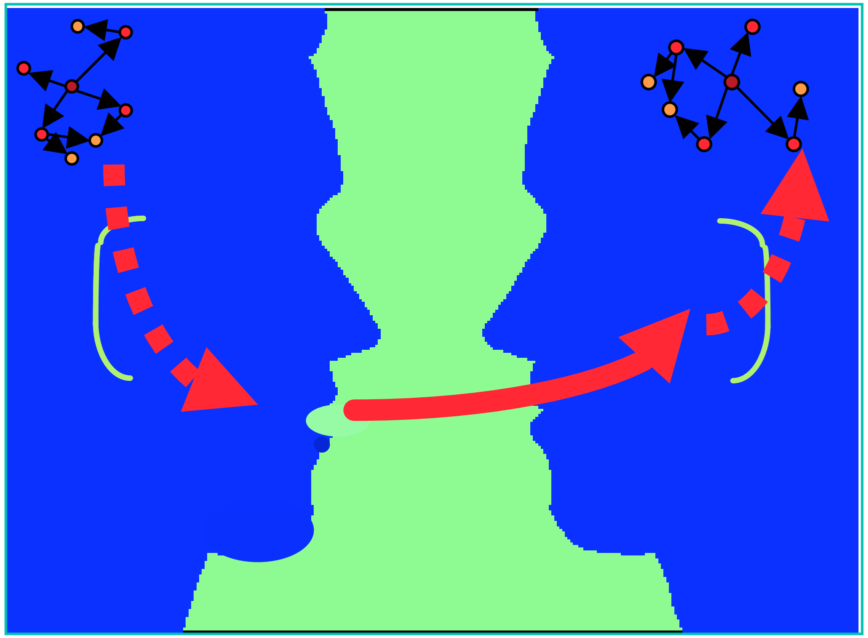
FLST

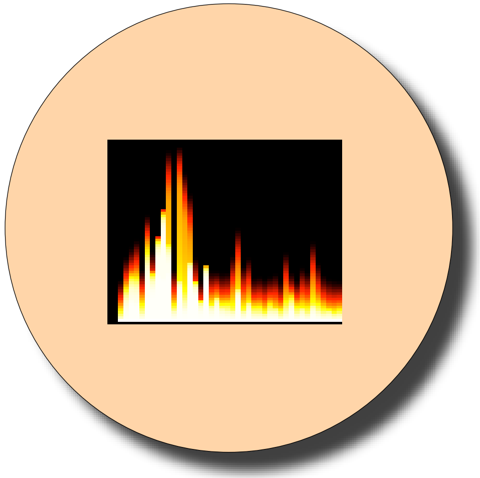
Grammars and Parsing

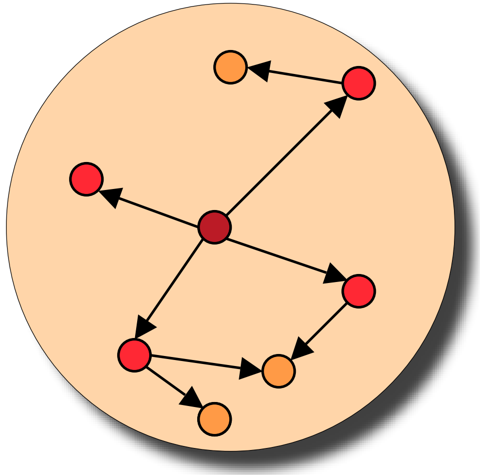
Hans Uszkoreit

WHAT HAPPENS IN BETWEEN?



WHAT HAPPENS IN BETWEEN?

