Grammaires hors contexte

Grammaires catégorielles

engendre les m̂ langages

lexicalisées

analyses très proche de la sémantique logique de la phrase

apprenables de manière exacte (à la limite)

1953) grammaires catégorielles AB
1930) Ajdukiewicz Bar-Hillel.

2 règles les mêmes pour tous les langages

catégories de base n, np, S
gn

noun phrase groupe nominal
phrase syntagme
sentence phrase

categorie complexes
construites à partir de 2 operateurs

$A \backslash B$ A sous B

B / A B sur A

$C ::= B \mid C/C \mid C \backslash C$

catégories de base

$$\underline{C ::=} \; B \;\; | \;\; C \backslash C \;\; | \;\; C / C$$

ou ... ou

$$B = \{n, np, S\}$$

si b est une categorie de base
alors b est une categorie
si C et C' sont des categorie,
alors C/C' et C'\C sont
des categories.
rien d'autre n'est une categories

$(np\backslash S)/np$ est une catégorie

$np\backslash S$ catégories $\longrightarrow$ $np\backslash S$ est une catégorie

np est une catégories

$(np\backslash S)/np$
est une catégories

2 règles

$$A \cdot A \backslash B \longrightarrow B$$

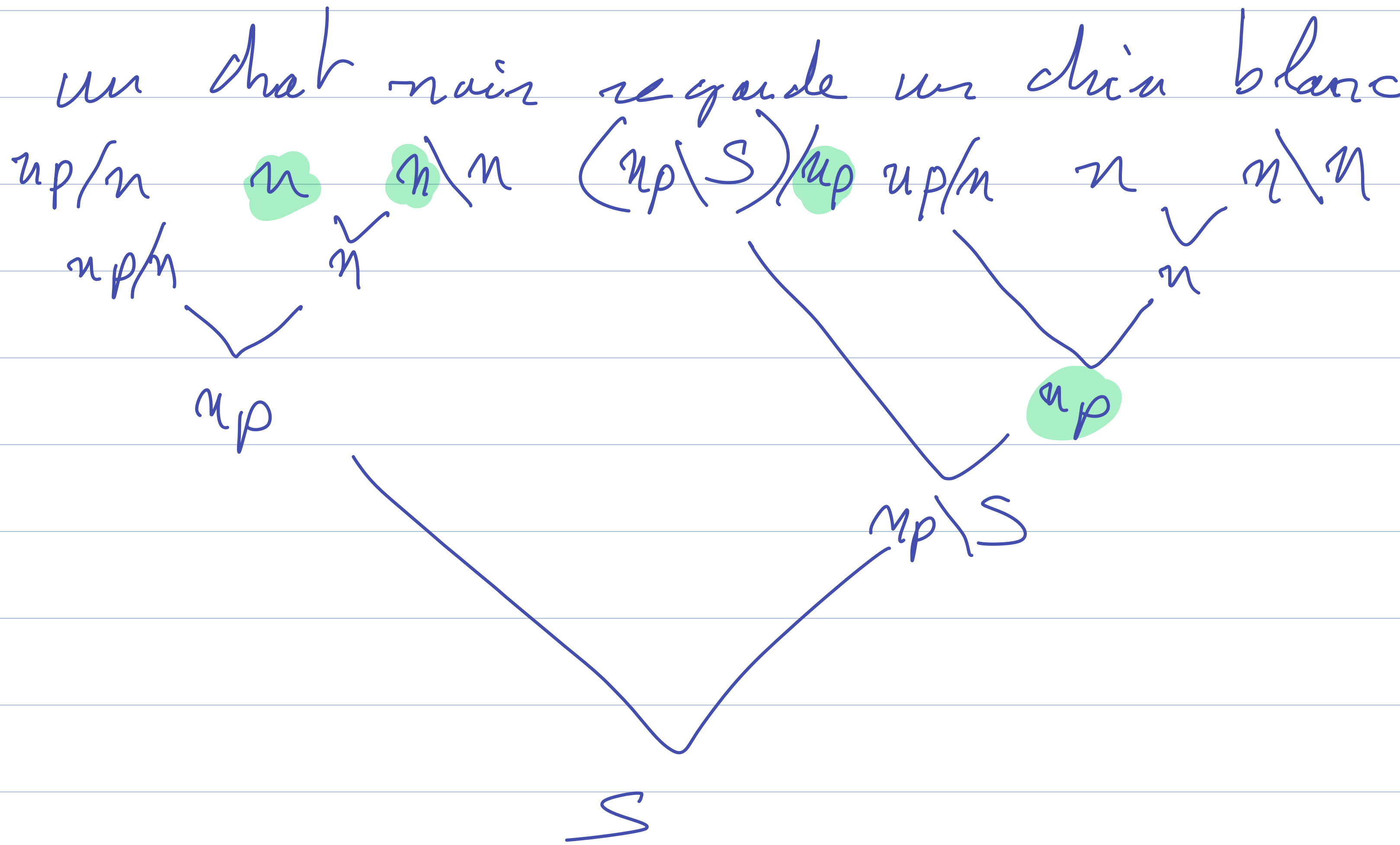
$$B/A \quad A \longrightarrow B$$

$$\cancel{A} \cdot \frac{B}{\cancel{A}} \longrightarrow B$$

$$\frac{B}{\cancel{A}} \cdot \cancel{A} \longrightarrow B$$

blanc, noir	$n\backslash n$
un	np/n
chien, chat	n
court, dort	$np\backslash S$
regarde	$(np\backslash S)/np$
mot	1 ou plusieurs catégories

un	chat	dort
np/n	n	$np\backslash S$
	np	$np\backslash S$
	S	



un chat noir regarde un chien blanc
np/n
n
n\n
(np\S)/np
np/n
n
n\n
np/n
n
n
np
np
np\S
S

un	chat	noir	regarde	un	chien	blanc
np/n	n	$n\backslash n$	$(np\backslash S)/np$	np/n	n	$n\backslash n$
np		$n\backslash n$				

Une grammaire AB est la donnée d'un lexique Lex qui à chaque mot associe une ou plusieurs catégories

(Il y a 2 règles, les mêmes pour toutes les grammaires.)

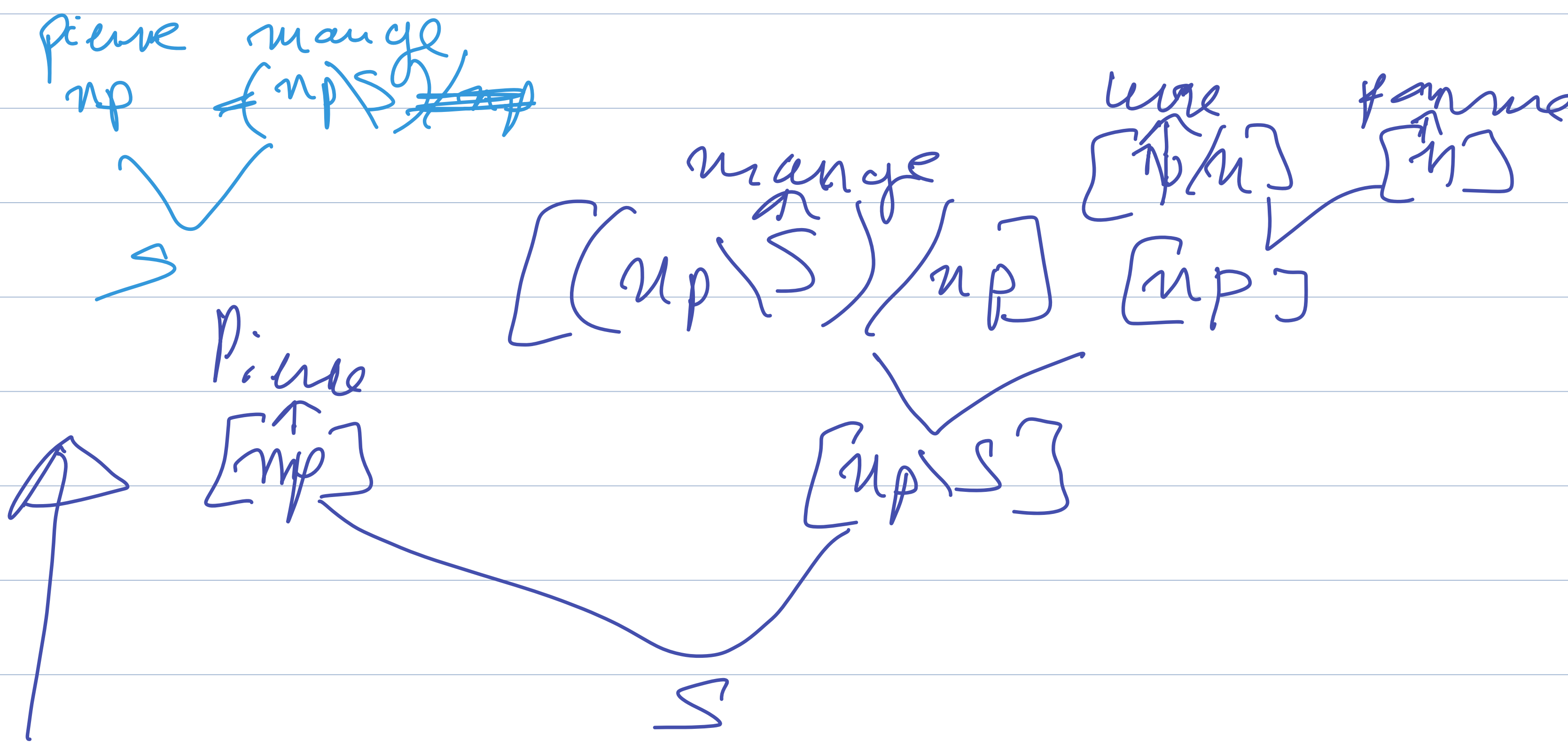
Une suite de mots appartient au langage engendré $\mathcal{L}$ (Lex) ssi il est possible de choisir une catégorie par mot parmi celles que le lexique associe à ce mot de sorte que la suite de catégorie se réduise en S

