Parsing Lambek calculus using partial composition

Denis Béchet

Denis.Bechet@univ-nantes.fr

LINA, University of Nantes

Introduction

Lambek languages: Context-free languages [Pentus 92]

$\Longrightarrow$ CYK parser : n^3 (n = nb of words)

Lambek calculus: NP-complete [Pentus 03]

$\Longrightarrow$ proof search : C^n (n = nb of primitive types)

Parsing of “whom have you seen ?”

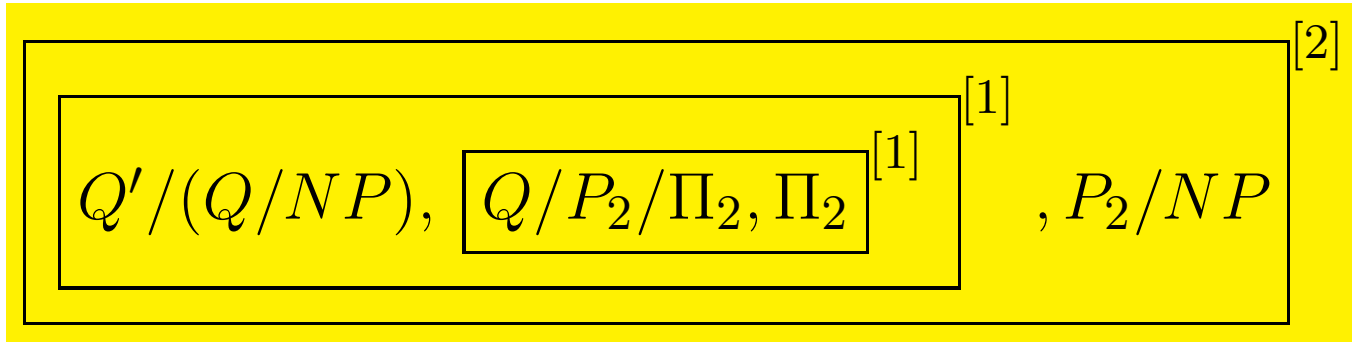
whom	have	you	seen
$Q'/(Q/NP)$	$Q/P_2/\Pi_2$	Π_2	P_2/NP

Motivation: Lambek calculus with a bounded size on types is n^3 (n = nb of primitive types)

Parsing with word separation

Parsing of “whom have you seen ?”

whom	have	you	seen
$Q'/(Q/NP)$	$Q/P_2/\Pi_2$	Π_2	P_2/NP



We use:

- Flat Interfaces of Modules of Proof-Nets for partial compositions
- Outerplanar graph properties to find a CYK algorithm

PLAN

- Introduction
- Background
  - Lambek calculus
  - Module and proof net
  - Interface calculus
- Parsing with word separation
  - Rewriting
  - Partial composition
  - Majority partial composition
- Conclusion

Lambek calculus

Categorial Grammars

Each word is associated to one or more types:

- Primitive types Pr:
  - N for common nouns “cat”, “mice”
  - NP for noun phrases or names “John”, “Mary”
  - S for correct sentences
- Composed types $Tp ::= Pr \mid Tp/Tp \mid Tp\backslash Tp \mid Tp \bullet Tp$:
  - NP/N for articles “a”, “the”, “this”
  - $NP\backslash S$ for intransitive verbs (3rd person) “sleeps”
  - $(NP\backslash S)/NP$ for transitive verbs (3rd person) “eats”
  - $S/(NP\backslash S)$ for pronouns “he”, “she” and “it”

Notation $[word \mapsto type]$: $[a \mapsto NP/N]$

Lambek calculus

Derivation rules for Lambek calculus $\Gamma \vdash A$

$$\frac{\Gamma, A, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, \Delta, \Gamma' \vdash C}\ \mathbf{Cut} \qquad \frac{}{A \vdash A}\ \mathbf{Ax}$$

$$\frac{\Gamma \vdash A \quad \Delta, B, \Delta' \vdash C}{\Delta, B/A, \Gamma, \Delta' \vdash C}\ /L \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash B/A}\ /R$$

$$\frac{\Gamma \vdash A \quad \Delta, B, \Delta' \vdash C}{\Delta, \Gamma, A\backslash B, \Delta' \vdash C}\ \backslash L \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ \backslash R$$

$$\frac{\Delta, A, B, \Delta' \vdash C}{\Delta, A \bullet B, \Delta' \vdash C}\ \bullet L \qquad \frac{\Gamma \vdash A \quad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \bullet B}\ \bullet R$$

Lambek calculus

“a cat” is a noun phrase

- We have: $[a \mapsto NP/N]$ and $[cat \mapsto N]$
- Let us prove that $[a \mapsto NP/N], [cat \mapsto N] \vdash NP$:

$$\dfrac{\dfrac{}{N \vdash N}\,\mathbf{Ax} \quad \dfrac{}{NP \vdash NP}\,\mathbf{Ax}}{NP/N, N \vdash NP}\,/L$$

Lambek calculus

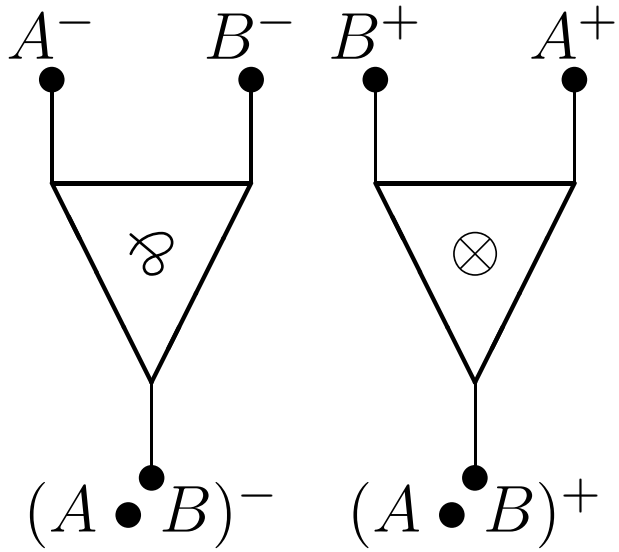
"he eats" is equivalent to S/NP

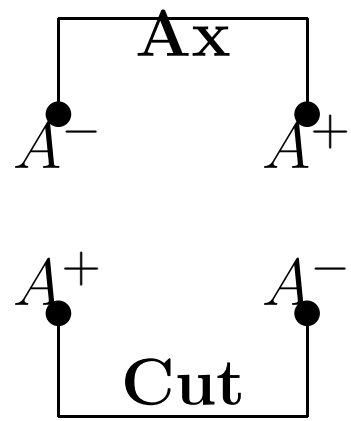
- We have: $[he \mapsto S/(NP\backslash S)]$ and $[eats \mapsto (NP\backslash S)/NP]$
- Let us prove that $S/(NP\backslash S), (NP\backslash S)/NP \vdash S/NP$:

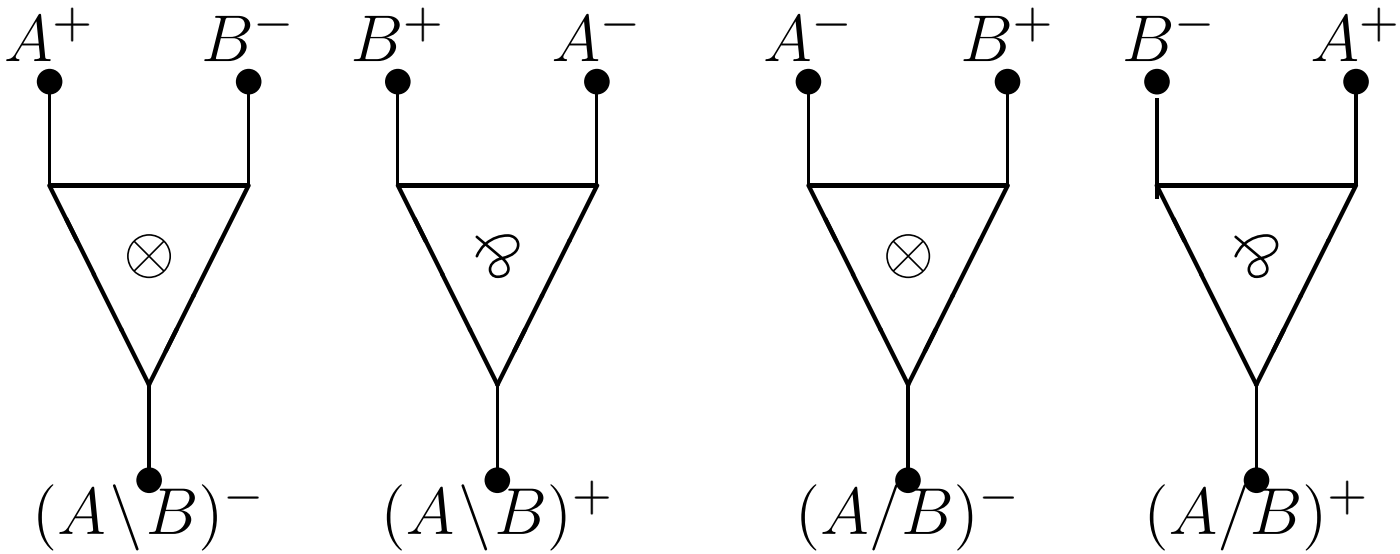
$$\dfrac{\dfrac{\dfrac{\dfrac{}{S \vdash S}\,\mathbf{Ax} \quad \dfrac{\dfrac{}{NP\backslash S \vdash NP\backslash S}\,\mathbf{Ax} \quad \dfrac{}{NP \vdash NP}\,\mathbf{Ax}}{S/(NP\backslash S), NP\backslash S \vdash S}\,/L}{S/(NP\backslash S), (NP\backslash S)/NP \vdash S}\,/L}{S/(NP\backslash S), (NP\backslash S)/NP \vdash S/NP}\,/R$$