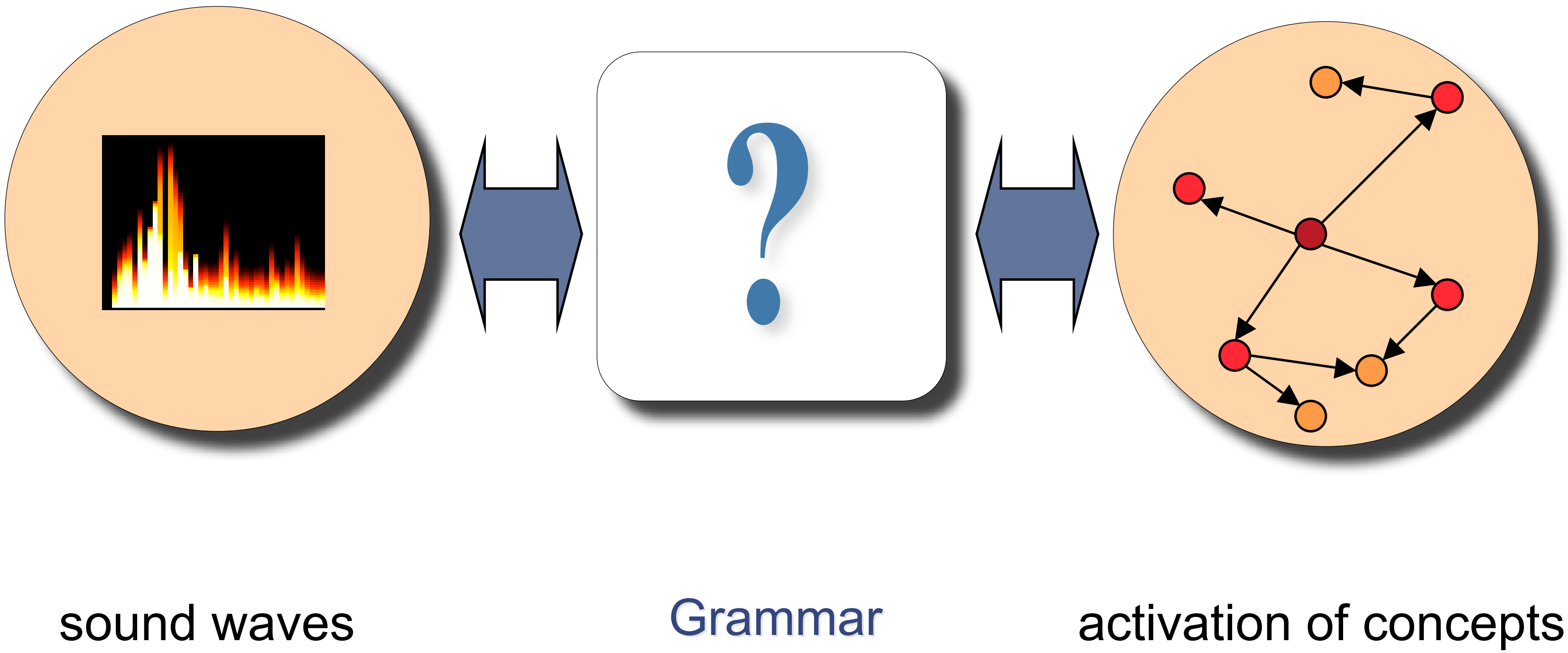




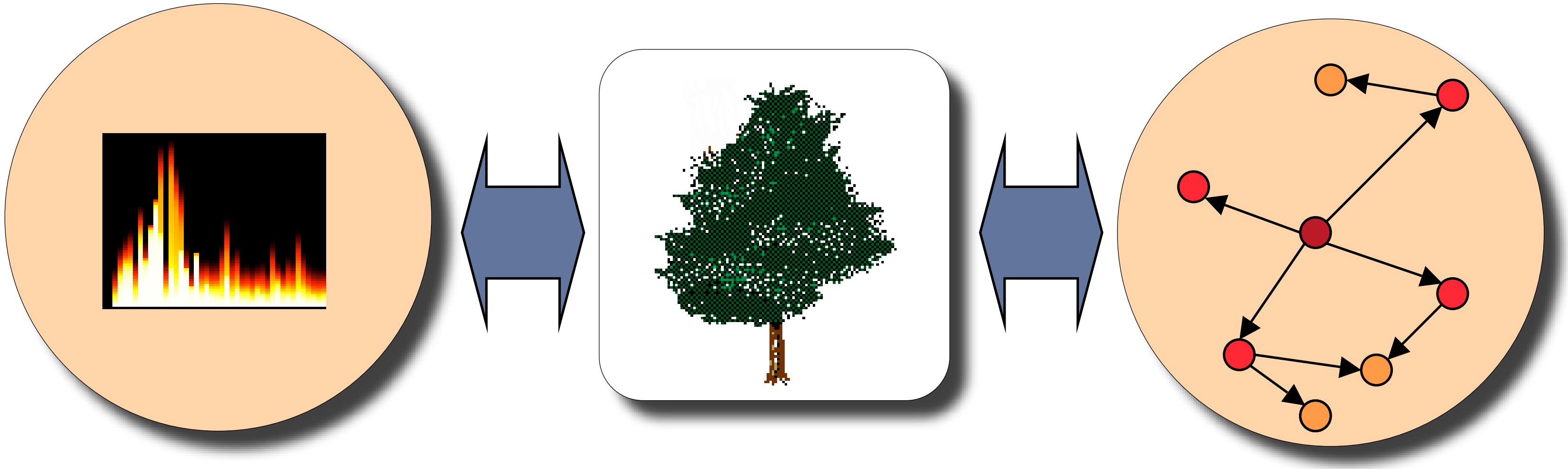
sound waves

activation of concepts

What Happens in Between?



What Happens in Between?

