

Combinatory Categorical Grammar (CCG)

Categories

- Categories = types
 - Primitive categories
 - N, NP, S, PP, etc
 - Functions
 - a combination of primitive categories
 - S/NP, (S/NP) / (S/NP), etc
 - V, VP, Adverb, PP, etc

Combinatory Rules

➔ Application

- forward application: $x/y \ y \rightarrow x$
- backward application: $y \ x \backslash y \rightarrow x$
- Composition
 - forward composition: $x/y \ y/z \rightarrow x/z$
 - backward composition: $x \backslash y \ y \backslash z \rightarrow x \backslash z$
 - forward crossing composition: $x/y \ y \backslash z \rightarrow x \backslash z$
 - backward crossing composition: $x \backslash y \ y/z \rightarrow x/z$
- Type-raising
 - forward type-raising: $x \rightarrow y / (y \backslash x)$
 - backward type-raising: $x \rightarrow y \backslash (y/x)$
- Coordination <&>
 - $x \ \text{conj} \ x \rightarrow x$

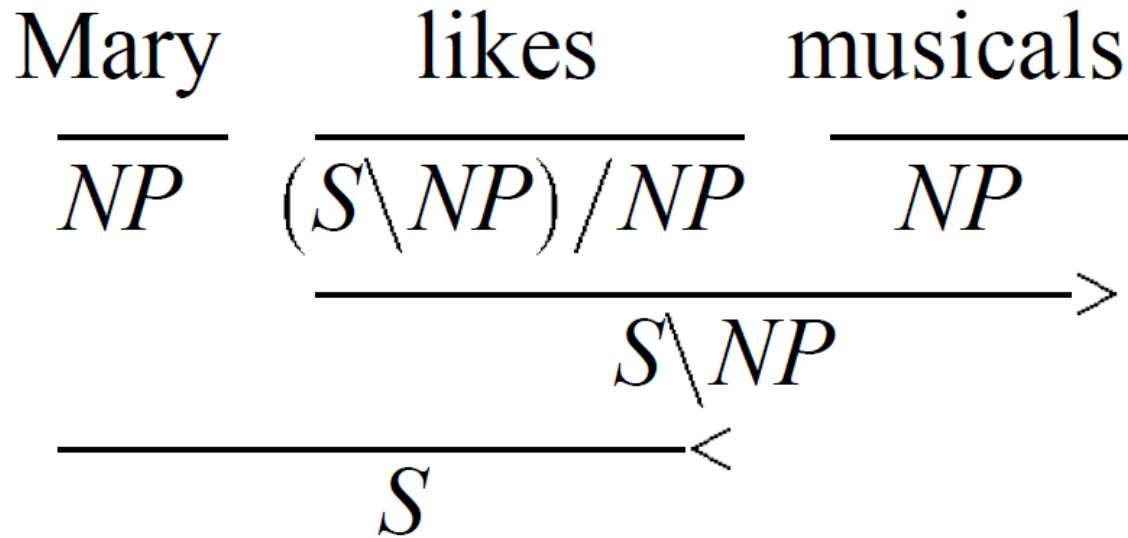
Combinatory Rules 1 : Application

- Forward application “>”
 - $X/Y \ Y \rightarrow X$
 - $(S\backslash NP) / NP \ NP \rightarrow S\backslash NP$
- Backward application “<”
 - $Y \ X\backslash Y \rightarrow X$
 - $NP \ S\backslash NP \rightarrow S$

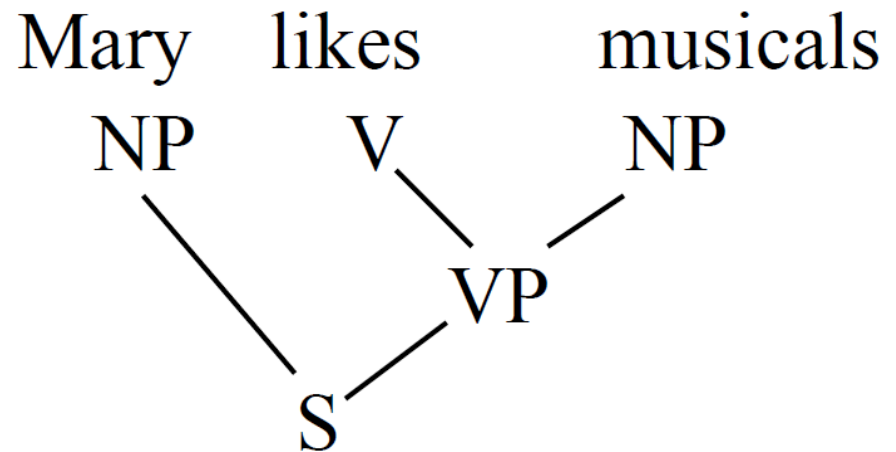
Function

- likes := $(S \setminus NP) / NP$
 - A transitive verb is a function from NPs into predicate S. That is, it accepts two NPs as arguments and results in S.
- VP or intransitive verb: $S \setminus NP$
- Transitive verb: $(S \setminus NP) / NP$
- Adverb: $(S \setminus NP) \setminus (S \setminus NP)$
- PP: $((S \setminus NP) \setminus (S \setminus NP)) / NP$
 $(NP \setminus NP) / NP$

CCG Derivation:



CFG Derivation:



Combinatory Rules

- Application
 - forward application: $x/y \ y \rightarrow x$
 - backward application: $y \ x \backslash y \rightarrow x$
- Composition
 - forward composition: $x/y \ y/z \rightarrow x/z$
 - backward composition: $x \backslash y \ y \backslash z \rightarrow x \backslash z$
 - forward crossing composition: $x/y \ y \backslash z \rightarrow x \backslash z$
 - backward crossing composition: $x \backslash y \ y/z \rightarrow x/z$
- Type-raising
 - forward type-raising: $x \rightarrow y / (y \backslash x)$
 - backward type-raising: $x \rightarrow y \backslash (y/x)$

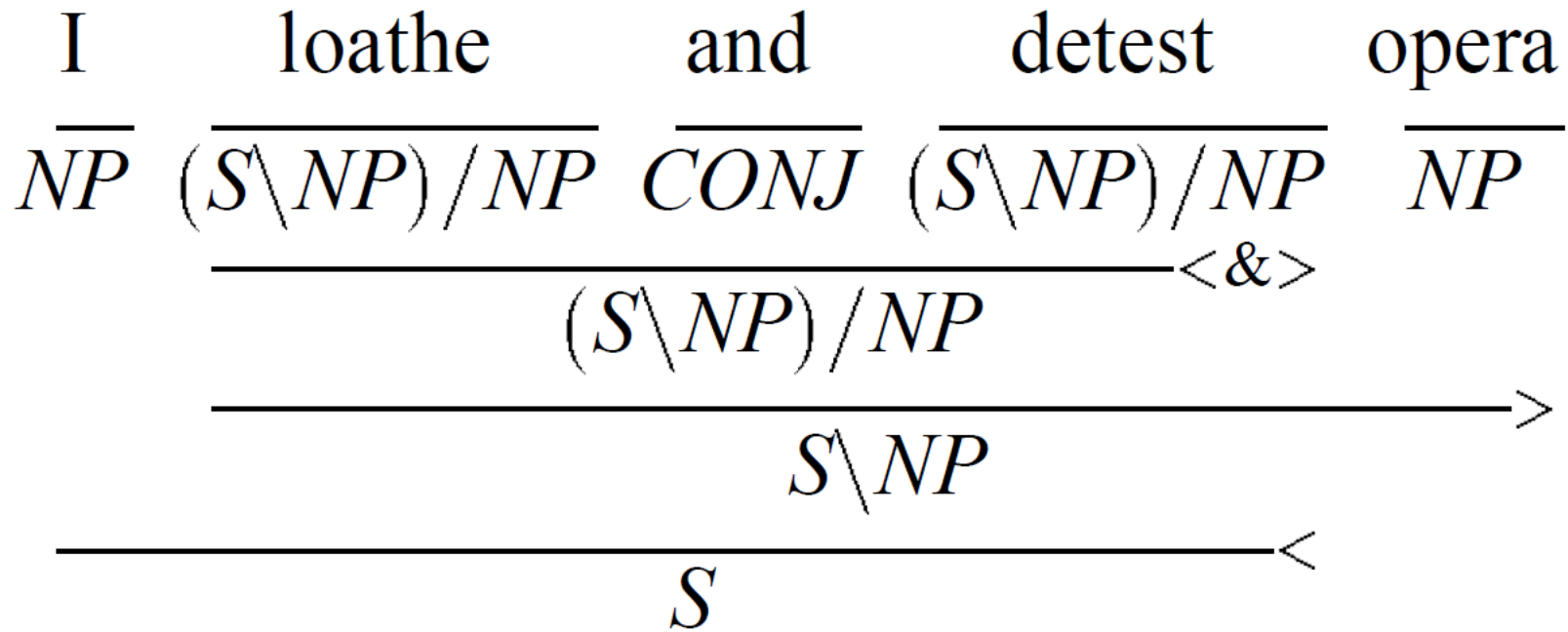
 Coordination <&>

- $x \ \text{conj} \ x \rightarrow x$

Combinatory Rules 4 : Coordination

- $X \text{ conj } X \rightarrow X$
- Alternatively, we can express coordination by defining conjunctions as functions as follows:
- $\text{and} := (X \setminus X) / X$