Proof-nets

Links

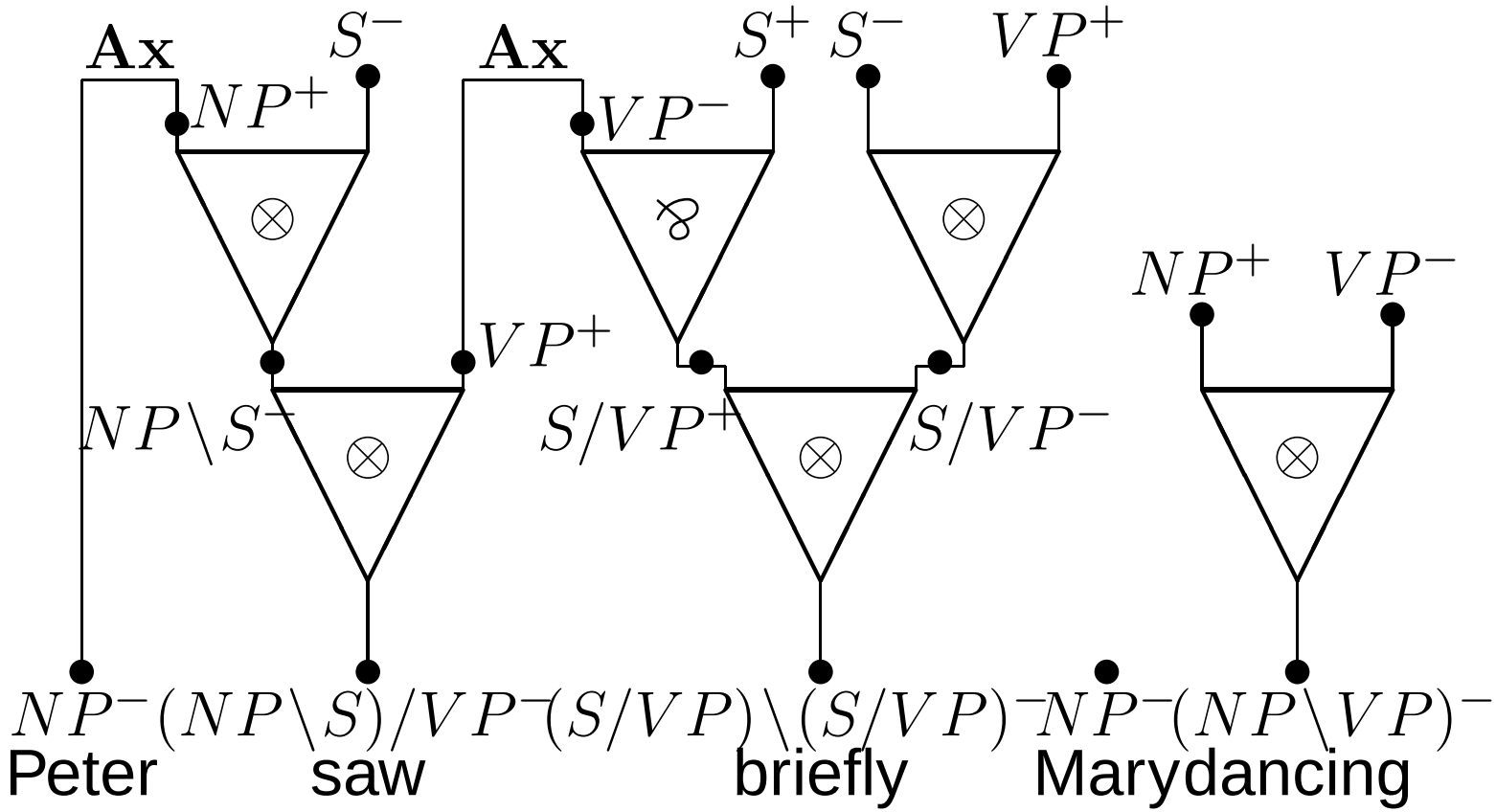






Proof-nets

Module

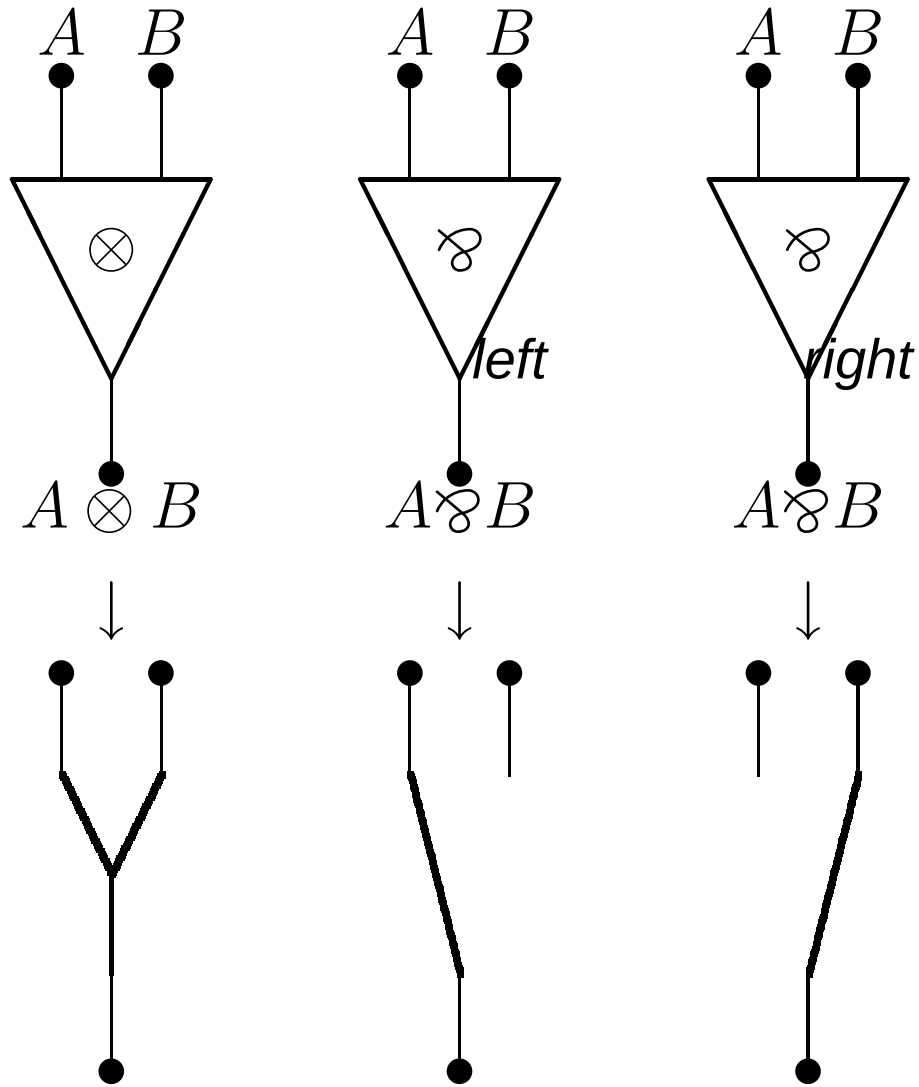


Proof-net: module that corresponds to a sequential proof in Lambek calculus.

Remark: Not every module without hypothesis is a proof-net $\Longrightarrow$ We need a **correctness criterion**

Proof-nets

Switching links

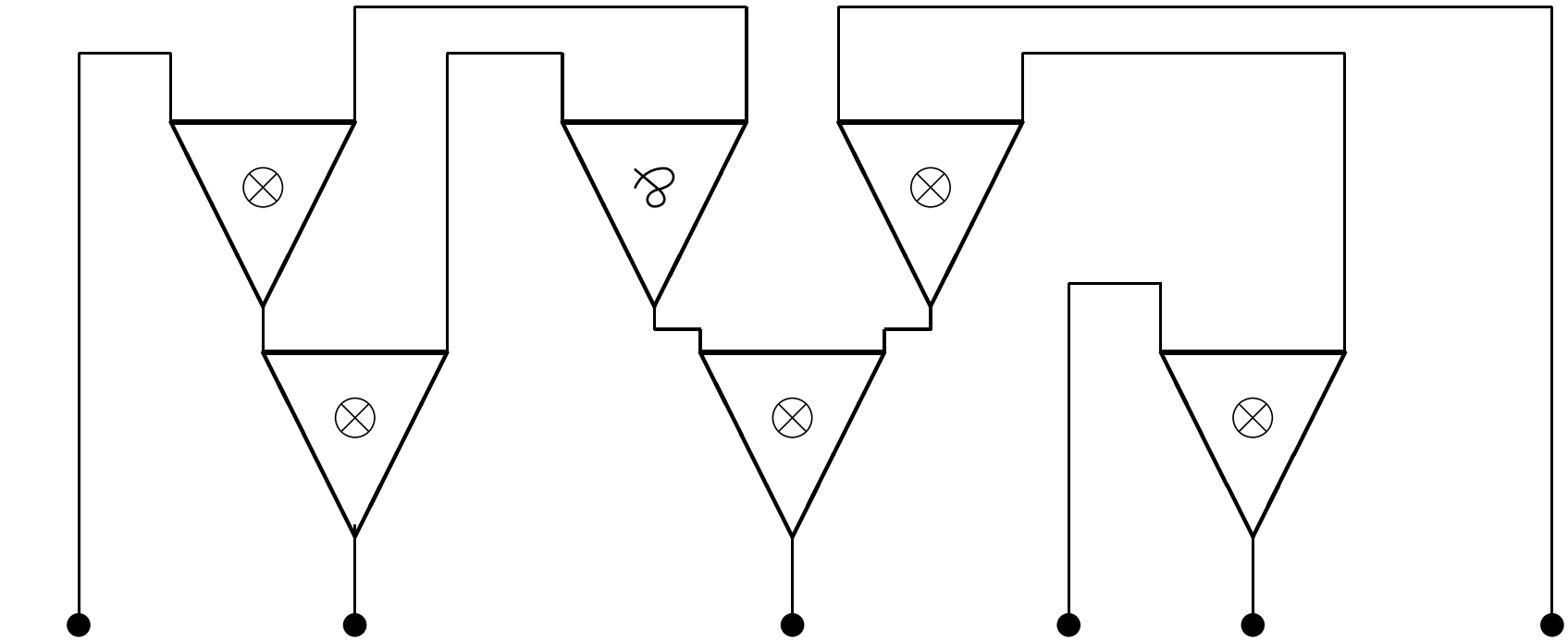


Proof-nets

Switching links

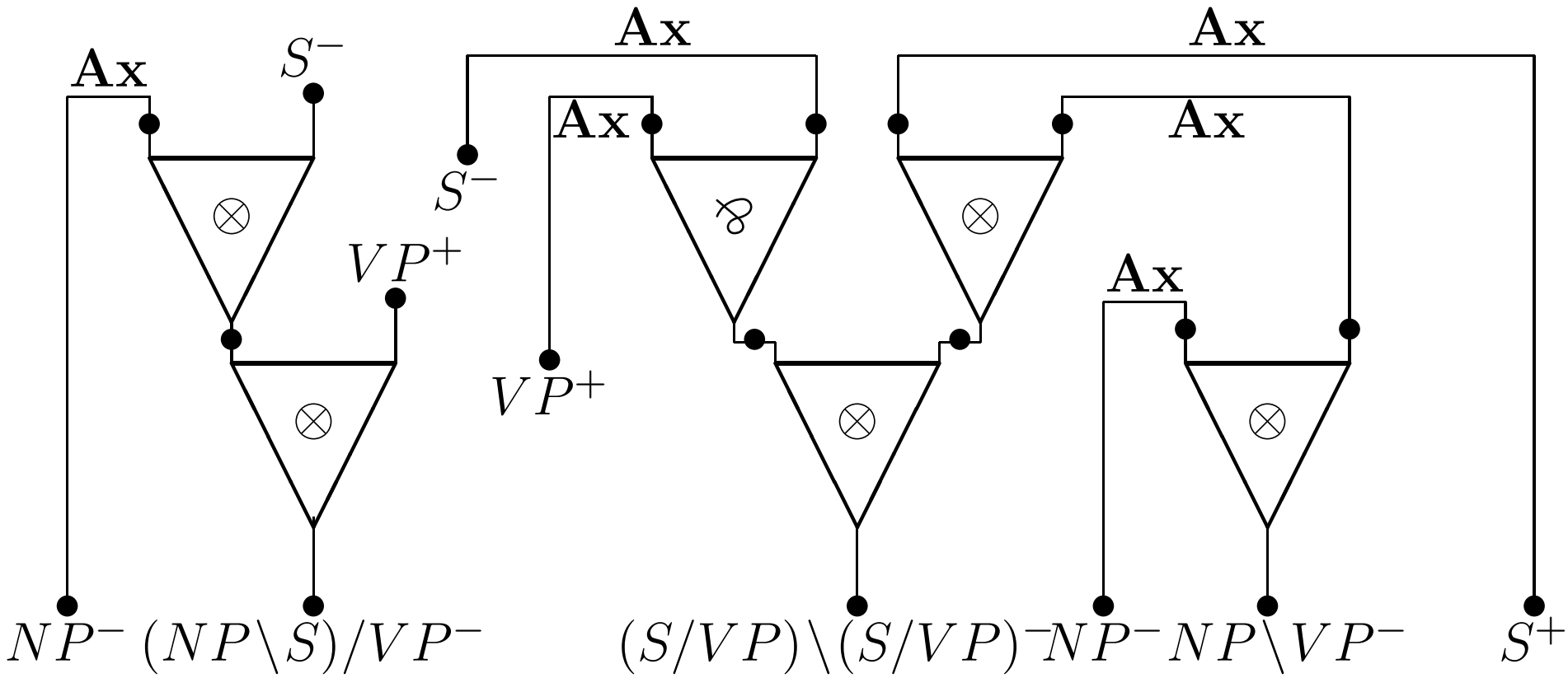
Theorem [Girard, Danos&Regnier, Roorda]. A planar module without hypothesis is a proof-net iff for every switching, the graph is acyclic and connected.