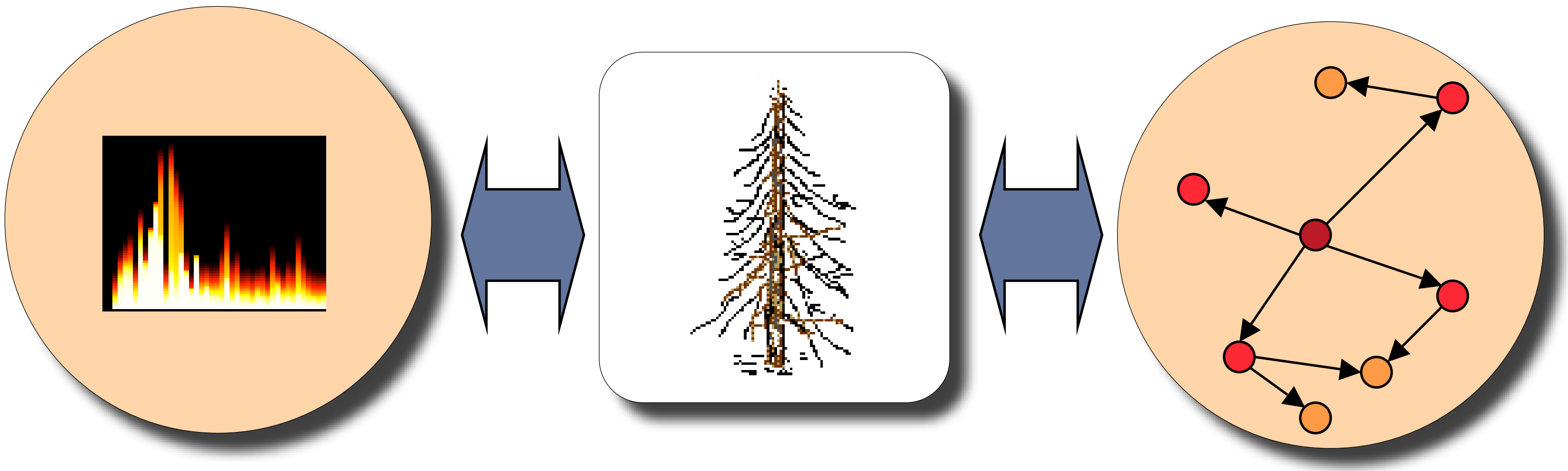


sound waves

Grammar

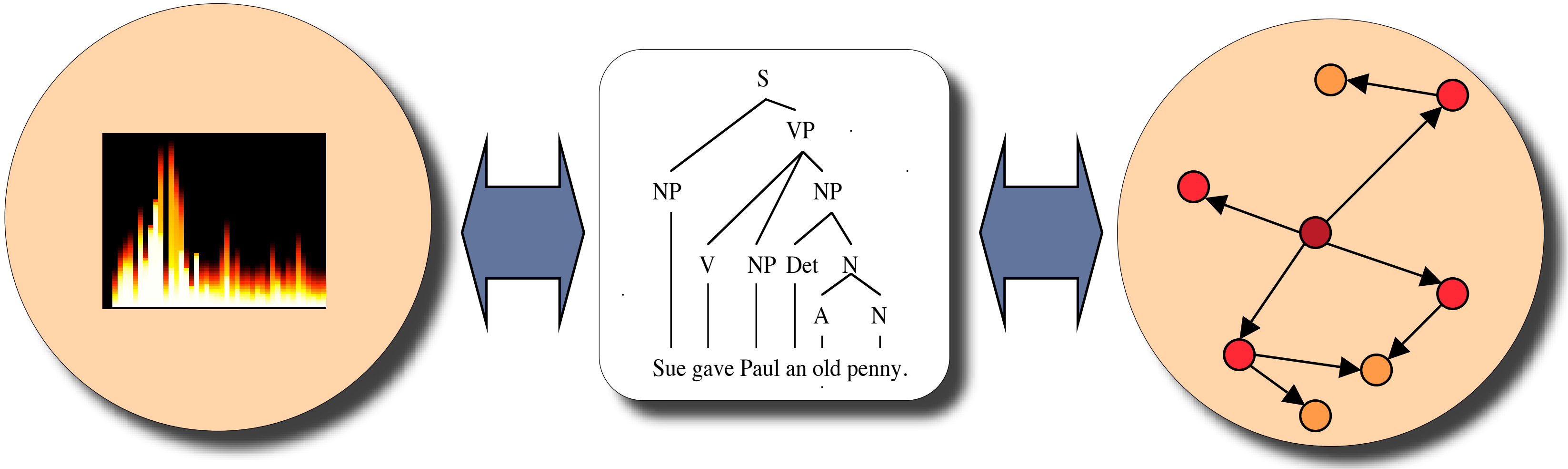
activation of concepts

What Happens in Between?



sound waves | Grammar | activation of concepts

What Happens in Between?