Coordination with CCG

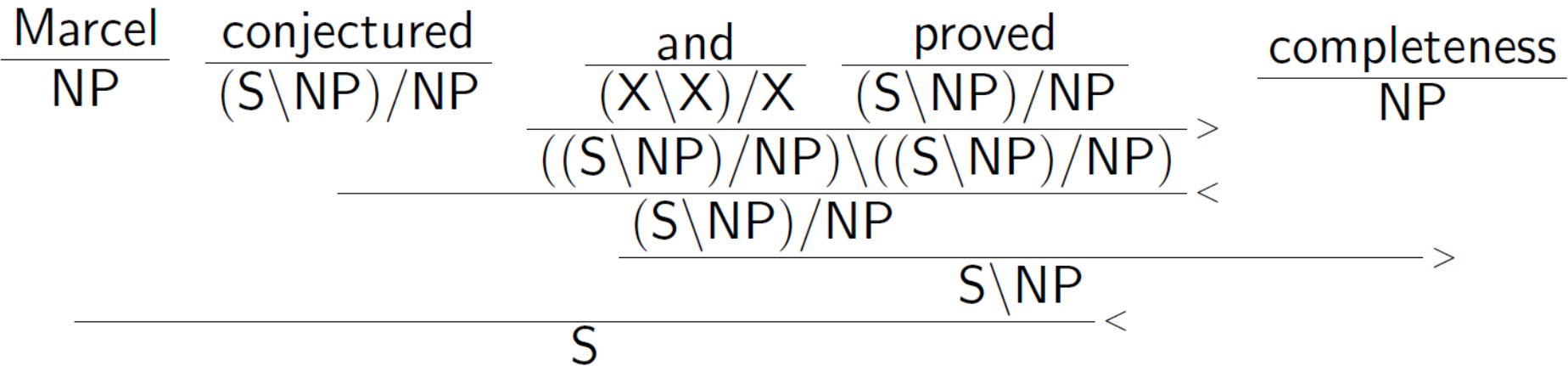


Coordination with CCG

$\frac{\text{Marcel}}{\text{NP}}$ $\frac{\text{conjectured}}{(\text{S}\backslash\text{NP})/\text{NP}}$ $\frac{\text{and}}{(\text{X}\backslash\text{X})/\text{X}}$ $\frac{\text{proved}}{(\text{S}\backslash\text{NP})/\text{NP}}$ $\frac{\text{completeness}}{\text{NP}}$

- Application
 - forward application: $x/y \ y \rightarrow x$
 - backward application: $y \ x\backslash y \rightarrow x$

Coordination with CCG



- Application
 - forward application: $x/y \ y \rightarrow x$
 - backward application: $y \ x \setminus y \rightarrow x$

Combinatory Rules

- Application

- forward application: $x/y \ y \rightarrow x$
- backward application: $y \ x \backslash y \rightarrow x$

Composition

- forward composition: $x/y \ y/z \rightarrow x/z$
- backward composition: $x \backslash y \ y \backslash z \rightarrow x \backslash z$
- forward crossing composition: $x/y \ y \backslash z \rightarrow x \backslash z$
- backward crossing composition: $x \backslash y \ y/z \rightarrow x/z$

- Type-raising

- forward type-raising: $x \rightarrow y / (y \backslash x)$
- backward type-raising: $x \rightarrow y \backslash (y/x)$

- Coordination <&>

- $x \ \text{conj} \ x \rightarrow x$

Coordination with CCG

$\frac{\text{Marcel}}{\text{NP}}$	$\frac{\text{conjectured}}{(S \backslash \text{NP}) / \text{NP}}$	$\frac{\text{and}}{(X \backslash X) / X}$	$\frac{\text{might}}{(S \backslash \text{NP}) / ((S \backslash \text{NP}))}$	$\frac{\text{prove}}{(S \backslash \text{NP}) / \text{NP}}$	$\frac{\text{completeness}}{\text{NP}}$
-----------------------------------	---	---	--	---	---

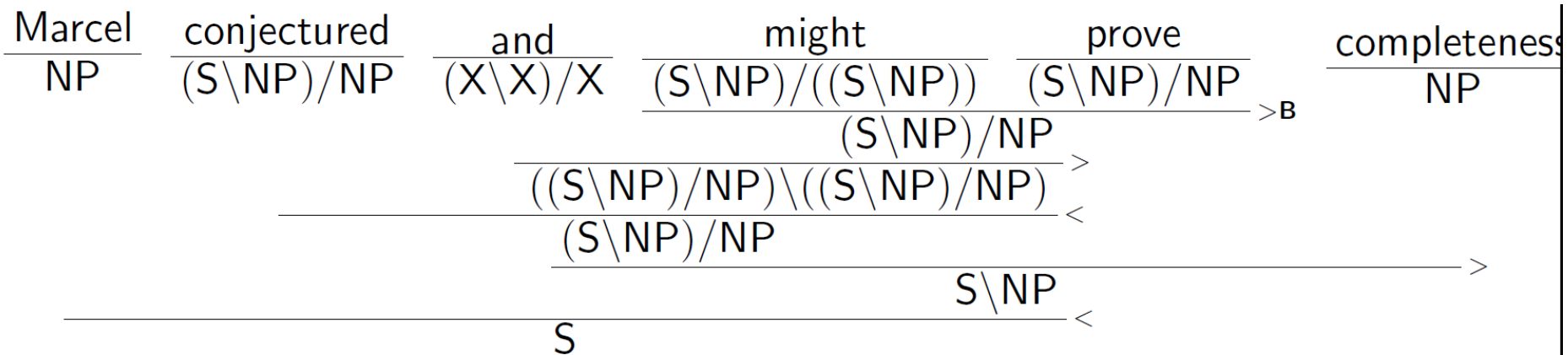
- Application

- forward application: $x/y \ y \rightarrow x$
- backward application: $y \ x \backslash y \rightarrow x$

- Composition

- forward composition: $x/y \ y/z \rightarrow x/z$
- backward composition: $x \backslash y \ y \backslash z \rightarrow x \backslash z$
- forward crossing composition: $x/y \ y \backslash z \rightarrow x \backslash z$
- backward crossing composition: $x \backslash y \ y/z \rightarrow x/z$

Coordination with CCG



- Application

- forward application: $x / y \quad y \rightarrow x$
- backward application: $y \quad x \backslash y \rightarrow x$

- Composition

- forward composition: $x / y \quad y / z \rightarrow x / z$
- backward composition: $x \backslash y \quad y \backslash z \rightarrow x \backslash z$
- forward crossing composition: $x / y \quad y \backslash z \rightarrow x \backslash z$
- backward crossing composition: $x \backslash y \quad y / z \rightarrow x / z$

Combinatory Rules

- Application
 - forward application: $x/y \ y \rightarrow x$
 - backward application: $y \ x \backslash y \rightarrow x$
- Composition
 - forward composition: $x/y \ y/z \rightarrow x/z$
 - backward composition: $x \backslash y \ y \backslash z \rightarrow x \backslash z$
 - forward crossing composition: $x/y \ y \backslash z \rightarrow x \backslash z$
 - backward crossing composition: $x \backslash y \ y/z \rightarrow x/z$

Type-raising

- forward type-raising: $x \rightarrow y / (y \backslash x)$
- backward type-raising: $x \rightarrow y \backslash (y/x)$
- Coordination <&>
 - $x \ \text{conj} \ x \rightarrow x$

Combinatory Rules 3 : Type-Raising

- Turns an argument into a function
- Forward type-raising: $X \rightarrow T / (T \setminus X)$
- Backward type-raising: $X \rightarrow T \setminus (T/X)$

For instance...

