LCG 1
LCG 2 (2 sechin)
LCG 3

A Categorial Grammars

A.1. What are categorial grammars?

- A *lexicon* mapping words to (small) sets of formulas
- A *logic* specifying the meaning and the behaviour of the logical connectives

Universal grammar is a logic. Language variation is restricted to the lexicon.

A.2. AB grammars

Not a logic (yet!) but the foundation of categorial grammars.

A.3. Atomic formulas

s (sentence),

np (noun phrase), for example: John, the tall student

n (noun), for example: student, book, ...

Maybe some others: *pp* (for prepositional phrases), *inf* (for infinitival phrases), ...

Goal: all grammatical sentence should be derivable as being of category *s* (in a sense we will make precise).

A.4. Formulas

Formulas are inductively defined as follows.

- Atomic formulas are formulas.
- If A and B are formulas, then (A/B) (we say A *over* B) and $(B\backslash A)$ (we say B *under* A) are formulas.

Intuition: a formula of the form A/B combines with a B to its *right* to form an A, a formula $B\backslash A$ combines with a B to its *left* to form an A.

A.5. Example formulas, example lexicon (strict)

The following are formulas: (np/n), $(np\backslash s)$, $((np\backslash s)/np)$, $((n\backslash n)/(np\backslash s))$

$$
\begin{aligned}
Lex(the) &= \{(np/n)\} \\
Lex(an) &= \{(np/n)\} \\
Lex(president) &= \{n\} \\
Lex(actress) &= \{n\} \\
Lex(likes) &= \{((np\backslash s)/np)\}
\end{aligned}
$$

A.6. Example formulas, example lexicon (sloppy)

The following are formulas: np/n, $np\backslash s$, $(np\backslash s)/np$, $(n\backslash n)/(np\backslash s)$

$$
\begin{aligned}
Lex(the) &= np/n \\
Lex(an) &= np/n \\
Lex(president) &= n \\
Lex(actress) &= n \\
Lex(likes) &= (np\backslash s)/np
\end{aligned}
$$

A.7. AB grammars: rules

$$\frac{A/B \quad B}{A}\,[/E] \qquad \frac{B \quad B\backslash A}{A}\,[\backslash E]$$

A.8. AB grammars: rules

$$\frac{A/B \quad B}{A}\,[/E] \qquad\qquad \frac{\dfrac{the}{np/n} \quad \dfrac{president}{n}}{np}\,[/E]$$

$A = np$, $B = n$

A.9. AB grammars: rules

$$\frac{A/B \quad B}{A}\,[/E] \qquad\qquad \frac{\dfrac{an}{np/n} \quad \dfrac{actress}{n}}{np}\,[/E]$$

$A = np$, $B = n$

A.10. AB grammars: rules

$$\frac{A/B \quad B}{A}\,[/E]$$

$$\frac{\dfrac{\textit{likes}}{(np\backslash s)/np} \quad \dfrac{\dfrac{\textit{an}}{np/n} \quad \dfrac{\textit{actress}}{n}\,[/E]}{np}}{np\backslash s}\,[/E]$$

$A = np\backslash s,\ B = np$

A.11. AB grammars: rules

$$\frac{B \quad B\backslash A}{A}\,[\backslash E]$$

$B = np$, $A = s$

$$\begin{array}{c} \textit{likes an actress} \\ \vdots \\ np\backslash s \end{array}$$