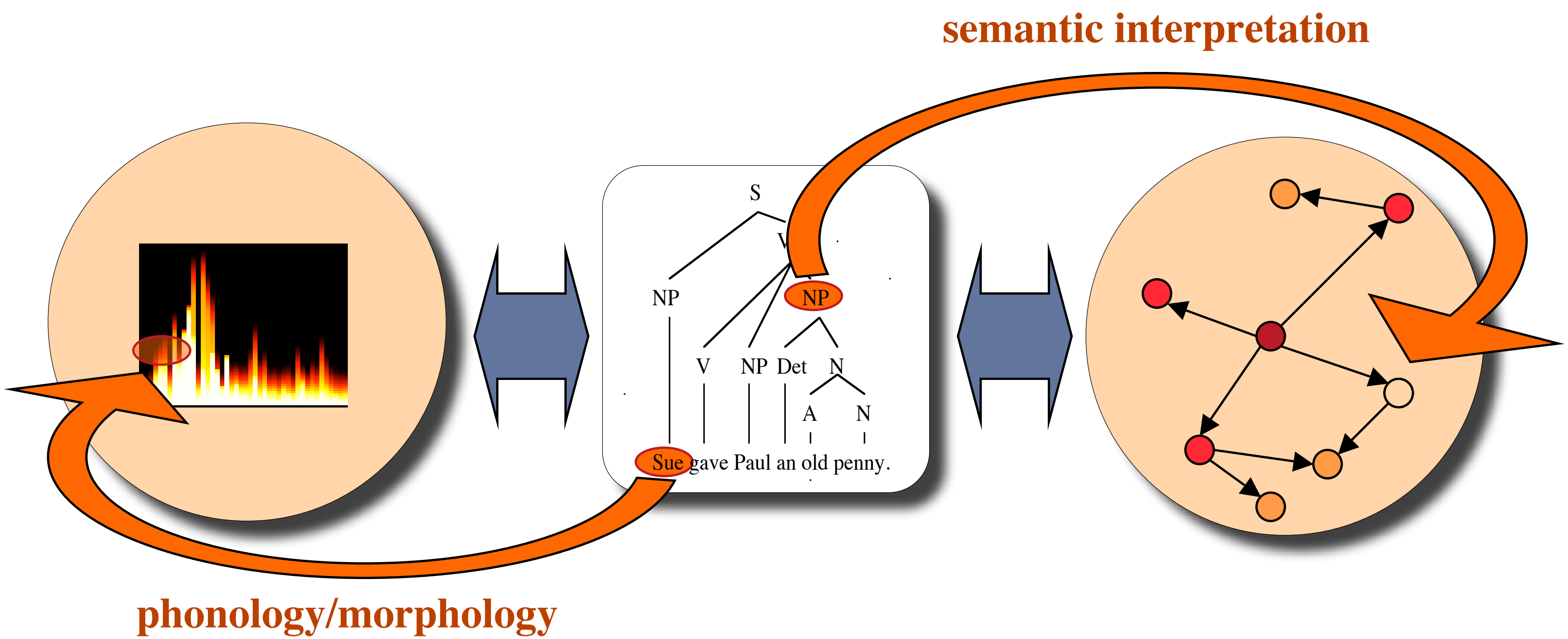


sound waves

Grammar

activation of concepts

What Happens in Between?