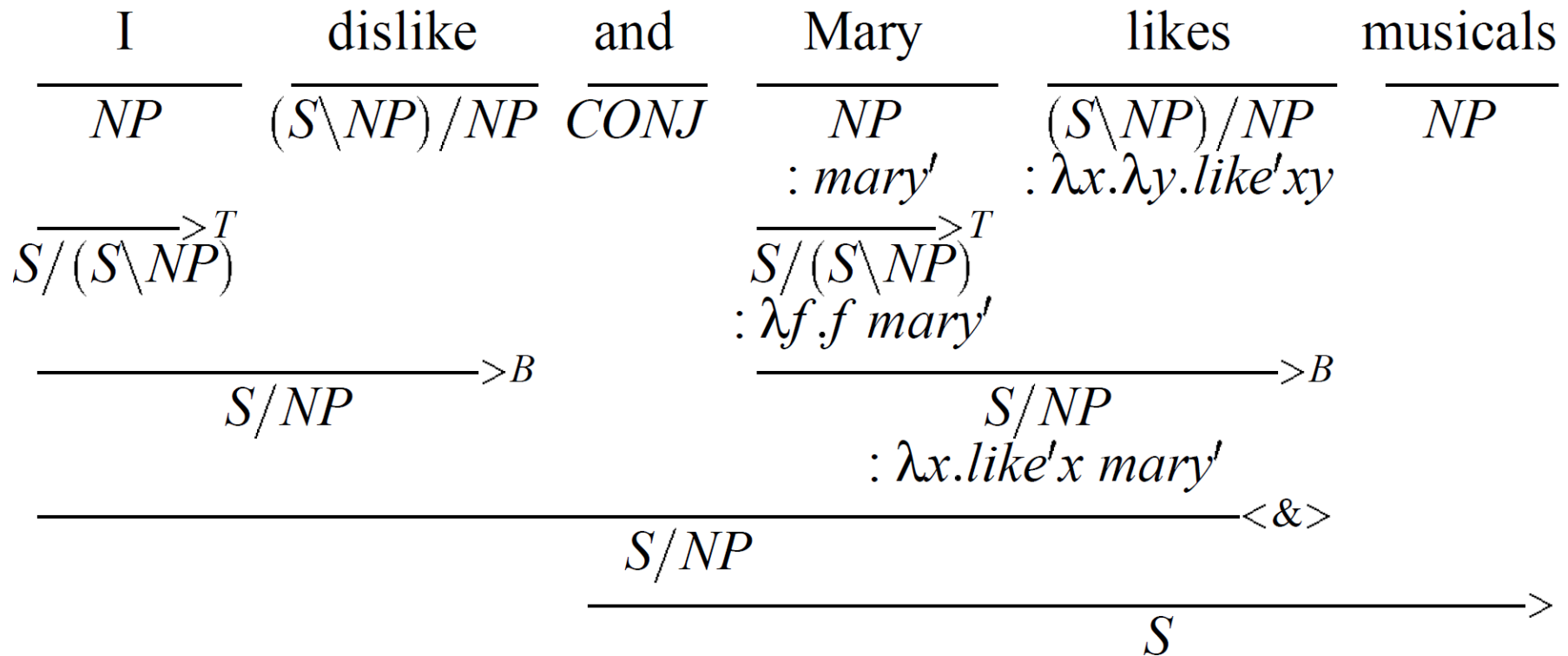
- Subject type-raising: $NP \rightarrow S / (S \setminus NP)$
- Object type-raising: $NP \rightarrow (S \setminus NP) \setminus ((S \setminus NP) / NP)$

Combinatory Rules 3 : Type-Raising

I	dislike	and	Mary	likes	musicals
NP	$(S \setminus NP) / NP$	$CONJ$	NP	$(S \setminus NP) / NP$	NP

