



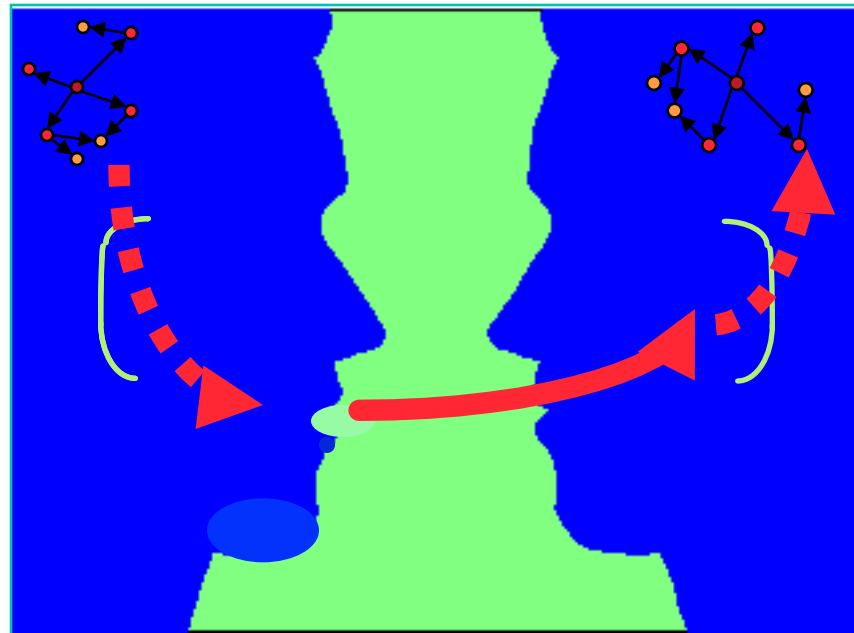
# **Vorlesung Grammatikformalismen:**

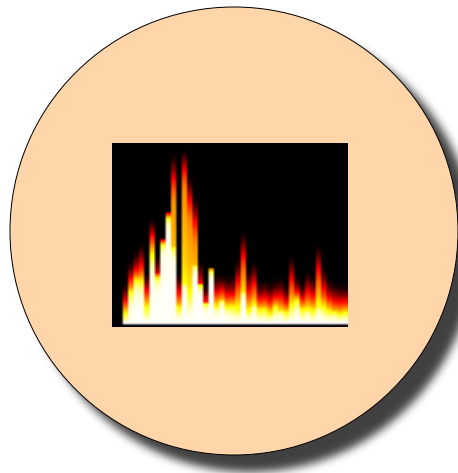
## **Die Rolle der Grammatik**