$m_1 \quad \ldots \quad m_n \in \mathcal{L}(Lex)$

si $\forall i \in \{1, \ldots, n\} \; \exists \, t_i \in Lex(m_i)$

$$t_1 \quad \ldots \quad t_n \longrightarrow S.$$

Pierre	np
mange	$(np\backslash S)/np$, $(np\backslash S)$
une	np/n
pomme	n



mot	catégorie
chat	n
un	np/n
regarde	$(np\backslash S)/np$

CFG

$n \rightarrow$ chat

$[np/n] \rightarrow$ un

$[(np\backslash S)/np] \rightarrow$ regarde

règles $[A]\ [A\backslash B] \rightarrow [B]$

$[B/A]\ [A] \rightarrow [B]$

avec $[B/A]$ $[A\backslash B]$ dans NT

NT: n, $[np/n]$, $[(np\backslash S)/np]$, $[np\backslash S]$, np, S

6 Non terminaux : toutes les catégories et sous catégories présentes dans le lexique

~~$a^n b^n c^n$~~

~~$S \rightarrow abc$~~

~~$S \rightarrow aabbcc$~~

~~$S \rightarrow aaabbbccc$~~

CFG en forme ~~normale~~ de Chomsky

Dans l'autre sens

CFG $\Rightarrow$ CFG $\longrightarrow$ AB-CG

en forme normale de Greibach forte

1 NT

$X \longrightarrow a$ 1 T

$X \longrightarrow a\,Y$ 1 T 1 NT

$X \longrightarrow \underline{a}\,Y\,Z$ 1 T 1 NT 1 NT

On simule la CFG Greibach
par une ABCG ainsi

$X \to a$
$X \to b\,Y$
$X \to c\,Y\,Z$

mot	catégorie
a	X
b	X/Y
c	$(X/Z)/Y$

$$\underbrace{(X/Z)/Y \quad Y}_{(X/Z)} \qquad Z$$

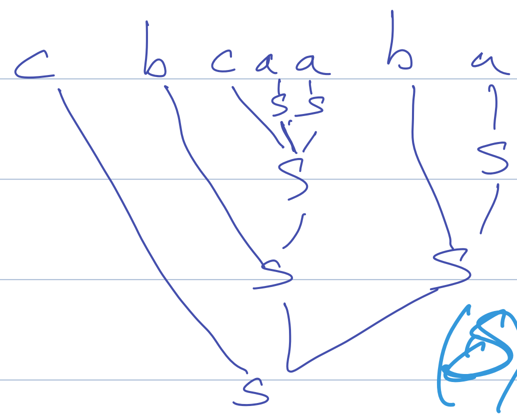
$$\underbrace{(X/Z) \qquad\qquad Z}_{X}$$