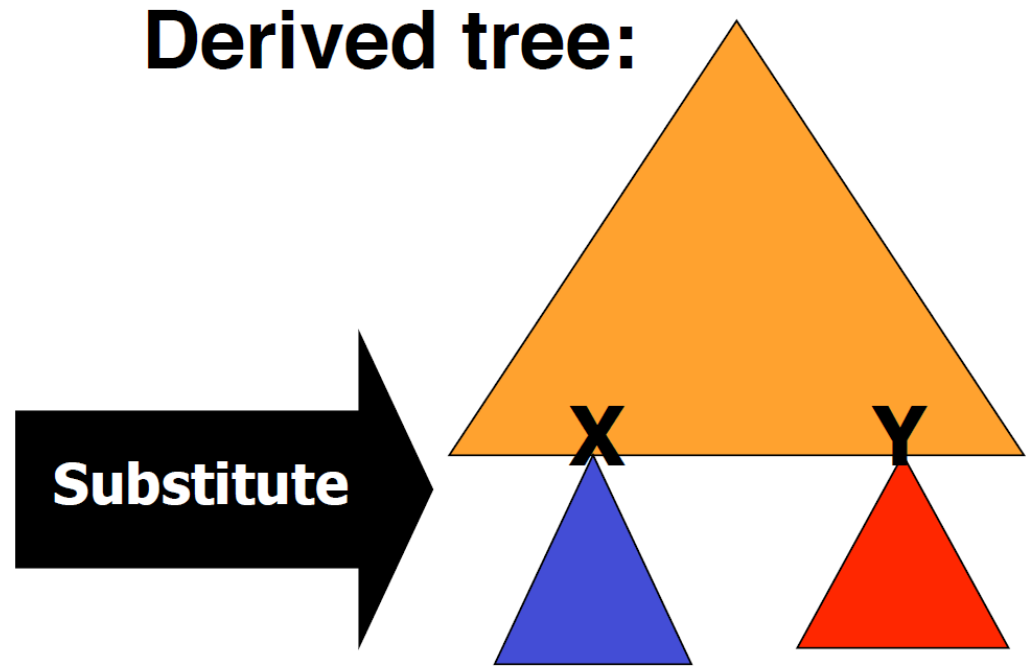
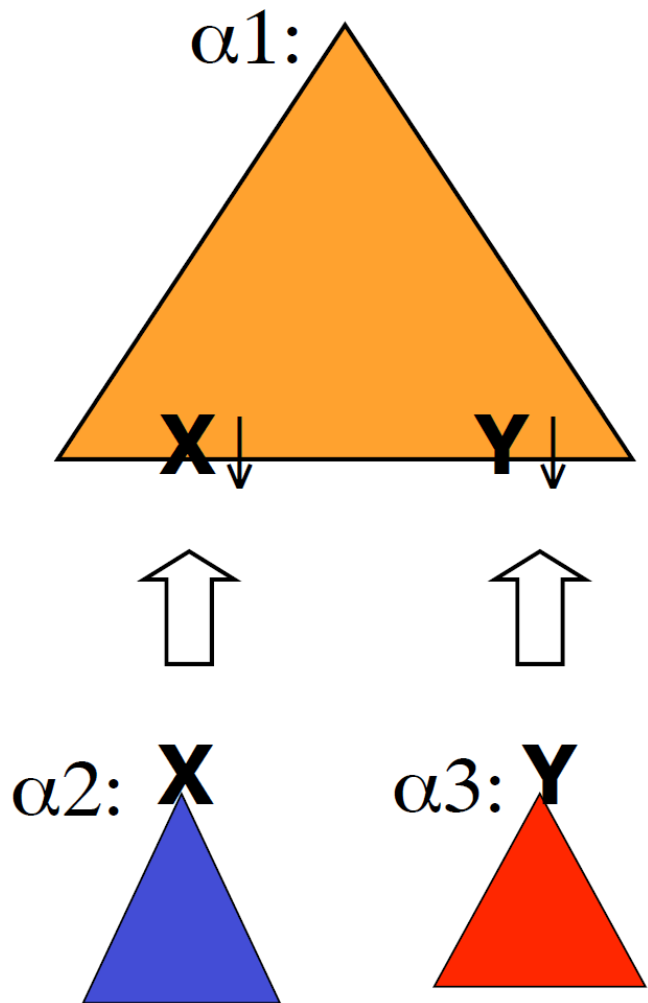
- Application
 - forward application: $x/y \ y \rightarrow x$
 - backward application: $y \ x \setminus y \rightarrow x$
- Type-raising
 - forward type-raising: $x \rightarrow y / (y \setminus x)$
 - backward type-raising: $x \rightarrow y \setminus (y/x)$
 - Subject type-raising: $NP \rightarrow S / (S \setminus NP)$
 - Object type-raising: $NP \rightarrow (S \setminus NP) \setminus ((S \setminus NP) / NP)$
- Coordination <&>
 - $x \ conj \ x \rightarrow x$

Combinatory Rules 3 : Type-Raising

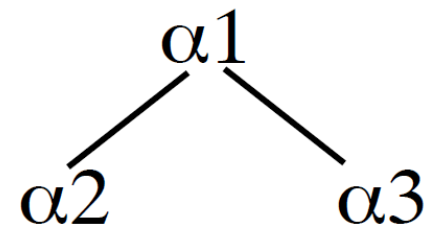


Tree Adjoining Grammar (TAG)