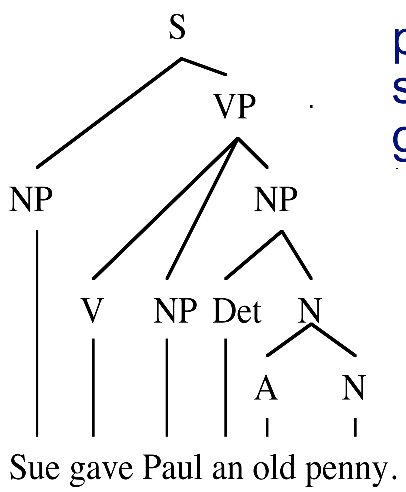


sound waves

Grammar

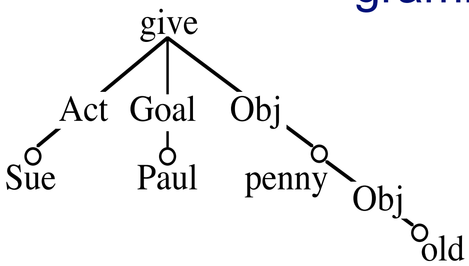
activation of concepts

THREE TRADITIONS

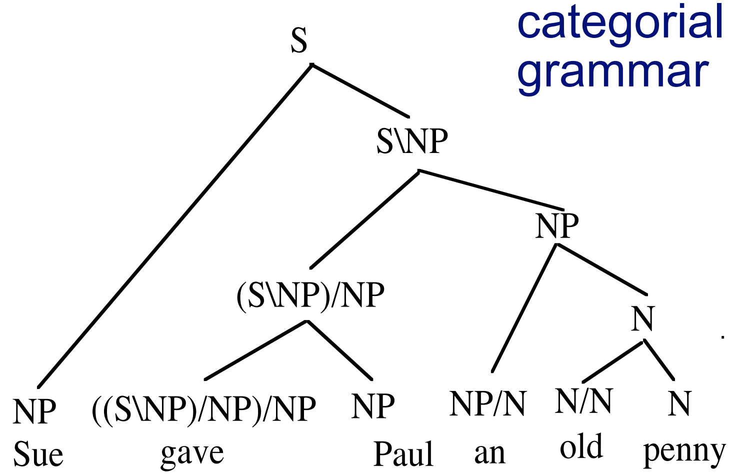


phrase structure grammar

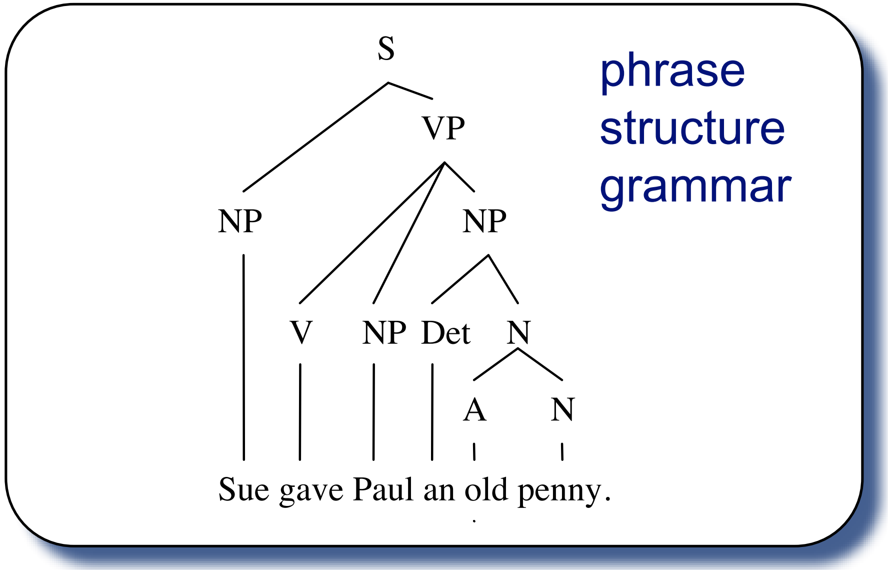
dependency grammar



categorial grammar

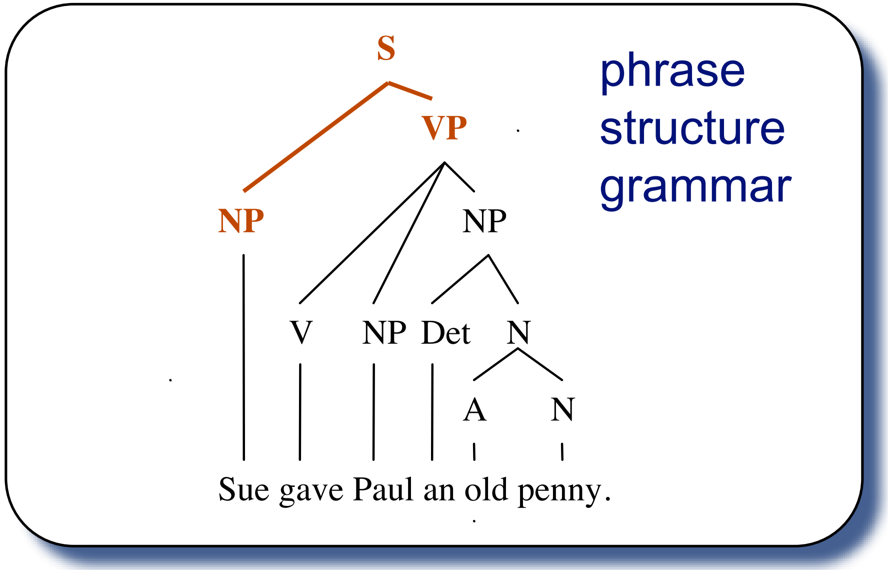


Grammatik



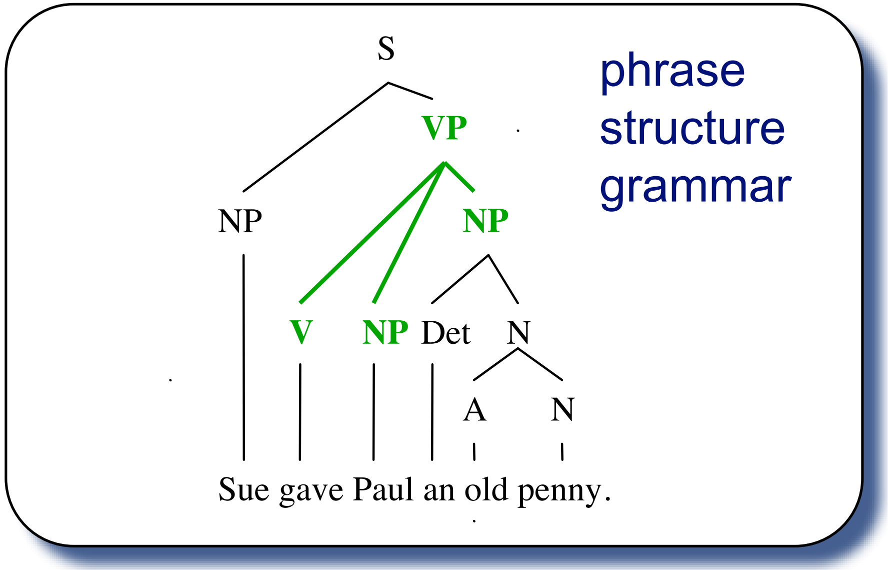
S → NP VP

Grammatik



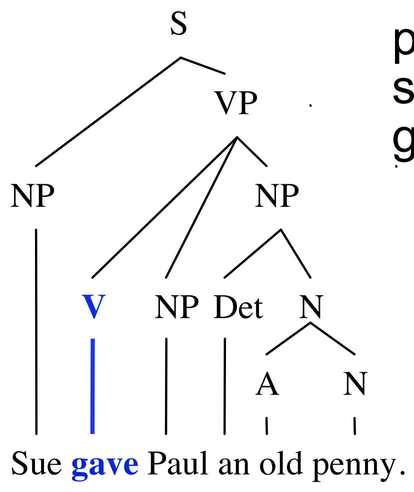