### **Ersetzungsregelgrammatiken**

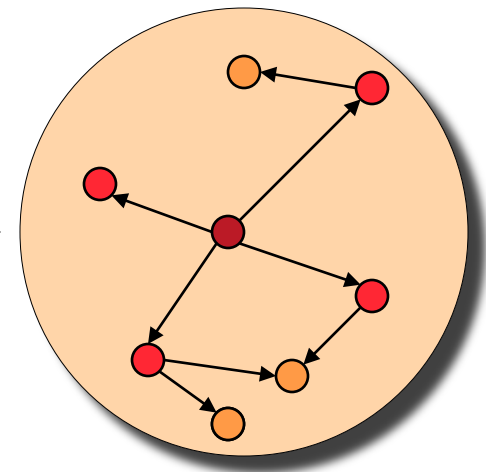
Hans Uszkoreit

# WHAT HAPPENS IN BETWEEN?



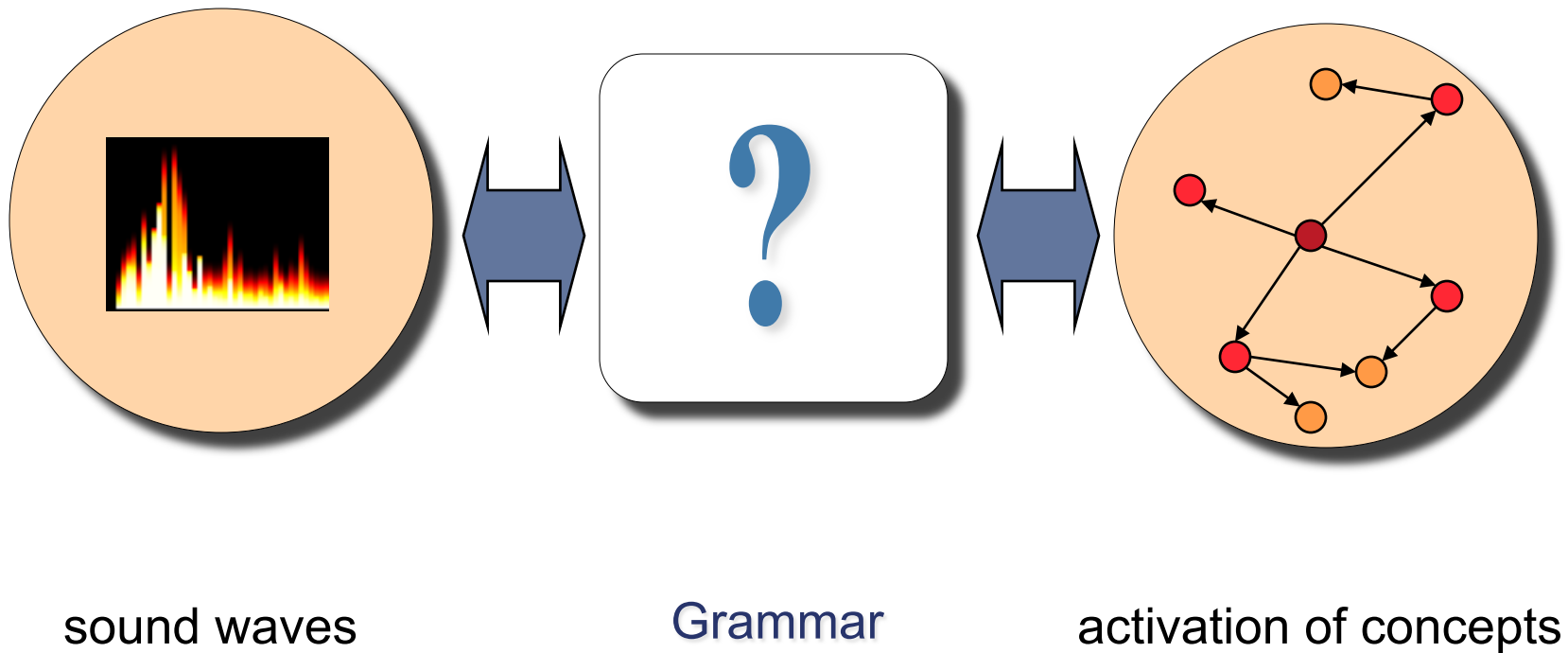