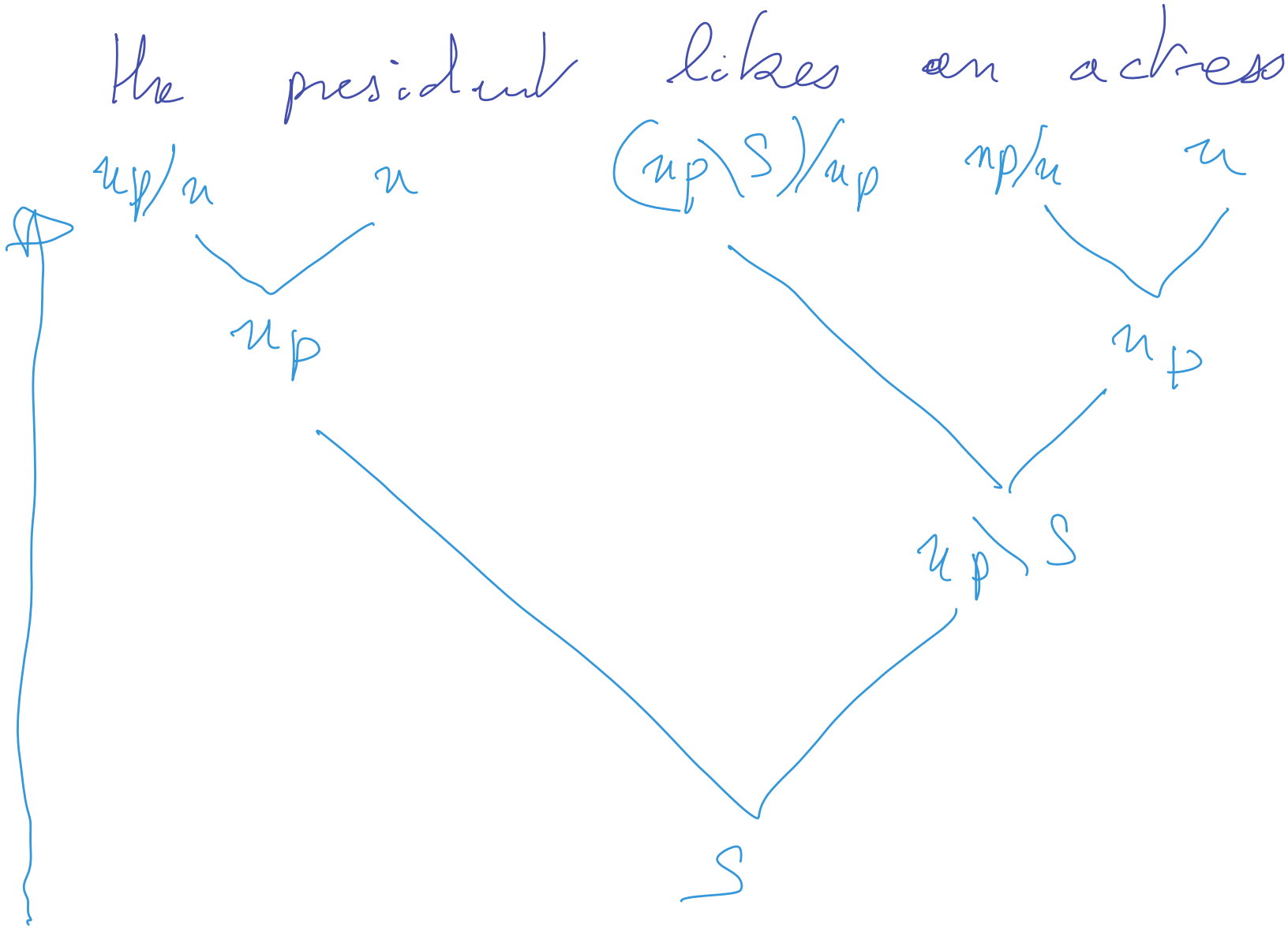
A.12. AB grammars: rules