S → NP VP

Grammatik



S → NP VP
VP → V NP NP

Grammar



phrase structure grammar

S → NP VP
VP → V NP NP

V → gave

symbol conventions

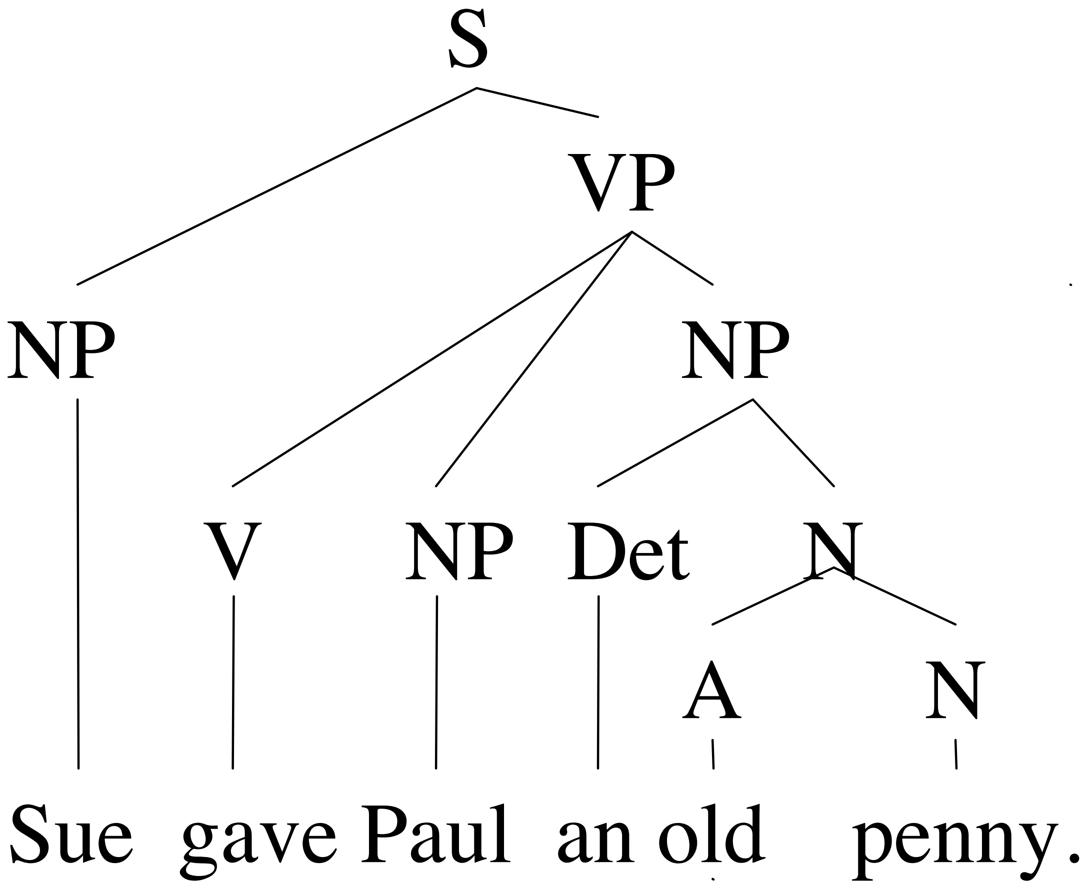
	single symbols	strings
non-terminal	A, B, C, ...	... , X, Y, Z
terminals	a, b, c, ...	... , x, y, z
unspecified symbols	$\alpha, \beta, \gamma, \ldots$	$\ldots, \phi, \chi, \psi, \omega$
start symbol	S	
empty string	e	
integers	..., i, j, k, l, m, n, ...	