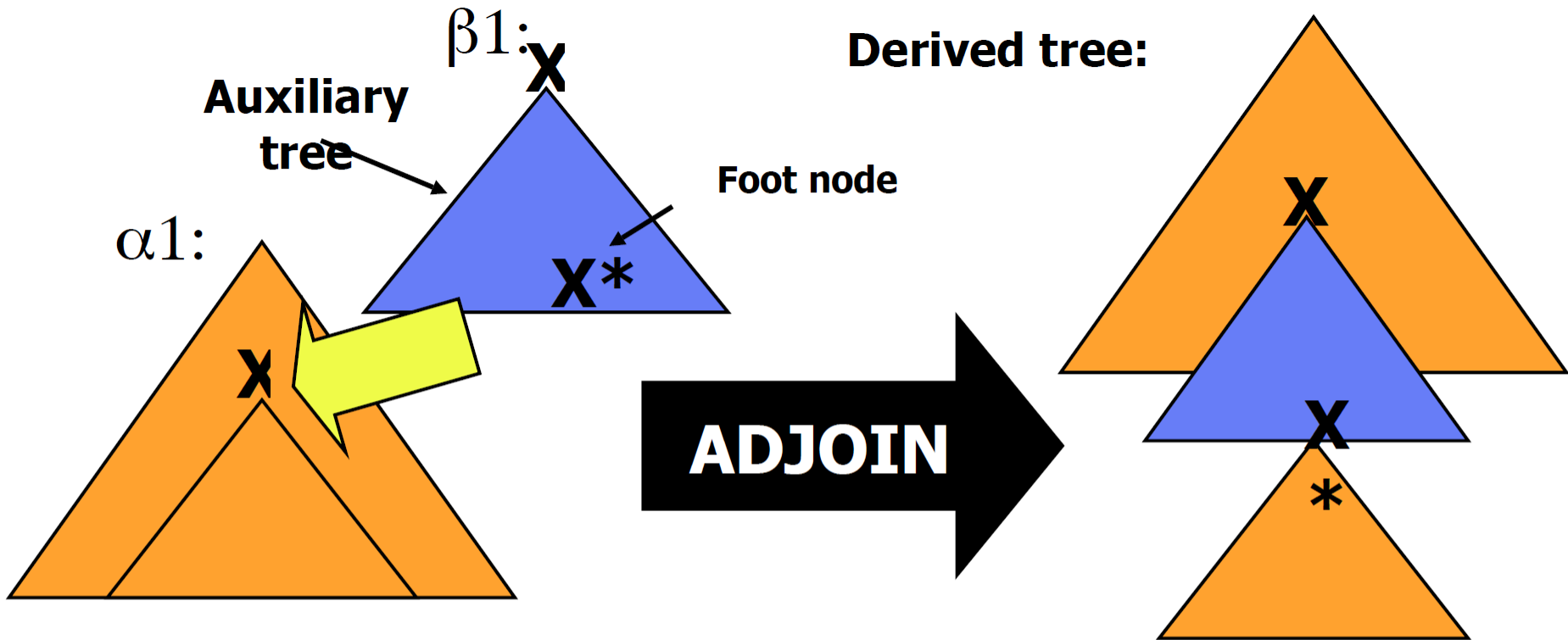
TAG rule 1: Substitution



Derivation tree:



TAG rule 2: Adjunction

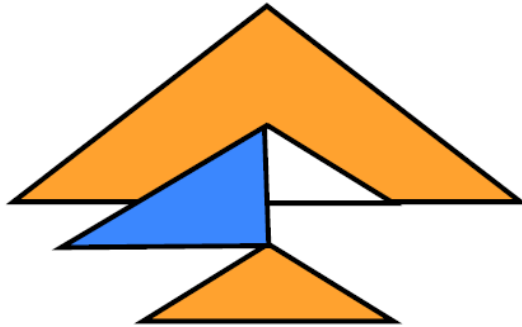


Derivation tree:

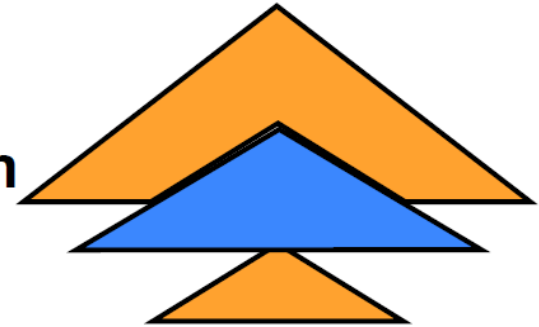
$\alpha 1$
⋮
 $\beta 1$

The effect of adjunction

**TIG:
sister
adjunction**

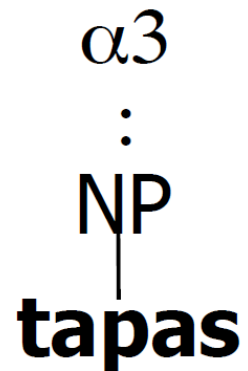
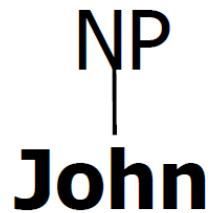


**TAG:
wrapping
adjunction**

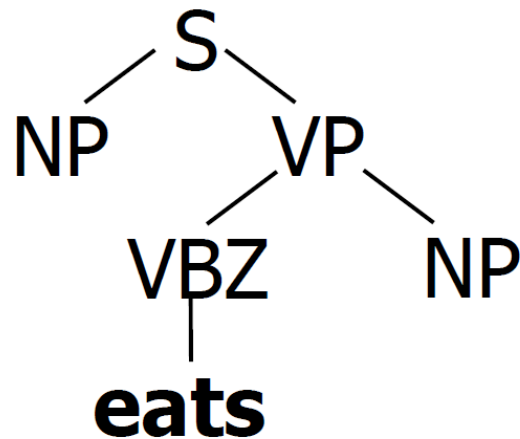


Example: TAG Lexicon

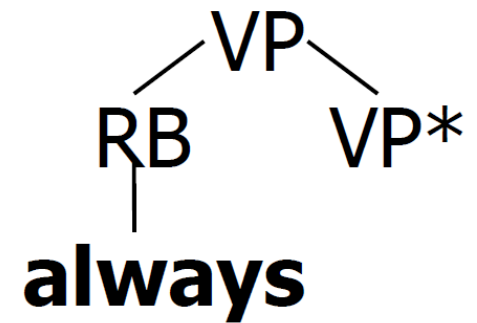
α_2 :