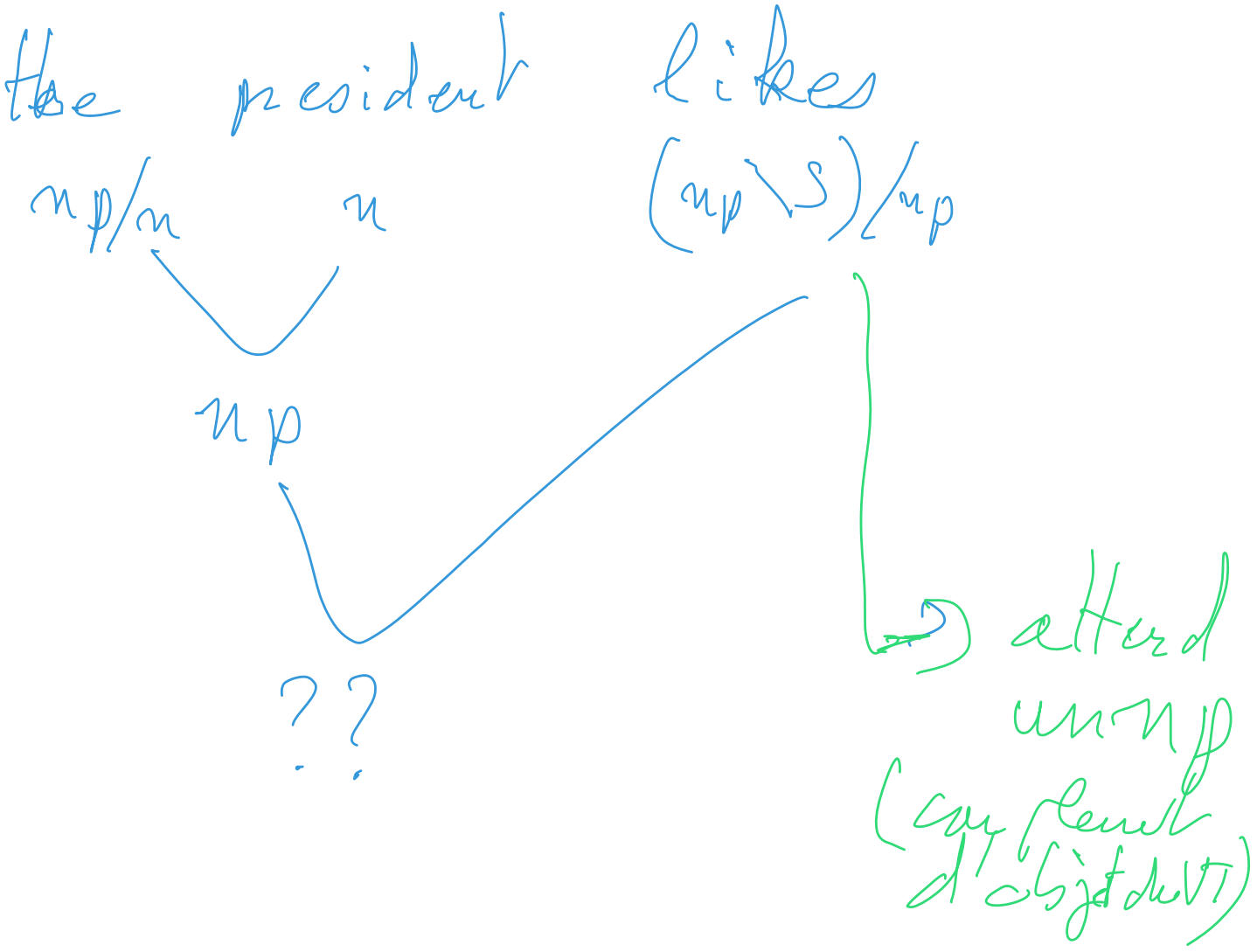
$$\frac{B \quad B\backslash A}{A}\,[\backslash E]$$