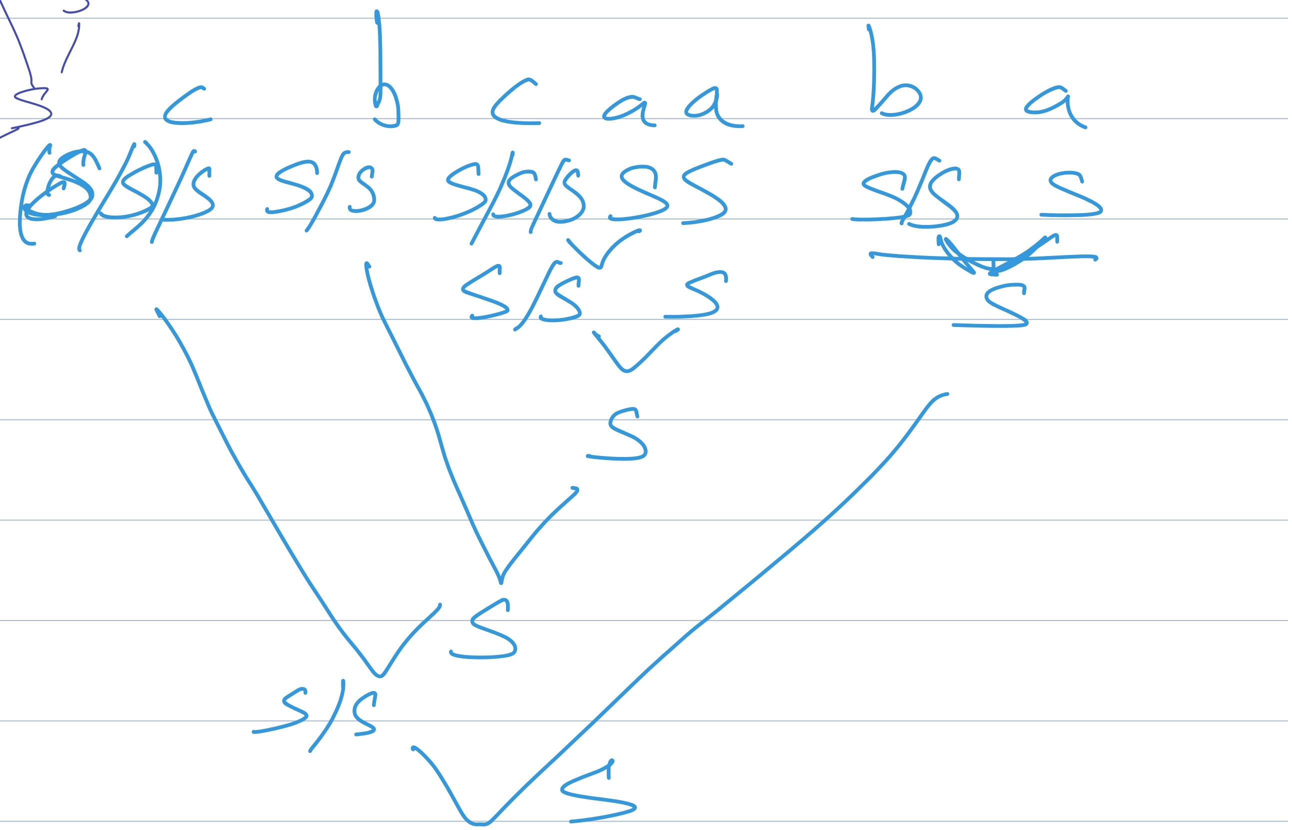
$S \to a$

$S \to b\,S$

$S \to c\,S\,S$

a	S
b	S/S
c	$(S/S)/S$