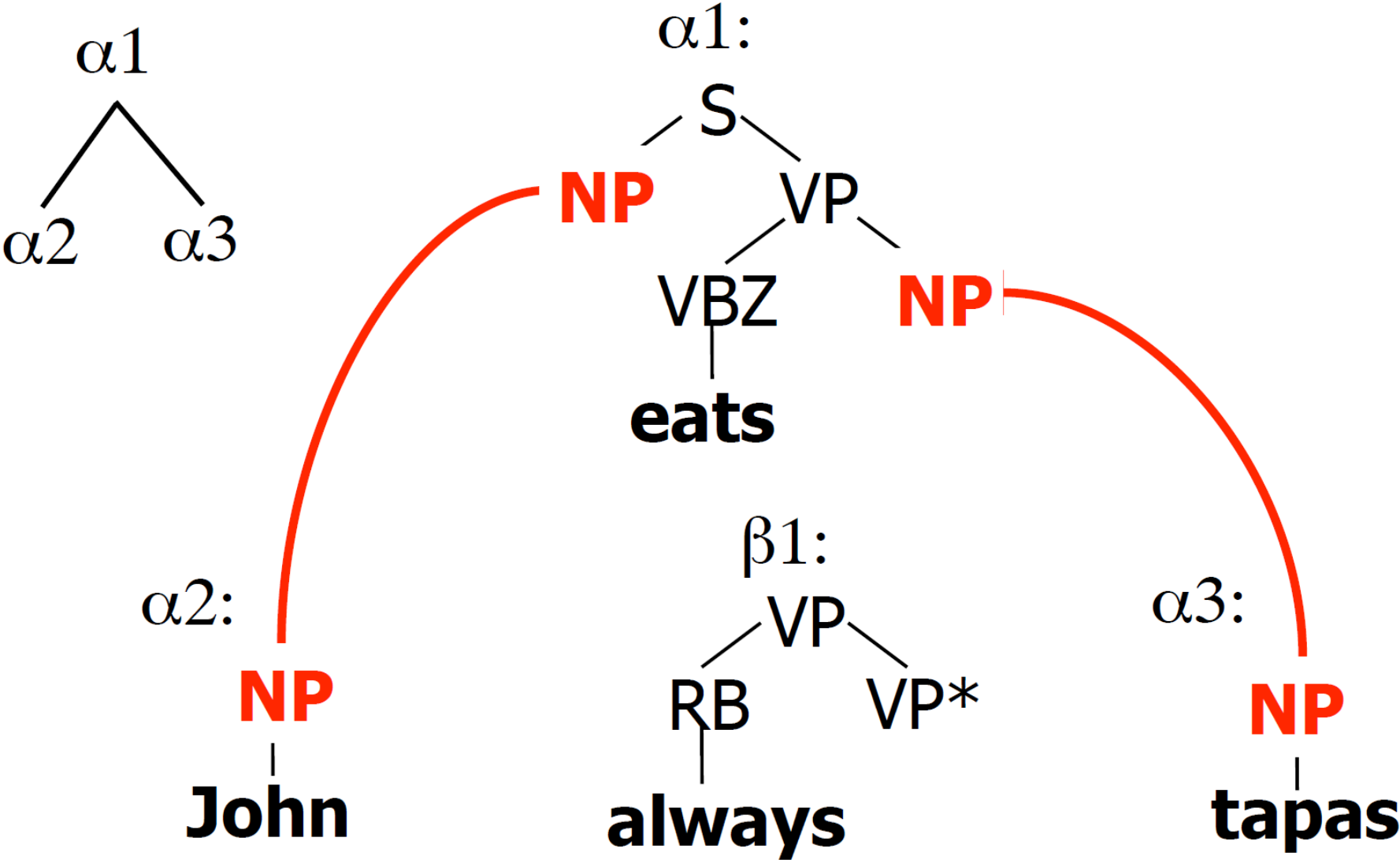
α_1 :



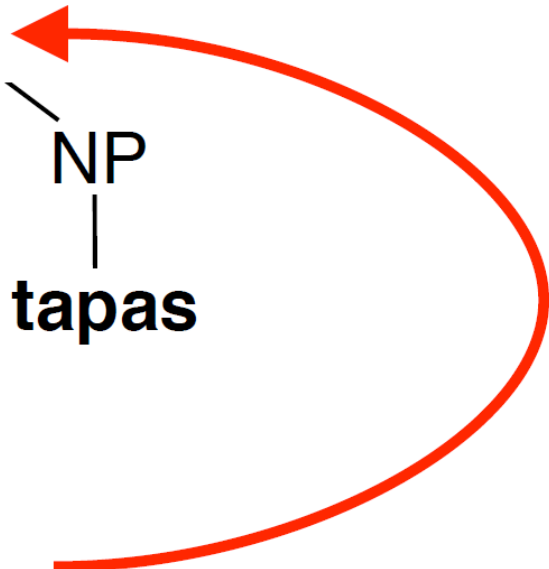
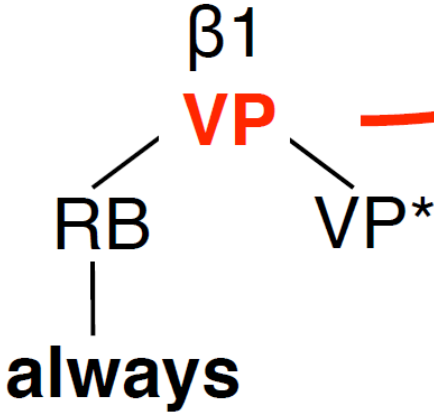
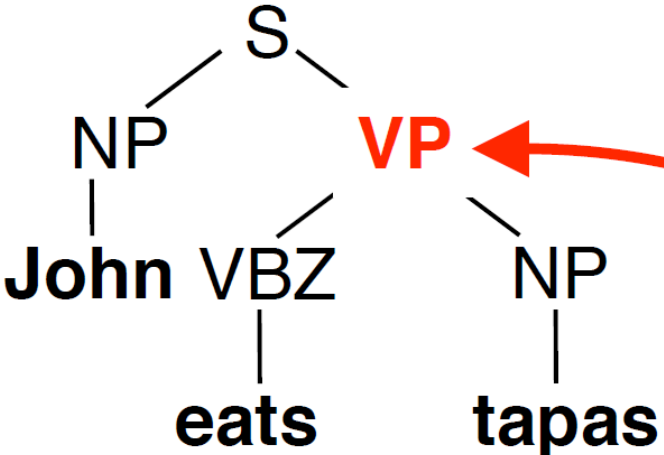
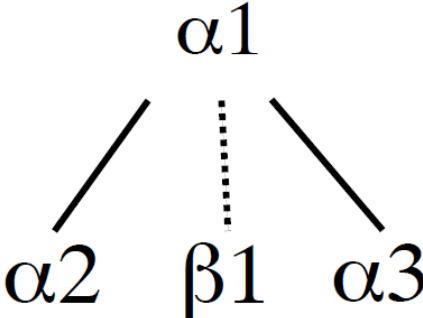
β_1 :