sound waves

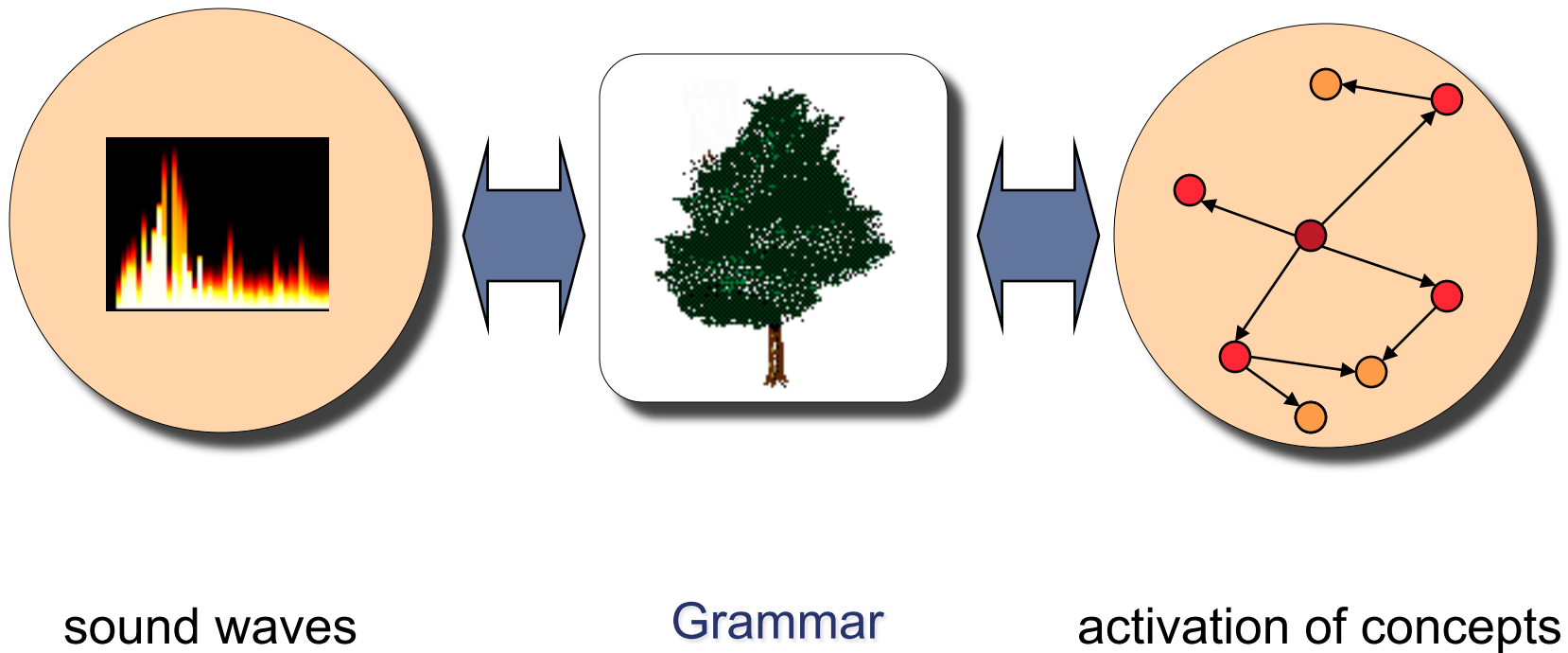


activation of concepts

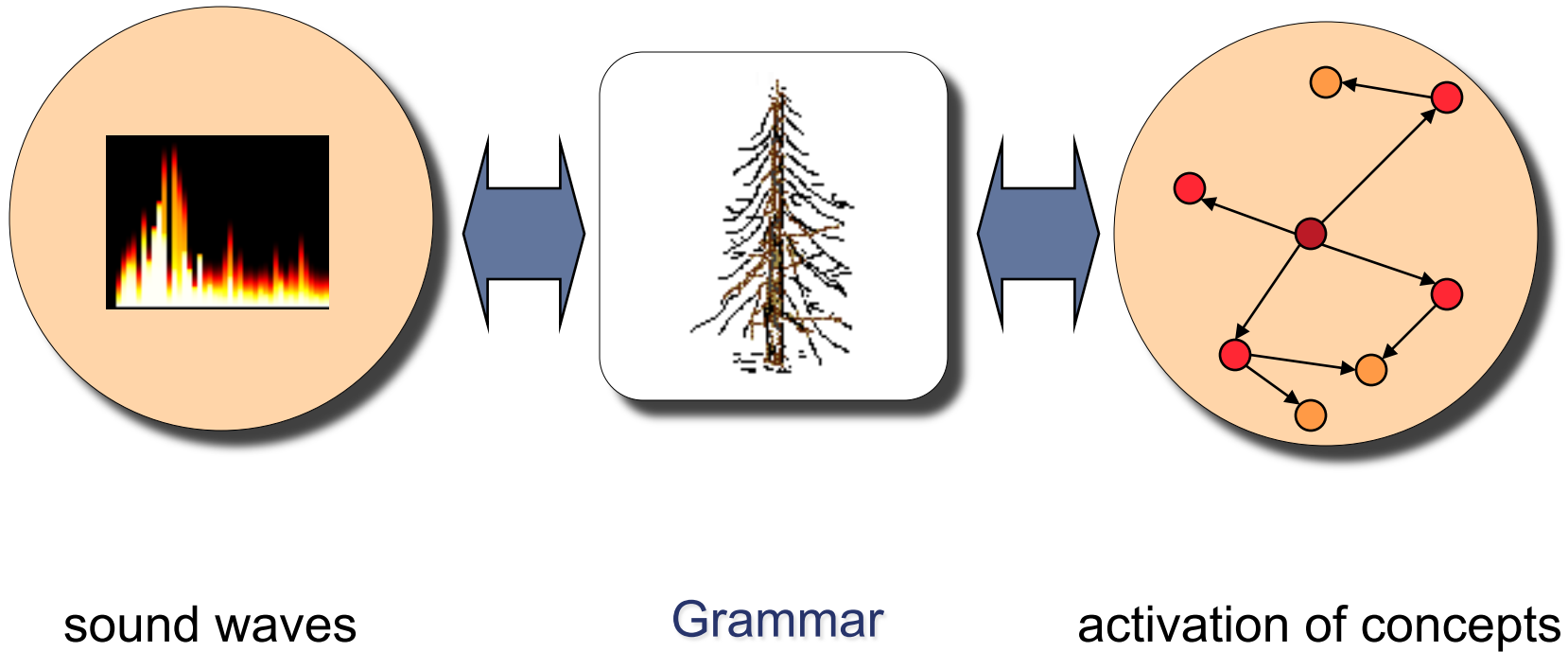
# WHAT HAPPENS IN BETWEEN?



