longrightarrow w_1 + w_2 + \dots + w_n$$

Given a c.f. grammar $G = \langle A, N, R, \sigma \rangle$, where $N = \{S_1, \dots, S_i, \dots, S_n\}$, its set of rules R can always be written under the form :

$$S_1 \longrightarrow \sum_{j=1}^{\ell_1} w_{1,j}, \quad S_i \longrightarrow \sum_{j=1}^{\ell_i} w_{i,j}, \quad S_n \longrightarrow \sum_{j=1}^{\ell_n} w_{n,j}.$$

A simple transformation

Let $G = \langle A, N, R \rangle$, be a grammar of the previous form and $U = S_n$ fulfills the assumption that :

$$\forall j \in [1, \ell_n], w_{n,j} \in (A \cup \{S_1, \dots, S_{n-1}\})^*.$$

We define the substitution $\sigma : \mathcal{P}(A \cup N)^* \rightarrow \mathcal{P}(A \cup N)^*$ by :

$$\forall x \in A, \sigma(x) = x, \quad \forall S \in N \setminus \{U\}, \sigma(S) = S, \quad \sigma(U) = \sum_{j=1}^{\ell_n} w_{n,j}.$$

Let $G' = \langle A, N \setminus \{U\}, R', \sigma \rangle$ be defined by : for every $1 \leq i \leq n-1$:

$$S_i \longrightarrow \sum_{j=1}^{\ell_i} \sigma(w_{i,j})$$

(i.e. R' consists of all the rules above with lhs and rhs in $N' = \{S_1, \dots, S_{n-1}\}$).

A simple transformation

This grammar does not use $U = S_n$ anymore. Let us show that, for every $i \in [1, n - 1]$:

$$L(G', S_i) = L(G, S_i).$$

1-

$$\forall w \in \sigma(U), \quad U \xrightarrow{1}_R w$$

Hence,

$$\forall v \in (A \cup N)^*, \forall w \in \sigma(v), \quad v \xrightarrow{*}_R w$$

this implies that

$$\forall i \in [1, n - 1], \quad \forall w \in \sum_{j=1}^{\ell_i} \sigma(w_{i,j}), \quad S_i \xrightarrow{*}_R w. \quad (1)$$

By (1) $R' \subseteq \xrightarrow{*}_R$, implying that, for every $i \in [1, n - 1]$

$$L(G', S_i) \subseteq L(G, S_i).$$

A simple transformation

2- Let us prove by induction on d that :for every $i \in [1, n - 1], w \in A^*$:

$$S_i \xrightarrow{d}_R w \Rightarrow S_i \xrightarrow{*}_{R'} w.$$

Let us suppose that :

$$S_i \xrightarrow{1}_R w_{i,j} = \alpha_1 S_n \alpha_2 S_n \cdots \alpha_k S_n \alpha_{k+1} \xrightarrow{d-1}_R w, \quad (2)$$

where $\alpha_i \in (A \cup \{S_1, \dots, S_{n-1}\})^*$.

Base : $d = 1$ i.e. $w_{i,j} = w$.

Then $w_{i,j} = w$ hence $w \in \sigma(w_{i,j})$, hence $S_i \xrightarrow{1}_{R'} w$.

A simple transformation

Induction : $d \geq 2$. Derivation (2) shows that

$w = \hat{\alpha}_1 w_1 \hat{\alpha}_2 w_2 \cdots \hat{\alpha}_k w_k \hat{\alpha}_{k+1}$ for some words $\hat{\alpha}_i, w_i \in A^*$ such that $\alpha_i \xrightarrow{a_i}_R \hat{\alpha}_i, S_n \xrightarrow{d_i}_R w_i$ and $(\sum_{i=1}^k a_i + d_i) + a_{k+1} = d - 1$. Hence

$$S_n \xrightarrow{1}_R w_{n,j_i} \xrightarrow{d_i-1}_R w_i$$

A simple transformation

By induction hypothesis, since $w_{n,j_k} \in (A \cup \{S_1, \dots, S_{n-1}\})^*$ and $a_i < d, d_i < d$

$$\alpha_i \xrightarrow{*}_{R'} \hat{\alpha}_i, \quad w_{n,j_i} \xrightarrow{*}_{R'} w_i \quad (3)$$

Note that

$$w' = \alpha_1 w_{n,j_1} \alpha_2 w_{n,j_2} \cdots \alpha_k w_{n,j_k} \alpha_{k+1} \in \sigma(w_{i,j})$$

