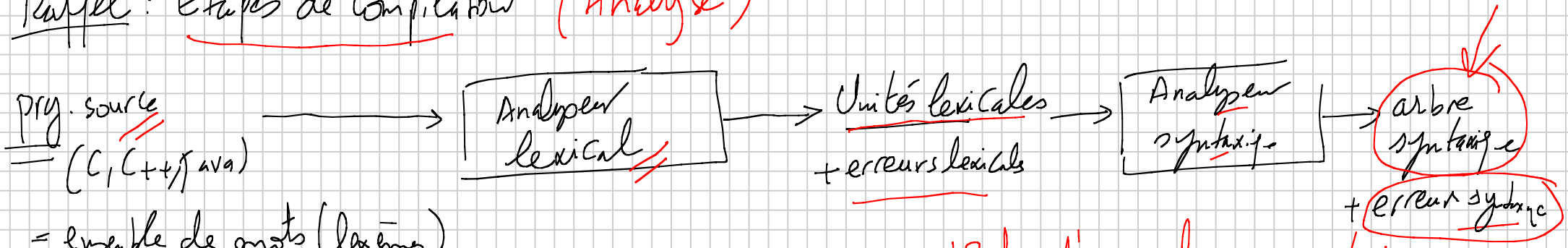
Chapitre III : Analyse syntaxique.

Titre de la note 23/11/2021

Rappel : Etapes de compilation (Analyse)

elses



Tâche d'un analyseur syntaxique :

Reconnaître les phrases / instructions

→ Correctes
→ erreurs

Exemple :

1) int a; // inst correcte
2) float x // inst non correcte (oublier le ;)
3) x = 4; // correcte
4) x + 1 = 6; // incorrecte

Variable = Exp (affectation)

~~Exp = variable / Exp~~

Comment : Pour reconnaître une phrase / instruction pour un langage (pg, naturel) on doit définir une grammaire pour ce langage.

Def : grammaire = ensemble de règles qui permet de reconnaître les phrases / instructions d'un langage et aussi de produire / générer des phrases / instruction de ce langage.

Analyse

génération

Ist V = Exp

x = x + 1

Exemple 1 : Extrait du grammaire Français

R1 : Une phrase est composée d'un groupe nominal (GN) suivi du verbe noir au présent ensuite d'un autre groupe nominal

R1 : PHRASE ⟶ GN noir GN

peut s'écrire

R2 : Un groupe nominal peut être composé d'un déterminant (DET) suivi d'un nom (NOM).

R2 : GN ⟶ DET NOM

R3 : Exemple de déterminant = { le, un }.

R3 : DET ⟶ le ou un

R4 : Exemple de nom = { chat, chien }

NOM ⟶ chat | chien

Question : Donner les étapes de génération / dérivation pour obtenir la phrase : le chat voit un chien.

PHRASE $\xrightarrow{R1}$ GN voit GN $\xrightarrow{R2}$ DET NOM voit GN $\xrightarrow{R3}$ le NOM voit GN

$\xrightarrow{R4}$ le chat voit GN $\xrightarrow{R2}$ le chat voit DET NOM $\xrightarrow{R3}$ le chat voit un NOM

$\xrightarrow{R4}$ le chat voit un chien

L1 L2 L3 mot L4 = istr

(= Dérivation / génération gauche

Définition : Une grammaire $G = (V_t, V_n, P, S)$ avec

1) V_t : ensemble des terminaux (lexèmes). $V_t = \{$ noir, le, un, chat, chien $\}$

2) V_n : ensemble des non-terminaux (unité lexicale) $V_n = \{$ PHRASE, NOM, DET, GN $\}$