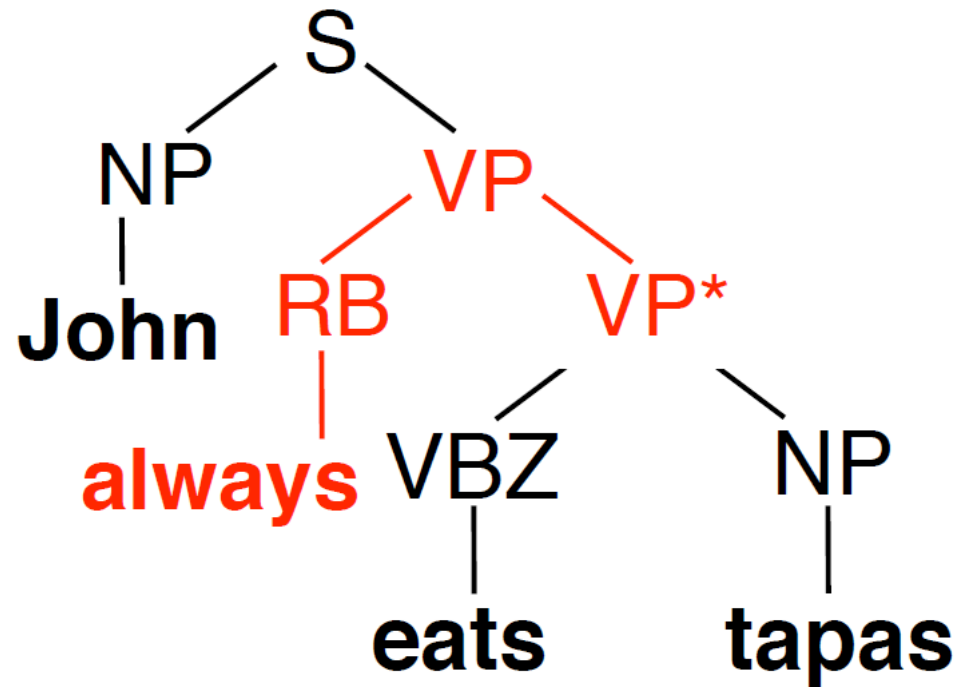
Example: TAG Derivation



Example: TAG Derivation

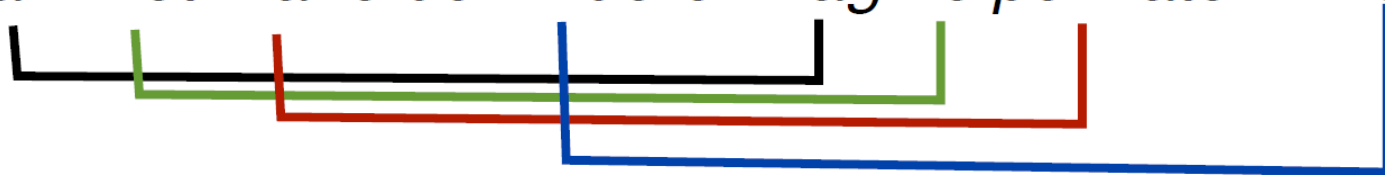


Example: TAG Derivation



Cross-serial Dependencies

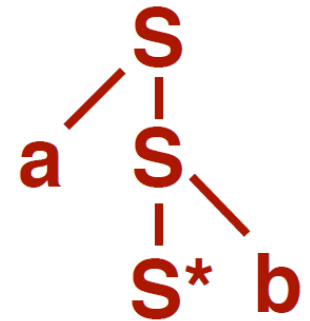
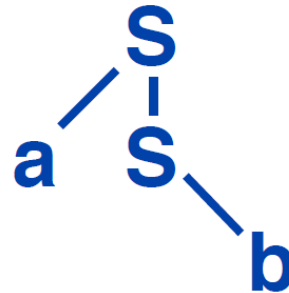
dat Jan Piet Marie de kinderen zag helpen laten zwemmen



- Dutch and Swiss-German
- Comparison to regular grammar and CFG?

$a^n b^n$: Cross-serial dependencies

Elementary trees:



Deriving **aabb**