Combining the rule $S_i \xrightarrow{1}_{R'} w'$ with derivations (3) give a derivation

$$S_i \xrightarrow{1}_{R'} w' = \alpha_1 w_{n,j_1} \alpha_2 w_{n,j_2} \cdots \alpha_k w_{n,j_k} \alpha_{k+1} \xrightarrow{*}_{R'} w.$$

Hence

$$L(G, S_i) \subseteq L(G', S_i).$$

A simple transformation : example

Example

Let $G = \langle A, N, R, S_1 \rangle$ with

$A = \{a, b, c, d\}$, $N = \{S_1, S_2, S_3, S_4\}$, and R consists of the rules :

$$S_1 \rightarrow aS_2bS_2 + \varepsilon$$

$$S_2 \rightarrow cS_3 + S_3d$$

$$S_3 \rightarrow cS_3 + \varepsilon$$

$$S_4 \rightarrow S_4d + \varepsilon$$

Since S_2 does not appear in its right-hand side, we can remove S_2 and obtain : $G = \langle A, N', R', S_1 \rangle$ with $N' = \{S_1, S_3, S_4\}$,

A simple transformation : example

Example

*(continued)**R' consists of the rules :*

$$S_1 \rightarrow a(cS_3 + S_3d)b(cS_3 + S_3d) + \varepsilon$$

$$S_3 \rightarrow cS_3 + \varepsilon$$

$$S_4 \rightarrow S_4d + \varepsilon$$

which can be written as :

$$S_1 \rightarrow acS_3bcS_3 + acS_3bS_3d + aS_3dbcS_3 + aS_3dbS_3d + \varepsilon$$

$$S_3 \rightarrow cS_3 + \varepsilon$$

$$S_4 \rightarrow S_4d + \varepsilon$$

w-Quadratic normal form

Proposition

Every c.f. grammar $G = \langle A, N, R, \sigma \rangle$ can be transformed into a c.f. grammar $G' = \langle A, N', R', \sigma' \rangle$ such that

- 1- $\forall (S, m) \in R', |m| \leq 2$
- 2- $L(G, \sigma) = L(G', \sigma')$.

Some remarks :

such a grammar is called in **weak quadratic** normal form (**quadratic** would mean that : $\forall (S, m) \in R', |m| = 2$ or $(S = \sigma$ and $m = \varepsilon)$).

w-Quadratic normal form

step 1 :Let $G = \langle A, N, R, \sigma \rangle$ such that one rule $S \rightarrow w$ is such that $|w| \geq 3$. We can decompose

$$w = w_1 \cdot w_2 \text{ with } |w_1| < |w|, |w_2| < |w|.$$

Let $G' = \langle A, N', R', \sigma \rangle$ be defined by :

$$\begin{aligned} N' &= N \cup \{W_1, W_2\}, \\ R' &= R \setminus \{S \rightarrow w_1 \cdot w_2\} \\ &\cup \{S \rightarrow W_1 \cdot W_2, W_1 \rightarrow w_1, W_2 \rightarrow w_2\}. \end{aligned}$$

Since W_1 (resp. W_2) does not appear in w_1 (resp. w_2), applying twice the above transformation, G' is mapped into G . Hence

$$L(G, \sigma) = L(G', \sigma).$$

w-Quadratic normal form

step 2 : As long as there remains some rhs with length ≥ 3 , we apply step 1 :

$$G = G_0 \rightsquigarrow G_1 \rightsquigarrow G_2 \rightsquigarrow \dots \rightsquigarrow G_n$$

until we obtain a c.f. grammar G_n where all rhs have length ≤ 2 . Since for every $i \in [0, n - 1]$, the grammars G_i, G_{i+1} generate the same language, we have obtained a new grammar G_n in **weak quadratic** normal form for the language $L(G, \sigma)$.

w-Quadratic normal form : example

Example

Let $G = \langle A, N, R, S \rangle$ where $A = \{a, b, c, d\}$, $N = \{S\}$ and R consists of the following rules :

$$S \rightarrow aSbSc + d$$

We obtain successively :

$$N = \{S, W_1, W_2\}, \quad S \rightarrow W_1W_2 + d, \quad W_1 \rightarrow aSb, \quad W_2 \rightarrow Sc,$$

$$N = \{S, W_1, W_2, W_3, W_4\}, \quad S \rightarrow W_1W_2 + d, \quad W_1 \rightarrow W_3W_4, \\ W_2 \rightarrow Sc, \quad W_3 \rightarrow aS, \quad W_4 \rightarrow b.$$