Proof-nets



A proof-net corresponding to “Peter saw briefly Mary dancing”

Modules and interfaces

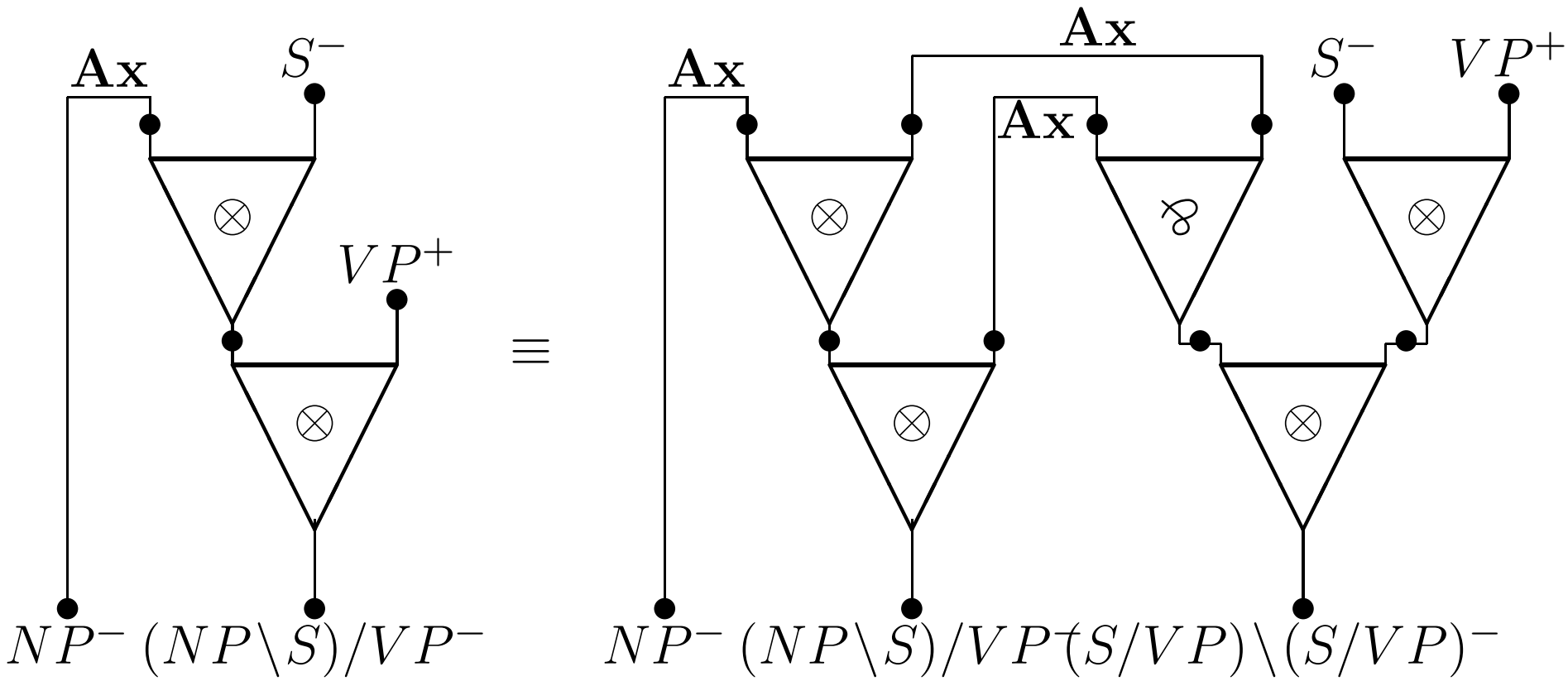


Two orthogonal modules = a splitting of a proof-net in two parts

Equivalent modules:

$M_1 \equiv M_2$ iff they have the same orthogonal modules

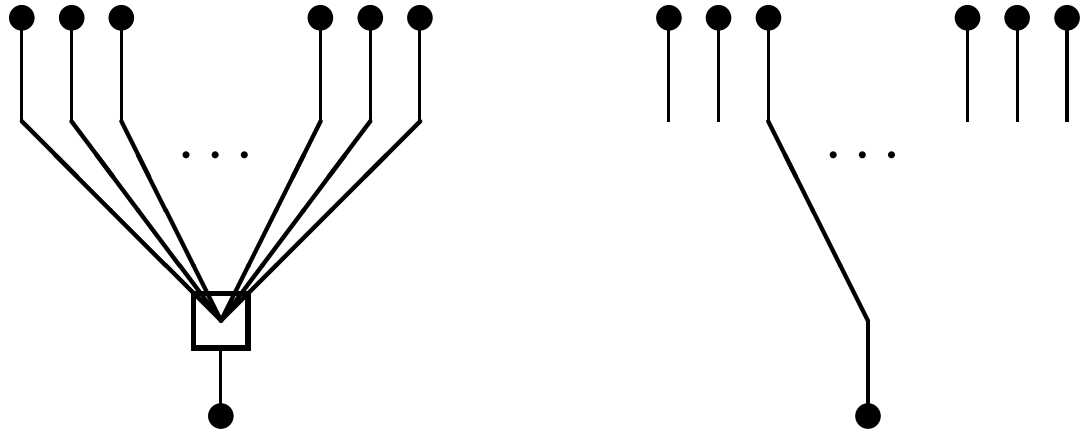
Modules and interfaces



Two equivalent modules

Modules and interfaces

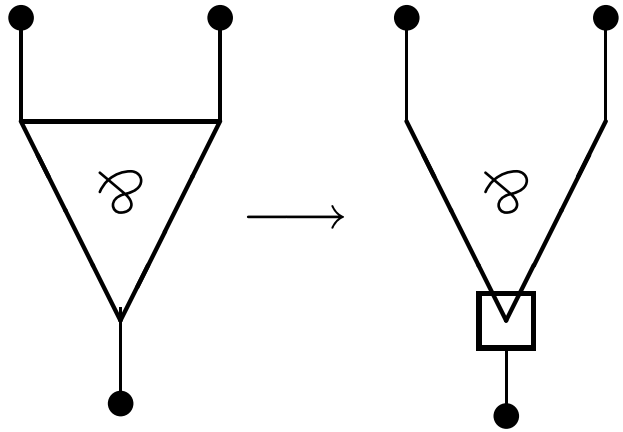
Interface links

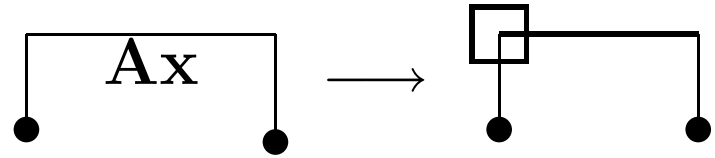


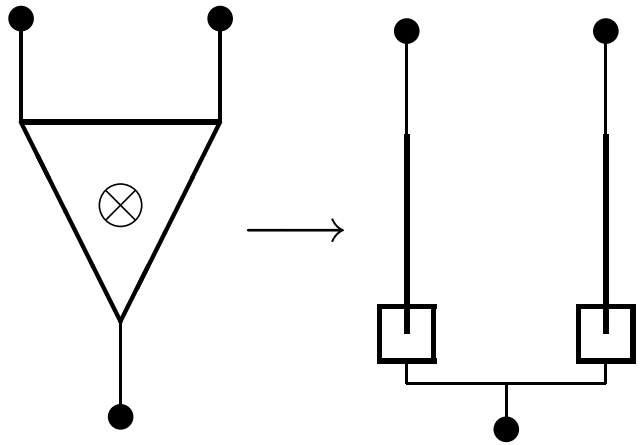
k-ary $\parr$-link and one of its switching positions