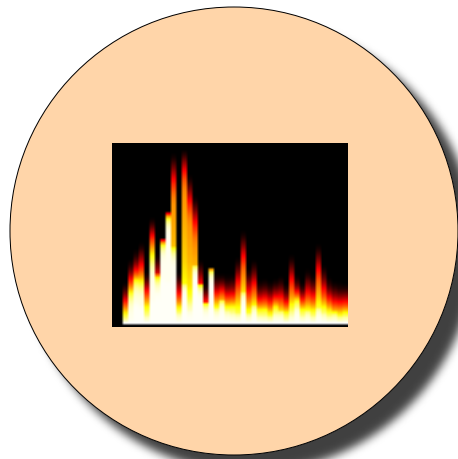
# WHAT HAPPENS IN BETWEEN?



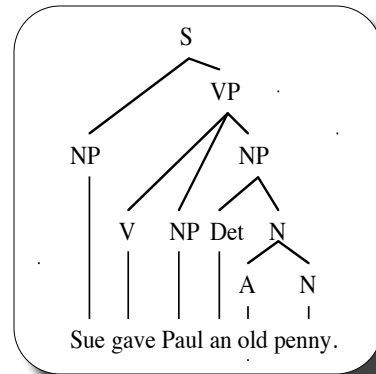
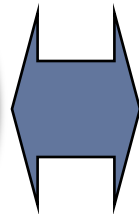
# WHAT HAPPENS IN BETWEEN?



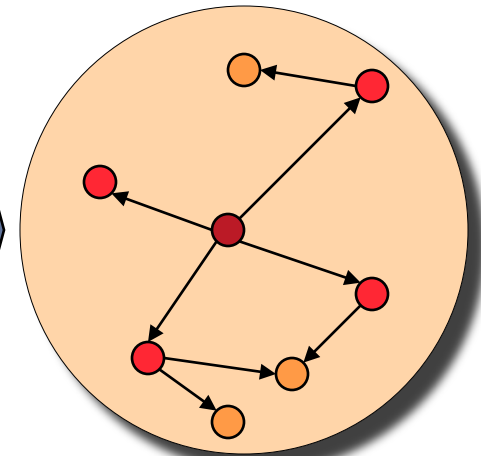
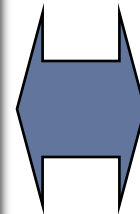
# WHAT HAPPENS IN BETWEEN?