Grammaires Équivalente au CFG

apprenables à la limite

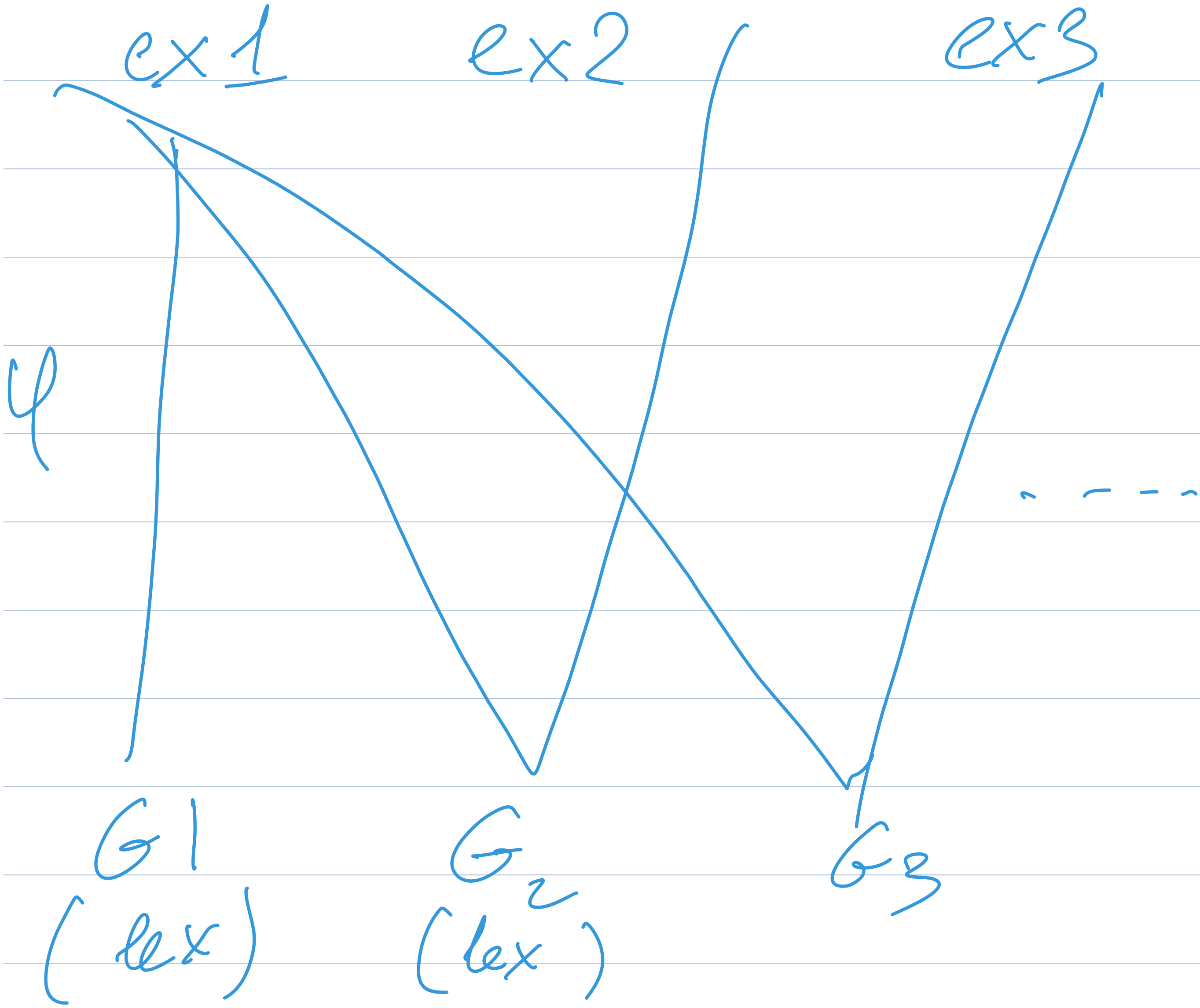
apprentissage exact.