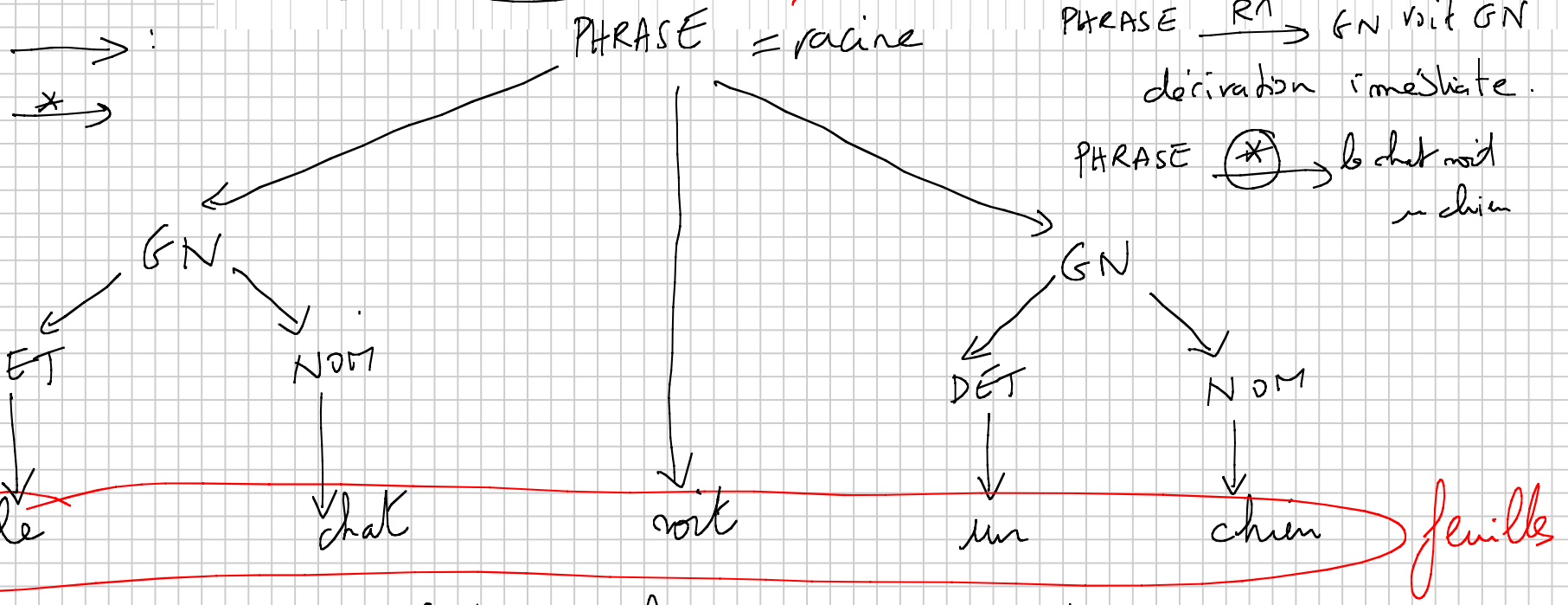
3) P : règles de production $P = \{ R1, R2, R3, R4 \}$

4) $S \in V_n$: Axiome $S =$ PHRASE.

Arbre syntaxique

PHRASE —R1→ GN voit GN —R2→ DET NOM voit GN —R3→ le NOM voit GN —R4→ le chat voit GN —R2→ le chat voit DET NOM —R3→ le chat voit un NOM —R4→ le chat voit un chien

(= Dérivation / génération gauche)



PHRASE $\xrightarrow{R1}$ GN voit GN

dérivation immédiate.

PHRASE $\xrightarrow{*}$ le chat voit un chien

PHRASE2 : voit le chat un chien : erreur syntaxique.

Arbre syntaxique = Arbre de dérivation = représentation graphique d'une séquence de lexèmes ($\in V_t$). Les nœuds représentent les non-terminaux ($\in V_n$). Les arcs représentent les étapes de dérivation

Exemple : Considérons la grammaire définie par :