sound waves

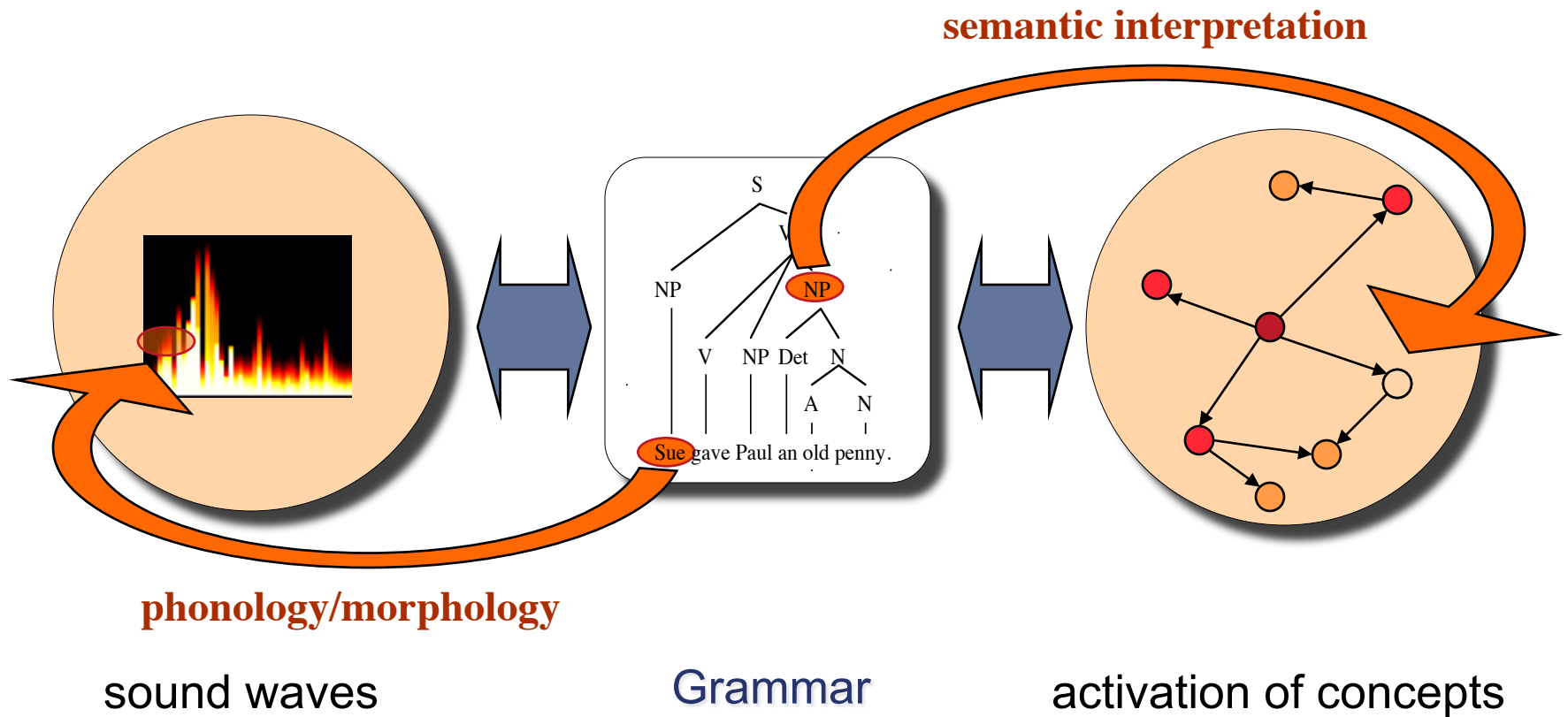


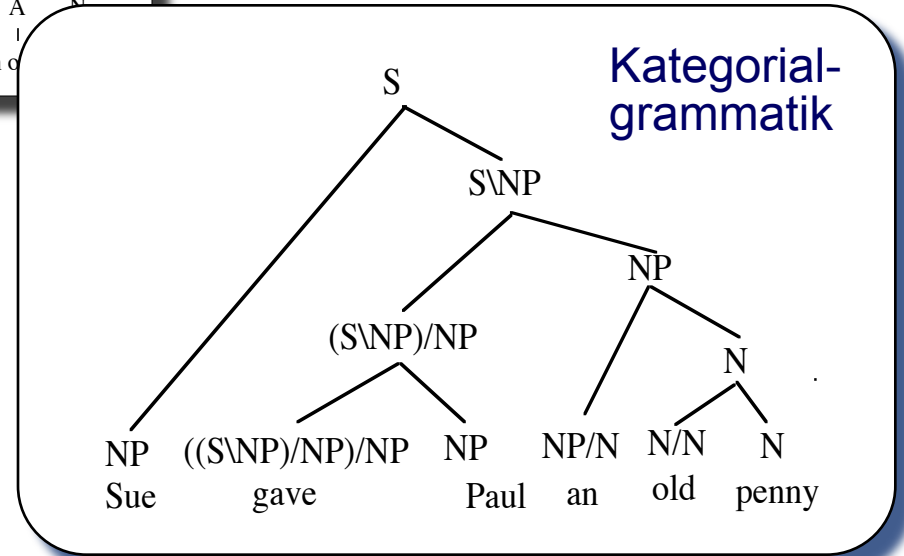