Why Syntax

- Einen Hund hat dieser Mann gebissen.
- Ein Hund hat diesen Mann gebissen.

- This man has bitten a dog.
- A dog this man has bitten.
- A dog has bitten this man.

- Peter promised Paul, to process the files.
- Peter persuaded Paul, to process the files.


S
VP
NP
NP
V
NP
Det
N
A
N
Sue gave Paul an old penny.

Formal Grammar

A language over an alphabet (vocabulary) Σ is a subset of Σ^*.

A formal grammar G_L for a language L is a quadruple $(V_N, V_T, \{S\}, P)$.

V_N - non-terminal vocabulary (auxiliary vocabulary)

V_T - terminal vocabulary

$(V_T \cap V_N = \emptyset, \quad L \subseteq V_T^*, \quad V = V_T \cup V_N)$

{S} - singleton with the start symbol (set of axioms)

P - set of productions, rule set

set of rules of the form $\omega_1 \phi \omega_2 \rightarrow \omega_1 \psi \omega_2$

usually written as $\phi \rightarrow \psi$

<u>relation “follows":</u>

If G = (V_N, V_T, {S}, P), then ψ follows from ϕ according to G iff there are strings ϕ_1, ϕ_2, χ, ω, so that $\phi = \phi_1, \chi, \phi_2$ und $\psi = \phi_1, \omega, \phi_2$ und $\chi \rightarrow \omega \in P$.

Notation: $! \ \#_{G}$

<u>derivation:</u>

A sequence of strings $\phi_1, \phi_1, ..., \phi_n$ is a derivation according to G iff $!_{i} \; {}_{G} \; !_{i+1}$ for all i, $1 \leq i \leq n$.

If there is *derivable* according to G from ϕ to ψ we can write this: $! \ \#^{*}_{G}$

The relation *derivable* is transitive and is moreover defined to be reflexive.

The generated language

<u>The language L:</u> A string ω is in L according to G_L iff the following three conditions are fulfilled:

1. $\omega \in V_T^*$
2. $S \ \underset{G}{\overset{*}{}} \ !$
3. There is no χ, so that $! \ \underset{G}{\overset{*}{\#}}$ and $\omega \neq \chi$.

We say that G_L generates the language L. The language L generated by G is also written as L(G).

<u>Weak Equivalence:</u> Two grammar G_1 and G_2 are weakly equivalent, if they generate the same language.

Types of Grammars

Type 0 (unrestricted rewriting systems):

Every formal grammar according to the definition is of type 0.

Type 1 (context sensitive grammars):

Every production is of the form $\phi A \psi \rightarrow \phi \omega \psi$, where $A \in V_N$, $\omega \neq \varepsilon$.

Type 2 (context free grammars):

Every production is of the form $A \rightarrow \omega$, where $\omega \neq \varepsilon$.

Type 3 (regular grammars):

Every production is of the form $A \rightarrow x\ B$ or $A \rightarrow x$, where $x \neq \varepsilon$.

context-free derivations

S

NP VP

DET ADJ N VP

DET ADJ N V NP

DET ADJ N V DET ADJ N

.

.

.

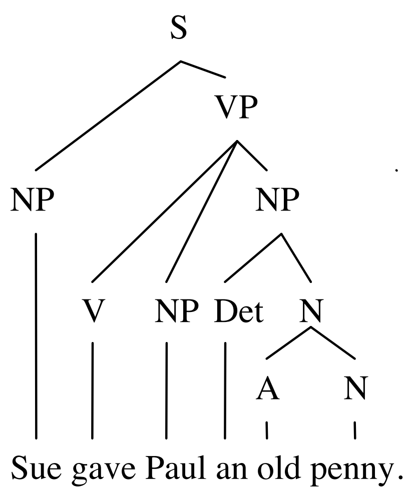
ein kleines Mädchen sucht ein kleines Mädchen

Trees

the notion of syntactic constituent tree

coded information

1. the hierarchic organisation of a sentence in terms of constituents
2. the assignment of each constituent to a linguistic class (category)
3. the linear sequence of the constituents