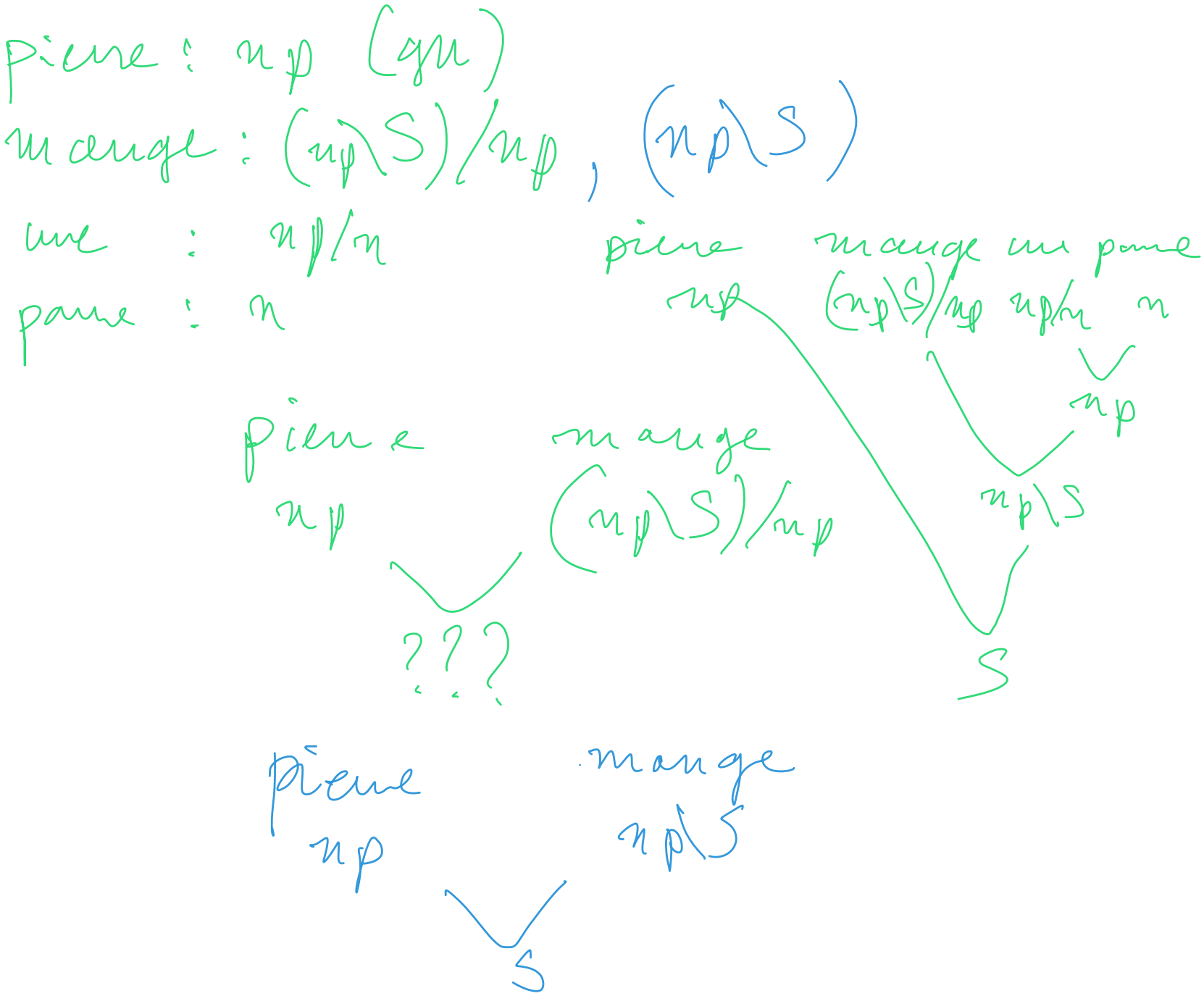
$B = np$, $A = s$

$$\frac{\overset{\vdots}{np} \qquad\qquad \overset{\vdots}{np\backslash s}}{s}\,[\backslash E]$$

the president likes an actress


The presidual likes an actress
u_p/u
u
u_p
$(u_p \setminus S)/u_p$
u_p/u
u
u_p
$u_p \setminus S$
S


the president likes
np/n n
np
??
(np\s)/np
→ attend
un np
(qui semble
d'objet direct)


pierre : np (qu)
mange : (np\S)/np , (np\S)
une : np/n
pomme : n
pierre
np
mange une pomme
(np\S)/np np/n n
np
np\S
S
pierre
np
mange
(np\S)/np
???
pierre
np
mange
np\S
S

une suite de mots est une phrase
si et seulement si pour chaque
mot m_i de la phrase
admet dans le lexique
une catégorie $c_i \in Lex(m_i)$
la $c_1 \ldots c_n \longrightarrow S$

dans la pratique

on a une catégorie par mot

manger_nt (np\S)/np

manger_vi (np\S)

preprocessing rapide (HMM)

---

et : 20 catégorie

est : 15 catégories

A.13. Modifiers

1. A student slept.
2. A student slept in class.
3. A student slept in class during the exam.
4. A student slept in class during the exam yesterday at 15h while snoring.

"*in class*" modifies a sentence s and is therefore assigned the formula $s\backslash s$ (or if you prefer, the vp modifier $(np\backslash s)\backslash(np\backslash s)$).