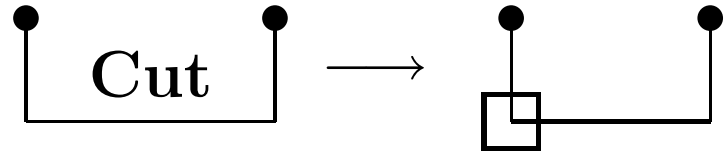
Modules and interfaces

From module links to interface



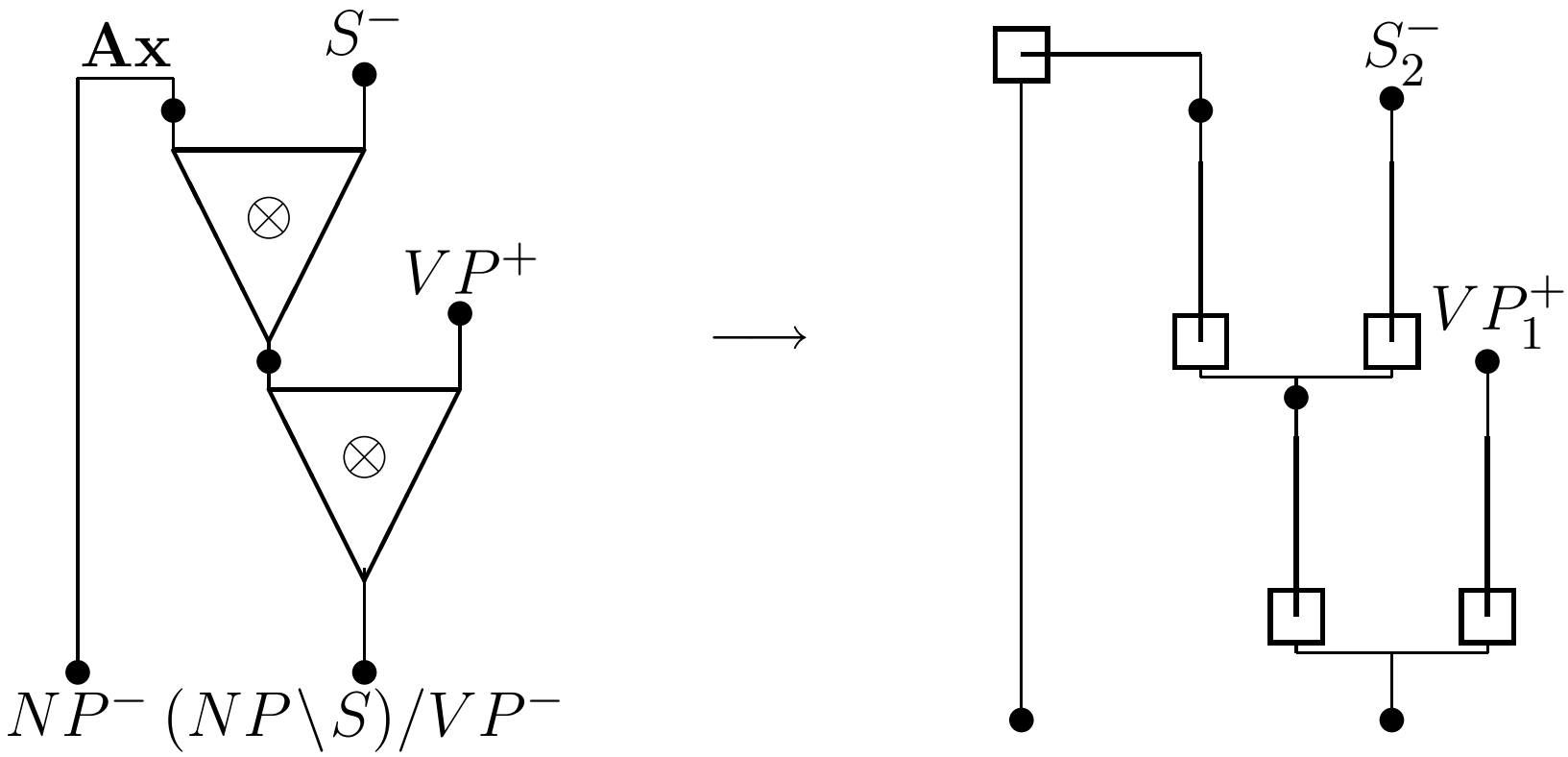






Modules and interfaces

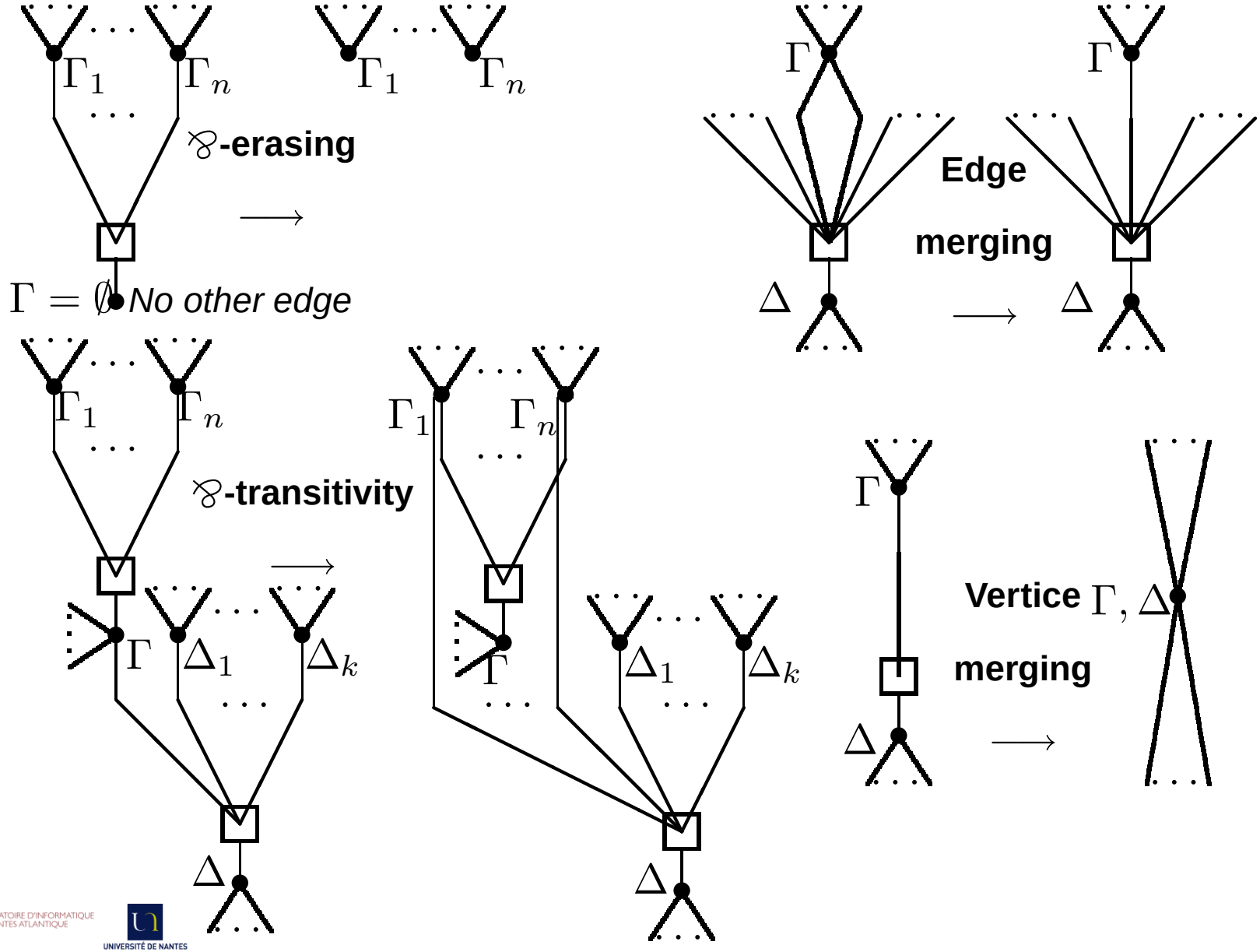
From module to interface



The natural interface of a module

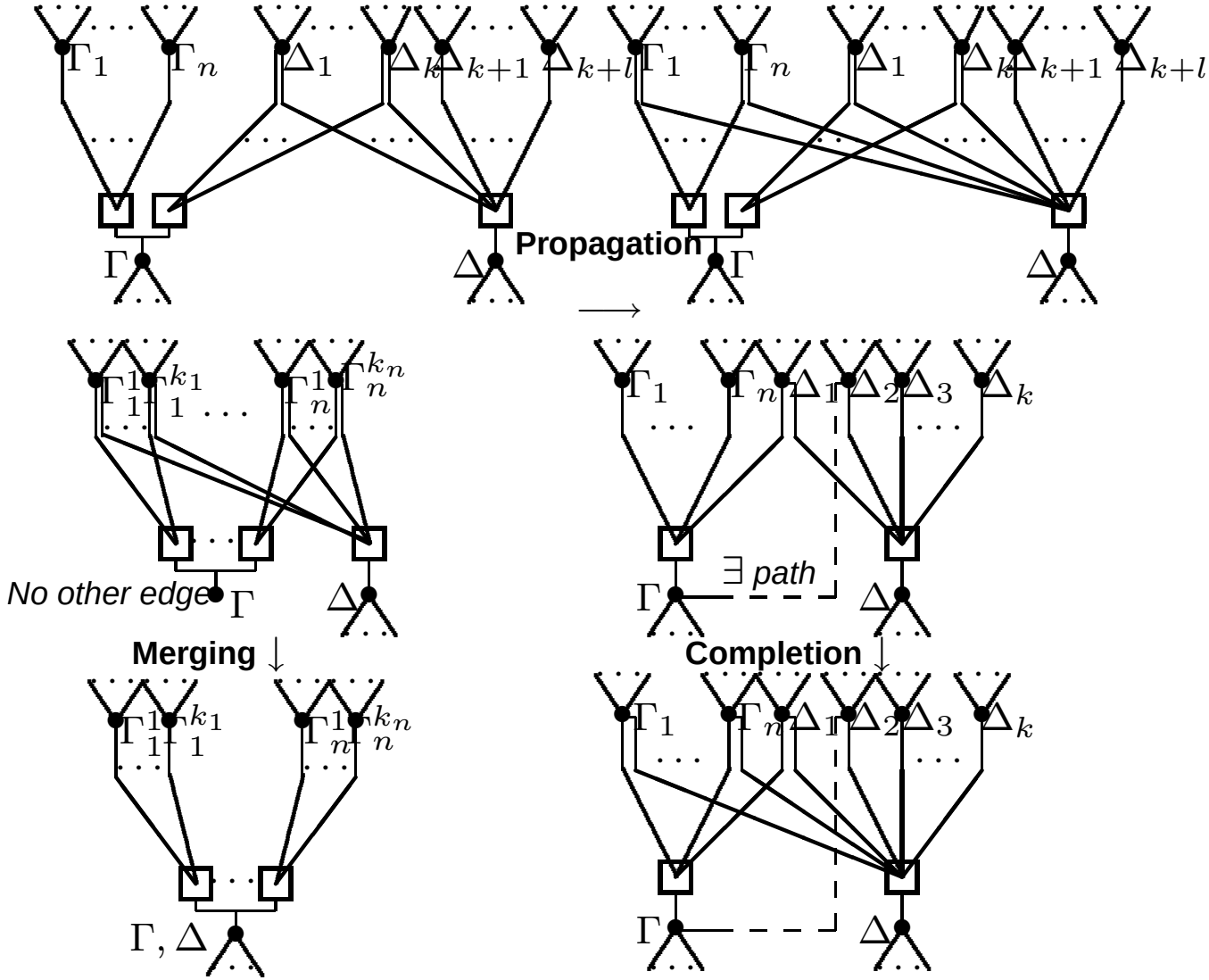
Interface normalization

From interfaces to flat interfaces



Interface normalization

Flat interface normalization



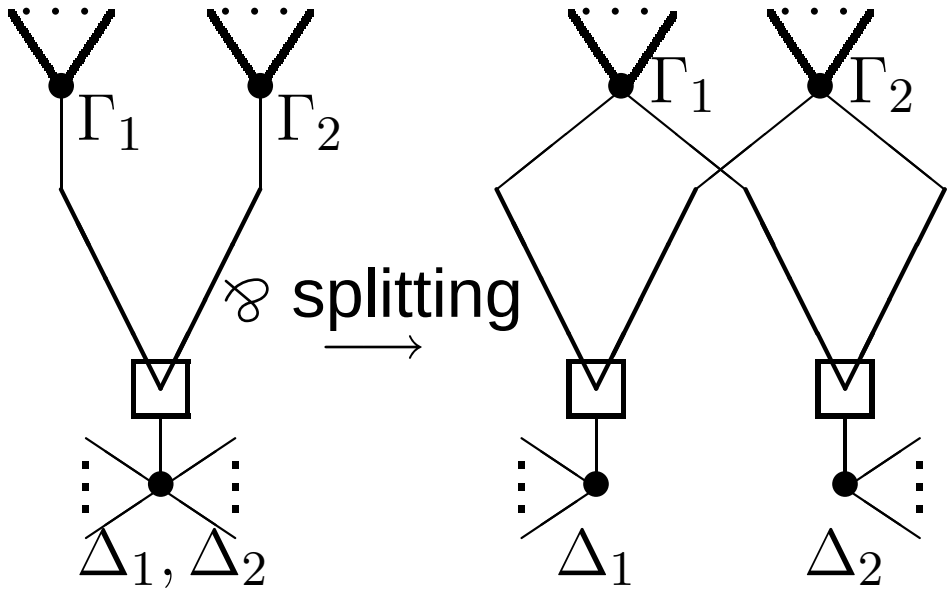
Theorems

Theorem: Normalisation is confluent and terminating

Theorem: Two modules are equivalent iff their normalized flat interfaces are equal

Remark: There exists only a finite number of (normalized) flat interfaces corresponding to a given frontier

Key transformation



Flat interface calculus

2 operations:

- $I|J$: I and J are juxtaposed
- for $I[\ldots, A, A^{\perp}, \ldots]$, if the addition of an axiom on A and $A^{\perp}$ gives a correct interface, we note $I[\ldots, \overline{\bullet, \bullet}, \ldots]$ the flat interface given after normalization

Parsing using rewriting

$v_1 \cdots v_n \in \mathcal{L}(G)$ iff for $1 \leq i \leq n$, $\exists I_i \in I(v_i)$ such that

$$I_1 | \cdots | I_n \xrightarrow{(1)^*} S$$

$\xrightarrow{(1)^*}$: the reflexive and transitive closure of $\xrightarrow{(1)}$:

$$I[A_1, \ldots, A_{i-1}, B, B^\perp, A_{i+2}, \ldots, A_n] \xrightarrow{(1)}$$
$$\xrightarrow{(1)} I[A_1, \ldots, A_{i-1}, \overbrace{\bullet, \bullet}, A_{i+2}, \ldots, A_n]$$

Parsing with word separation (list)

Three rewriting rules ($\Gamma, \Delta \in \mathcal{I}^*$, $I, J \in \mathcal{I}$, $A, B, A_i \in Pr$):

- $[\mathrm{M}]$ **(merge)**: $\Gamma, I, J, \Delta \xrightarrow{M} \Gamma, I|J, \Delta$.
- $[\mathrm{I}\,]$ **(internal)**:

  $\Gamma, I[\ldots, A, A^{\perp}, \ldots], \Delta \xrightarrow{I} \Gamma, I[\ldots, \overbrace{\bullet, \bullet}, \ldots], \Delta,$
- $[\mathrm{C}_k\,]$ **(k-partial composition)**:

  $\Gamma, I[\ldots, A_k, \ldots, A_1], J[A_1^{\perp}, \ldots, A_k^{\perp}, \ldots], \Delta \xrightarrow{C_k}$

  $\Gamma, I|J[\ldots, \overbrace{\bullet, \ldots, \bullet, \bullet, \ldots, \bullet}, \ldots], \Delta,$

Remark: merge = 0-partial composition

Parsing with word separation

Lemma: Parsing can be done using $\overset{(MIC_k)^*}{\longrightarrow}$:

$v_1 \cdots v_n \in \mathcal{L}(G)$ iff for $1 \leq i \leq n$, $\exists I_i \in I(v_i)$ such that

$$I_1, \cdots, I_n \overset{(MIC_k)^*}{\longrightarrow} S$$

Internal before Merge/Composition

Lemma: $\xrightarrow{I}$ can be performed before $\xrightarrow{M}$ and $\xrightarrow{C_k}$:

$v_1 \cdots v_n \in \mathcal{L}(G)$ iff
for $1 \leq i \leq n$, $\exists I_i \in I(v_i)$, $\exists Y_i \in \mathcal{I}$ such that:

$$\begin{cases} \text{for } 1 \leq i \leq n, I_i \xrightarrow{I^*} J_i \\ J_1, \cdots, J_n \xrightarrow{(MC_k)^*} S \end{cases}$$

Remark: Partial composition does not give a polynomial parsing algorithm because the result of partial composition is not bounded by the lexicon:
$\Longrightarrow$ We need a restricted partial composition

Majority partial composition

A partial composition $\xrightarrow{C_k}$ is a majority partial composition (noted $\xrightarrow{@}$ or $\xrightarrow{@_k}$) if the width of the result is less or equal to the maximum of the widths of the arguments

A non majoritory partial composition:

$$\Gamma, I[Q', O, Q^{\perp}], J[Q, P_2^{\perp}, \pi_2^{\perp}], \Delta \xrightarrow{C_1} \Gamma, I[Q', O, \bullet], J[\bullet, P_2^{\perp}, \pi_2^{\perp}], \Delta$$

A majoritory partial composition:

$$\Gamma, I[Q', O, Q^{\perp}], J[Q, O^{\perp}, \pi_2^{\perp}], \Delta$$

$$\xrightarrow{@_2} \Gamma, I[Q', \bullet, \bullet], J[\bullet, \bullet, \pi_2^{\perp}], \Delta$$

Parsing using majority composition

Main theorem:

$v_1 \cdots v_n \in \mathcal{L}(G)$ iff for $1 \leq i \leq n$, $\exists I_i \in \mathcal{R}_{I^*}(I)(v_i)$ such that:

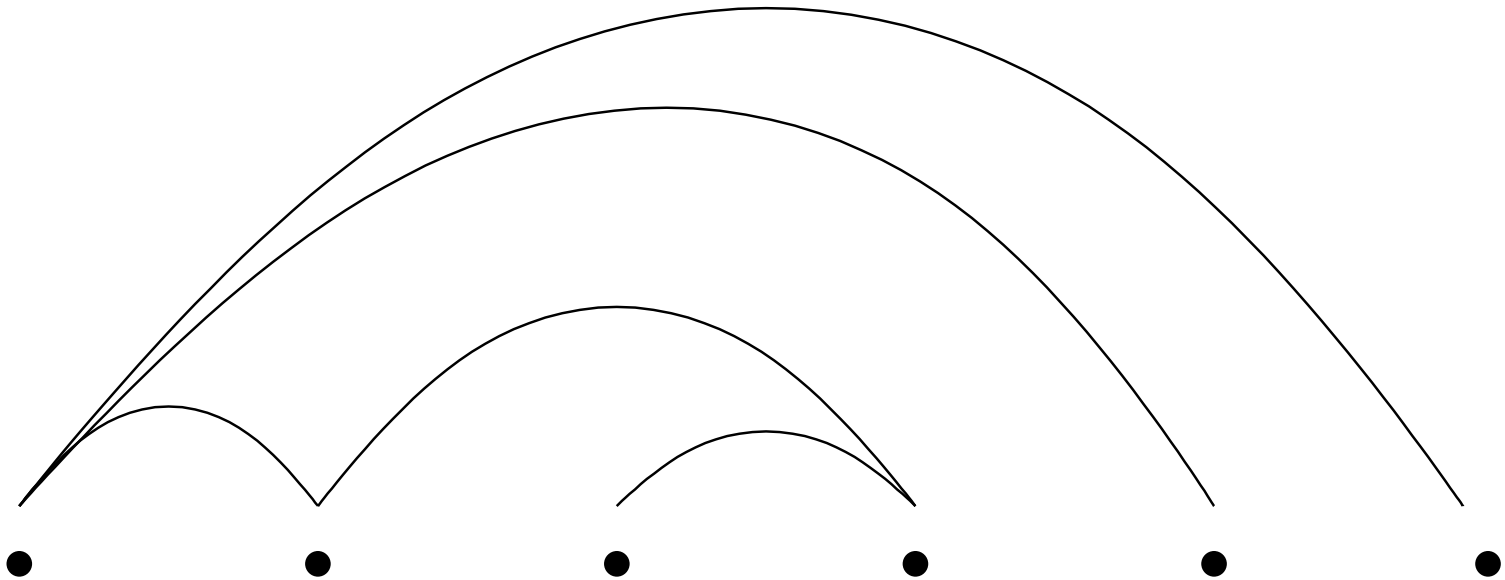
$$I_1, \cdots, I_n \xrightarrow{@^*} S$$