Trees II

relations: immediate dominance - dominance
immediate precedence - precedence

constituent structure tree: quintuple(N, Q, D, P, L)

N - finite set of nodes
Q - finite set of labels
D - weak partial order in N x N, the dominance relation
(reflexive, transitive und antisymmetric)
P - strong partial order in N x N, the precedence relation
(irreflexive, transitive und asymmetric)
L - function from N into Q, the labelling function

conditions:

(single) root condition
exclusivity condition
no crossing condition / no tangling condition

Conditions

root condition

There is exactly one node that dominates all other nodes of the tree.

exclusivity condition

For any two nodes x and y holds either D(x,y) or D(y,x)
or P(x,y) or P(y,x)
but never both.

no-tangling condition

If P(x,y) then for all x' dominated by x [D(x,x')] and for all y' dominated by y [D(y, y')] that x' precedes y' [P(x', y')]

Parsing

- The syntactic analysis of strings according to a grammar we call parsing
- input: terminal string

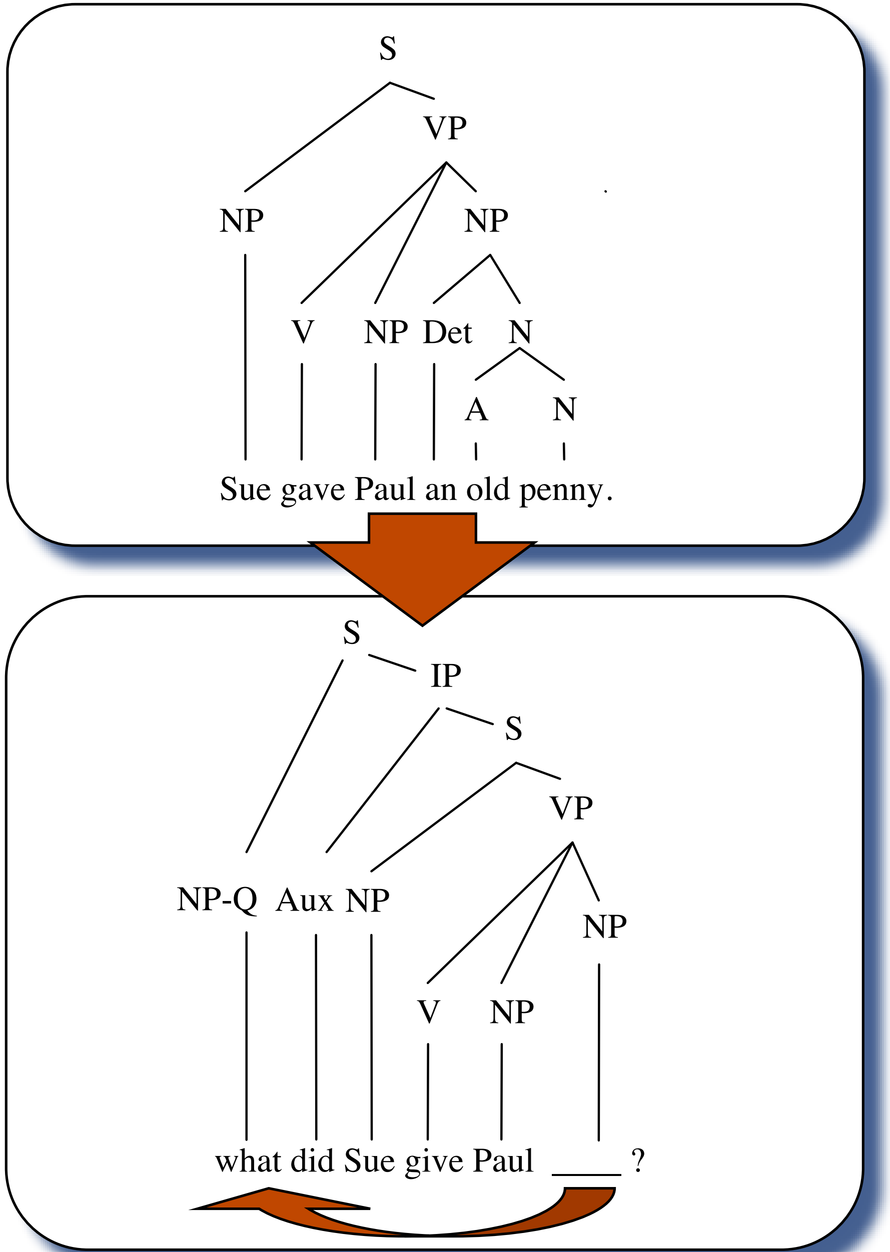
  output: structure of the sentence, i.e. all constituents usually a tree
- parsing algorithms:

  top-down vs bottom-up

  left-right vs right left vs birectional or island parsing

  deterministic vs. non-deterministic

Grammar



transformational grammar

Grammar

Unification grammar