R1 : $S \longrightarrow aSbT \mid cT \mid d$

R2 $T \longrightarrow aT \mid bS \mid c$

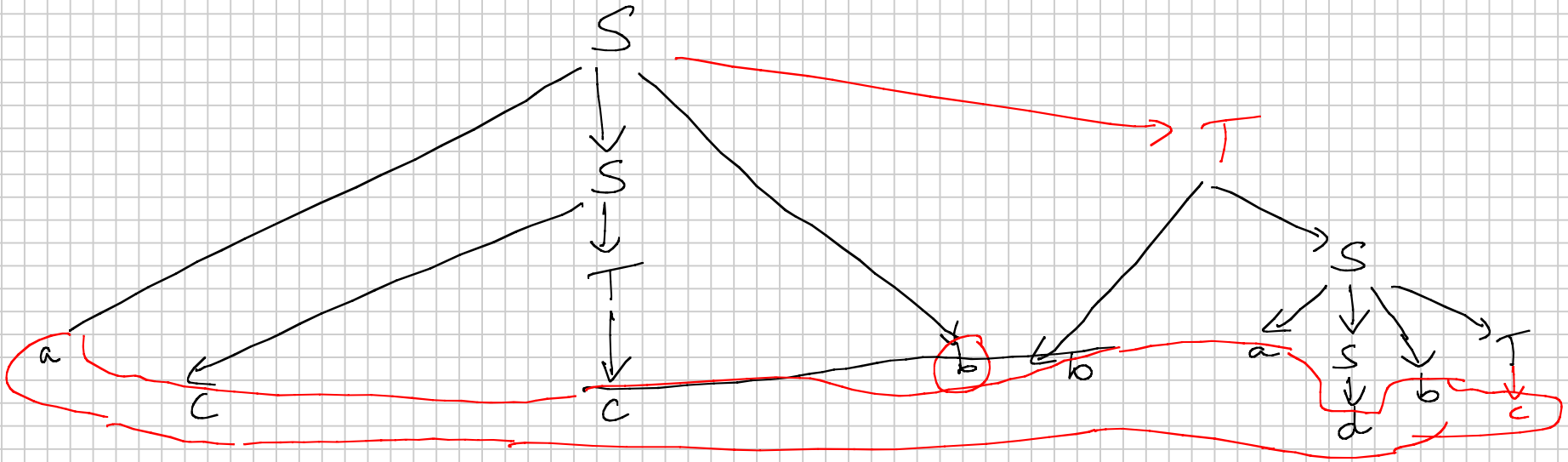
avec $V_t = \{a, b, c, d\}$

$V_n = \{S, T\}$

Axiome : S

Question : Donner l'arbre syntaxique pour le mot

instruction = mot $w = @ccbbadbc$ (@ : lexème) $\equiv$ if $(x < y)$

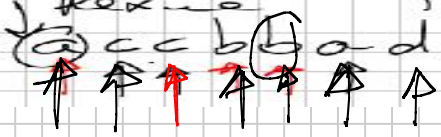

S
S
T
a
C
C
b
T
S
a
S
d
b
T
c
b

Considérons la grammaire définie par

R1 : $S \longrightarrow aSbT \mid cT \mid d$

R2 : $T \longrightarrow aT \mid bS \mid c$

Donner l'arbre syntaxique pour le mot $w = accbbadbc$



$S \longrightarrow aSbT \longrightarrow acTbT \longrightarrow accbT \longrightarrow accbbS \longrightarrow accbbaSbT$

$\longrightarrow accbbadbT \longrightarrow accbbadbc = w$

Résumé : Analyse syntaxique

Donnés : 1) phrase = ensemble de lexèmes

2) $G=(V_t, V_n, P, S)$ grammaire

Sortie : Arbre syntaxique

+ erreurs syntaxique.

Exemple : Soit la grammaire G définie par

R1 : $S \longrightarrow aB$

R2 $B \longrightarrow bc \mid bB$

$V_t = \{a, b, c\}$

$V_n = \{S, B\}$

$P = \{R1, R2\}$

Axiome = S.

Question : Est-ce que le mot

$w = abc$ est accepté par G ?

$S \longrightarrow aB \longrightarrow abc$

(lire le prochain lexème pour décider)

Si on choisit $B \longrightarrow bB$: $S \longrightarrow aB \longrightarrow abB$: echec

LL(1)

LL(2)

LL(K)

Rq : En général, on deux méthodes d'analyse syntaxique

① Méthode d'analyse descendante (TOP DOWN) : On construit l'arbre syntaxique de haut en bas : en partant de la racine vers les feuilles.

② Méthode d'analyse ascendante : (On construit l'arbre syntaxique en partant des feuilles vers la racine.

Méthode utilisée : Analyse prédictive : méthode descendante dans laquelle, on peut tjs choisir une règle unique en se basant sur le prochain lexème et sans effectuer aucun retour arrière.

L'analyse prédictive nécessite le calcul de certaines fonctions :
Nullable, Premier (First), Suivant (Follow).

La fct Nullable : $G = (V_t, V_n, P, S)$

α : terminal
$\alpha = a$
α : non terminal
$\alpha = S$
$\alpha = aS$

Def $\alpha \in (V_t \cup V_n)^*$ une forme de G, on dit que α est Nullable et on écrit Nullable(α) = Vrai
($\Leftrightarrow$) on peut dériver le mot vide ε à partir de α.
($\Leftrightarrow$) $\alpha \xrightarrow{*} \varepsilon$

Exple : R1 : $R \to bR \mid \varepsilon$
R2 : $T \to R \mid aTc$

$R \to \varepsilon$ Nullable(R) = True
$T \xrightarrow{*} \varepsilon$ Nullable(T) = True $R \xrightarrow{*} \varepsilon$
$T \to R \to \varepsilon$

Règles de calcul de Nullable :

1) Nullable(ε) = True $\quad \varepsilon \rightarrow \varepsilon$

2) $a \in V_t$, Nullable(a) = False $\quad a \not\rightarrow \varepsilon$

3) $\alpha, \beta \in (V_t \cup V_n)^*$ $\quad$ Nullable($\alpha.\beta$) = Nullable(α) AND Nullable(β)

4) $\forall X \in V_n$ avec $\quad X \longrightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n \quad \alpha_i \in (V_t \cup V_n)^*$

Nullable(X) = Nullable(α_1) OR Nullable(α_2) OR ... Nullable(α_n)

Exeple: Soit la grammaire G définie par

$$T \longrightarrow R \mid aTc$$
$$R \longrightarrow bR \mid \varepsilon$$

Question: Calculer Nullable(T), Nullable(R).