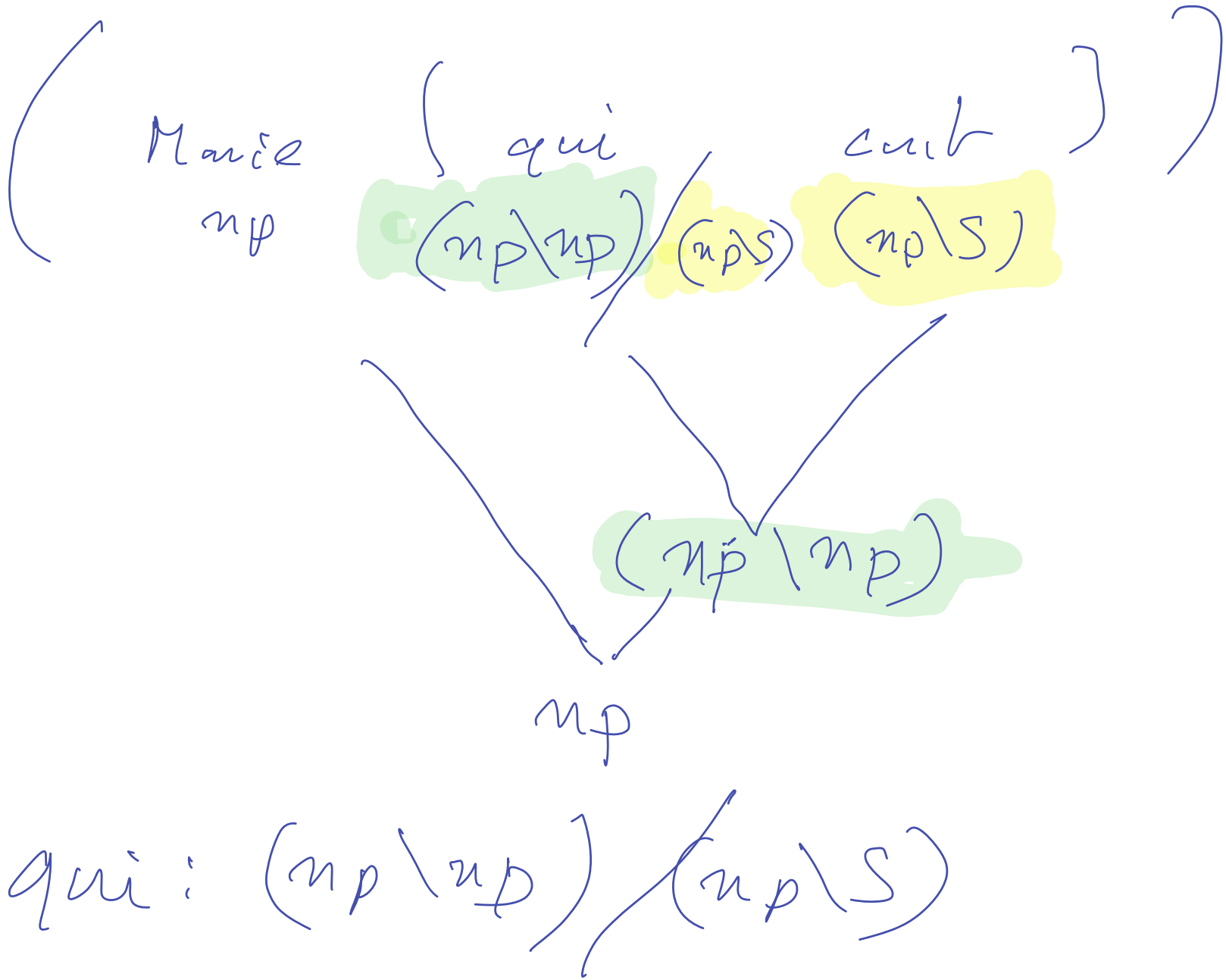
"*class*" is a noun n, therefore a lexical possibility for "*in*" should be $(s\backslash s)/n$ or $((np\backslash s)\backslash(np\backslash s))/n$.

Exercices

Exo Simple : le lexique est donnée
. montrer que w1 w2
est une phrase
n'est pas une phrase

Exo plus compliqué :
. trouver la catégorie syntaxique
de certains

Exo méta : propriétés du système


Marie
np
qui
(np\np)/(np\S)
(np\S)
(np\np)
np
qui : (np\np)/(np\S)

1 ((le chien)(qui aboie)) regarde le chat

2 (le (chien (qui aboie))) regarde le chat

qui : $(np\backslash np)/(np\backslash S)$ 1

quel est l'autre catégorie de qui 2

le chauffeur qui n'a pas bu raccompagnera les invités

plusieurs chauffeurs
"qui n'a pas bu" sert à identifier le chauffeur

le chauffeur, qui n'a pas bu, raccompagnera les invités

un chauffeur
on ajoute la précision qu'il n'a pas bu

Marie qui - - - .

Seulement explicatif