D'abord cadre très contraints

- exemples très structurés
- on apprend une grammaire catégorielle "simple" rigide 1 catégorie par mot

  manger$_1$ $\quad np \backslash S$

  manger$_2$ $\quad (np \backslash S)/np$



ex1
ex2
ex3
φ
G1
(lex)
G2
(lex)
G3

Il converge si quand les exemples

énumère un langage L

produit par une AB CG

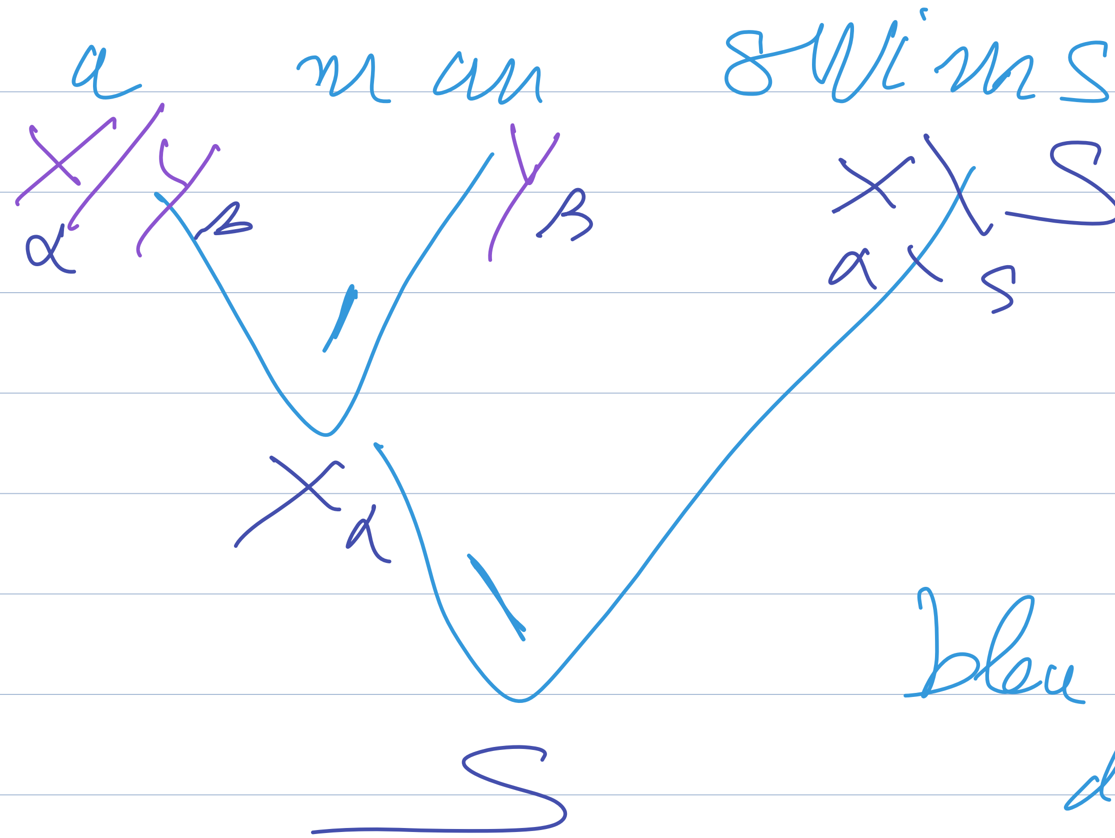
alors il existe N tel que

$\forall p > N \quad G_p = G_N$

$$\mathcal{L}(G_n) = L$$

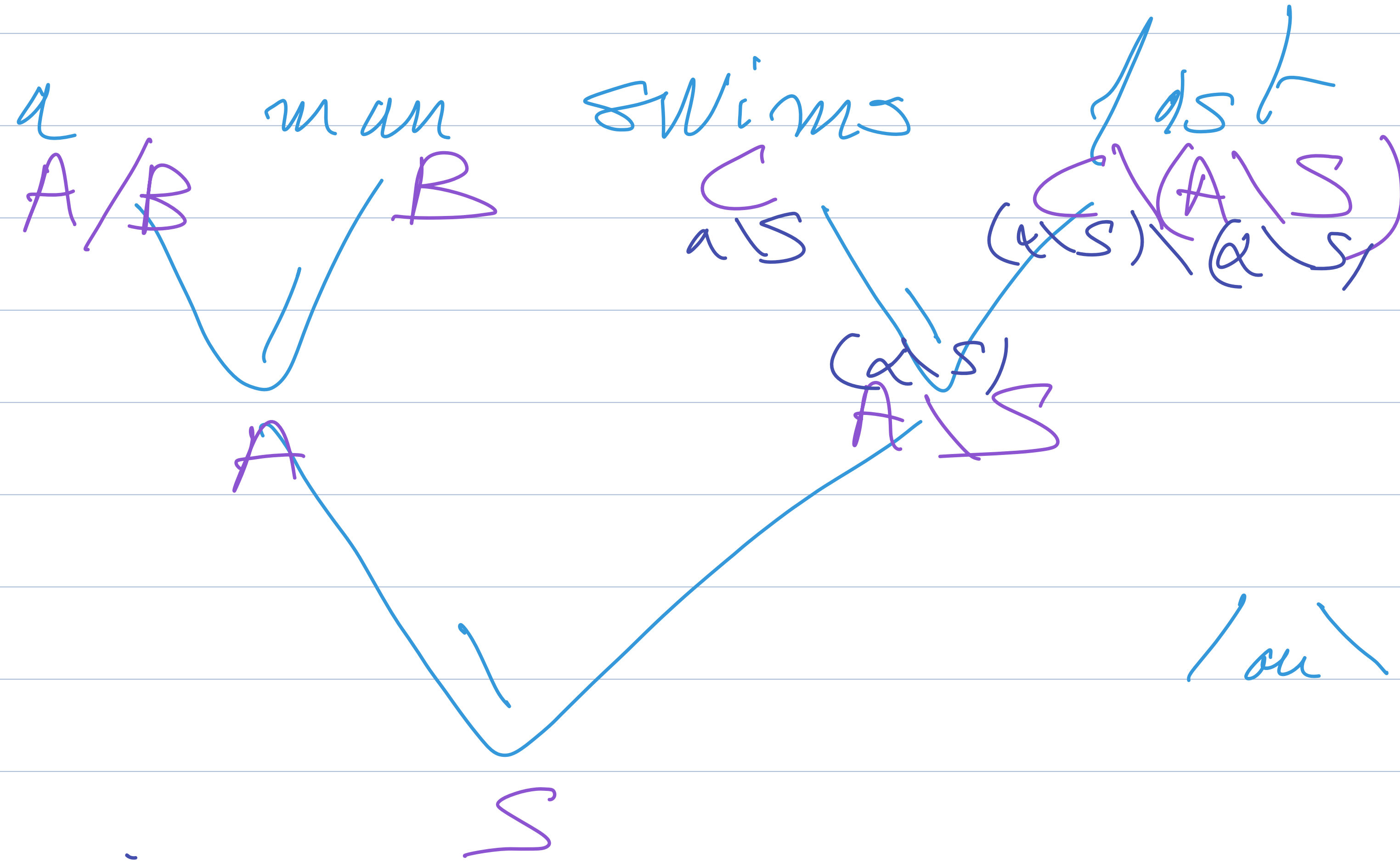
algo . . . . typage ~~unification~~

structure .



bleu clair :
donné

a	X/Y
man	Y
swims	X/S



après unification

$A \longrightarrow \alpha$ $\qquad$ $C \longrightarrow \alpha\backslash S$

$B \longrightarrow \beta$

la structure reste admissible

	ex1	ex2
a	X/Y	A/B
man	Y	B
swims	X\S	C
last		C\(A\S)

il faut limiter pour opérer la coupe
on limite le nbre de categories
par mot : le plus simple 1

			unification
a	X/Y	A/B	α/β
man	Y	B	β
sings	$X\backslash S$	C	$\alpha\backslash S$ $\quad C : \alpha\backslash S$
first		$C\backslash A\backslash S$	$(\alpha\backslash S)\backslash(\alpha\backslash S)$

$$X/Y = /(X, Y)$$

si on arrive à énumérer
les structures produites par
une grammaire AB
l'algorithme converge

si l'unification plante
c'est qu'il n'y a pas de
grammaire AB produisant
toutes les structures données en exemples

$$\overset{X}{(\gamma\backslash(\alpha\backslash\beta))/\gamma} \qquad \overset{X\ \ Y}{(\gamma\backslash(\alpha\backslash\beta))/\gamma\backslash\gamma}$$