where $\mathcal{R}_{I^*}(I)$ is the completion of I by $\xrightarrow{I^*}$

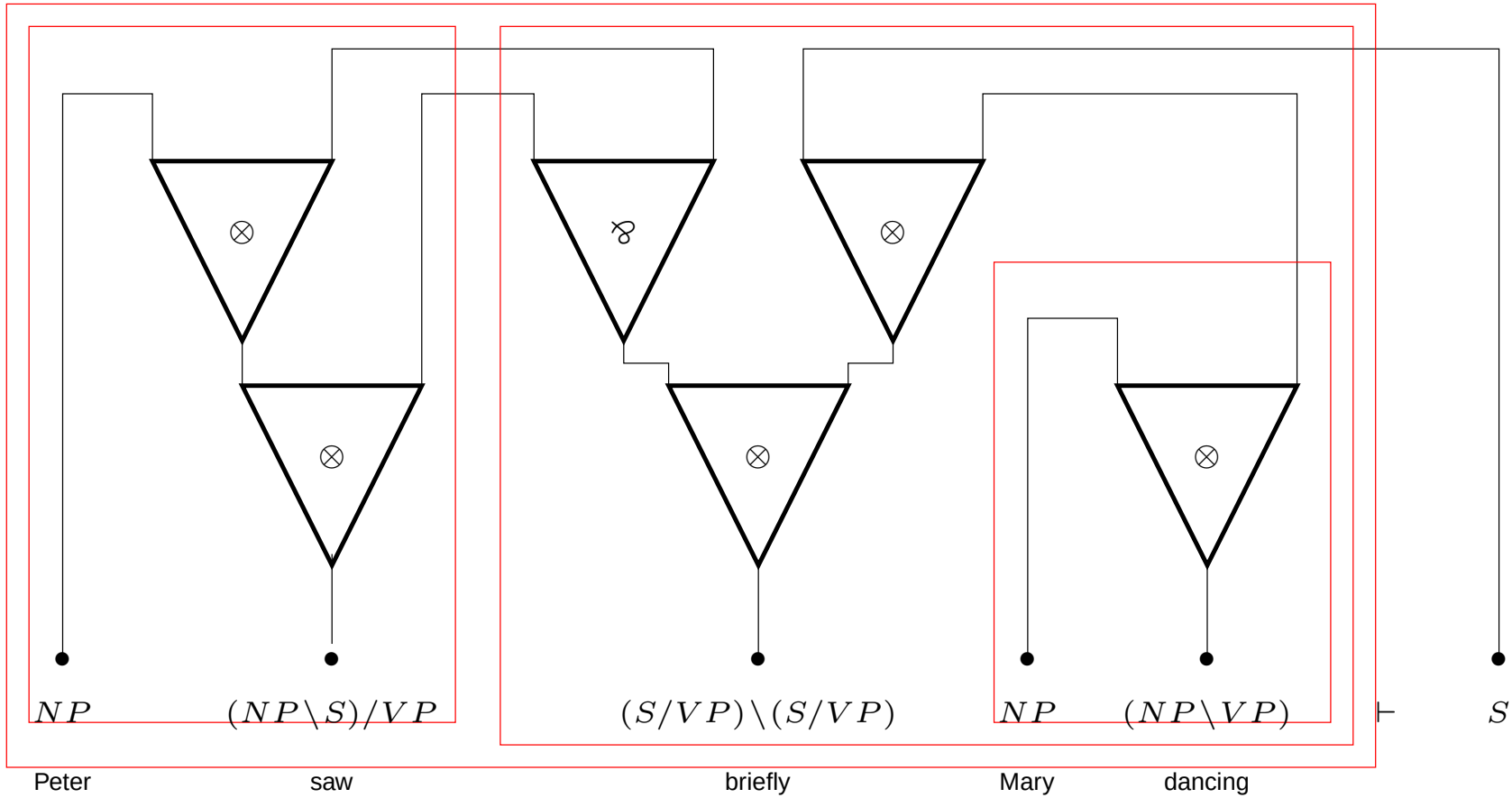
Proof: Property of outerplanar graphs applied to planar modules. There exists a conclusion in Γ that is only linked to its immediate neighbour(s)

Outerplannar graph

Lemma: In an outerplanar graph, there exists a vertex that is only connected to its immediate neighour(s)



Proof-nets


NP
$(NP\backslash S)/VP$
$(S/VP)\backslash(S/VP)$
NP
$(NP\backslash VP)$
$\vdash$
S
Peter
saw
briefly
Mary
dancing

Polynomial parsing using @

Partial composition gives a polynomial parsing algorithm because the size of the result of majority partial composition is bounded by the maximum width of the types of the lexicon.

For a grammar G and a list of words $v_1, \cdots, v_n \in \Sigma^+$, we compute for $1 \leq i \leq j \leq n$, $\mathcal{I}^G_{v_1,\cdots,v_n}(i,j) \subset \mathcal{I}$, the possible interfaces associated to the sublist of words $v_i, \cdots, v_j$ using majority partial composition:

$$i = j : \quad \mathcal{I}^G_{v_1,\cdots,v_n}(i,j) = \mathcal{R}_{I^*}(I)(v_i)$$

$$i < j : \quad \mathcal{I}^G_{v_1,\cdots,v_n}(i,j) = \bigcup_{k=i}^{j-1} \left\{ Z \;\middle|\; \begin{array}{c} \exists X \in \mathcal{I}^G_{v_1,\cdots,v_n}(i,k) \\ \exists Y \in \mathcal{I}^G_{v_1,\cdots,v_n}(k+1,j) \\ X, Y \xrightarrow{@} Z \end{array} \right\}$$

Polynomial parsing using @

$v_1, \cdots, v_n \in \mathcal{L}(G)$ iff $S \in \mathcal{I}^G_{v_1,\cdots,v_n}(1, n)$

Algorithm:

1. Search the flat interfaces associated by G to each word
2. Add the flat interfaces deduced by $\xrightarrow{I^*}$ (internal axioms)
3. Compute recursively the possible types associated to a contiguous segment of words of the string using $\xrightarrow{@}$
4. Look at $\bullet[S]$ in the final set of flat interfaces

Conclusion

- Polynomial parsing algorithm using majority partial composition of the flat interfaces associated to the words of a string
- Need a completion of the lexicon with types deduced using “internal” rewriting