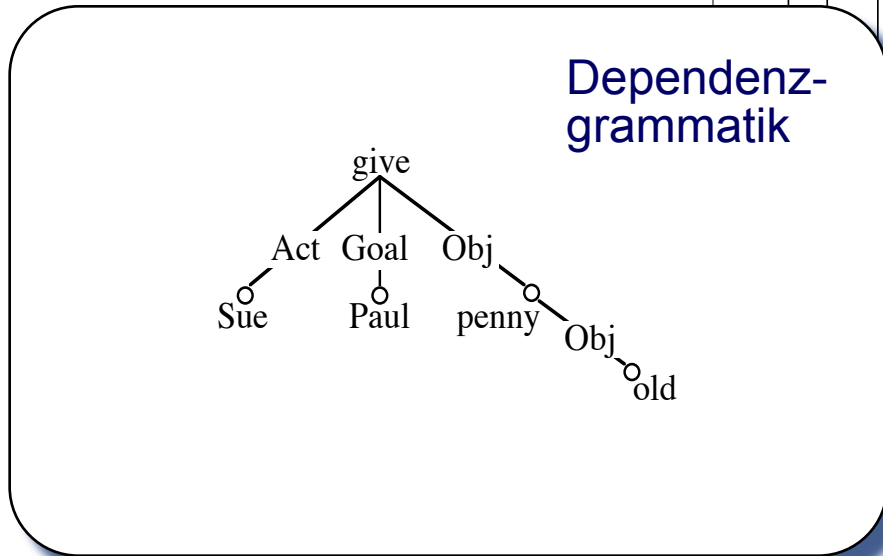
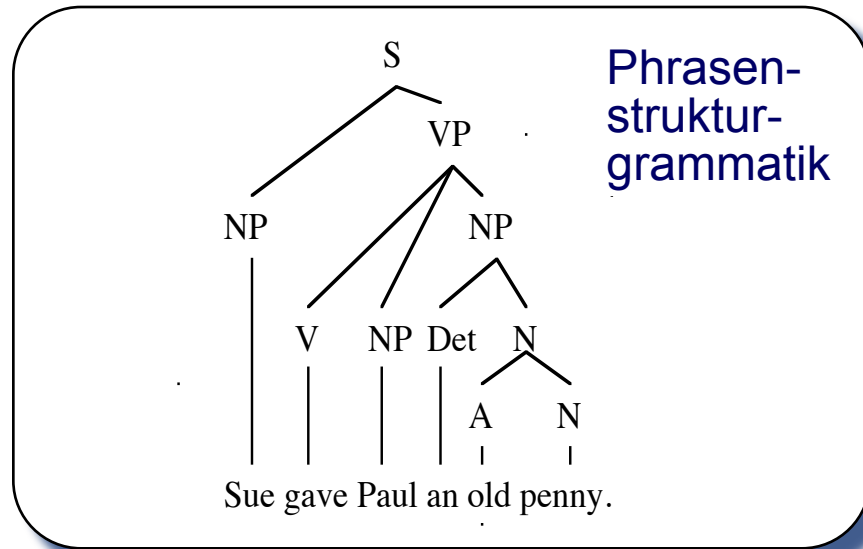
Grammar

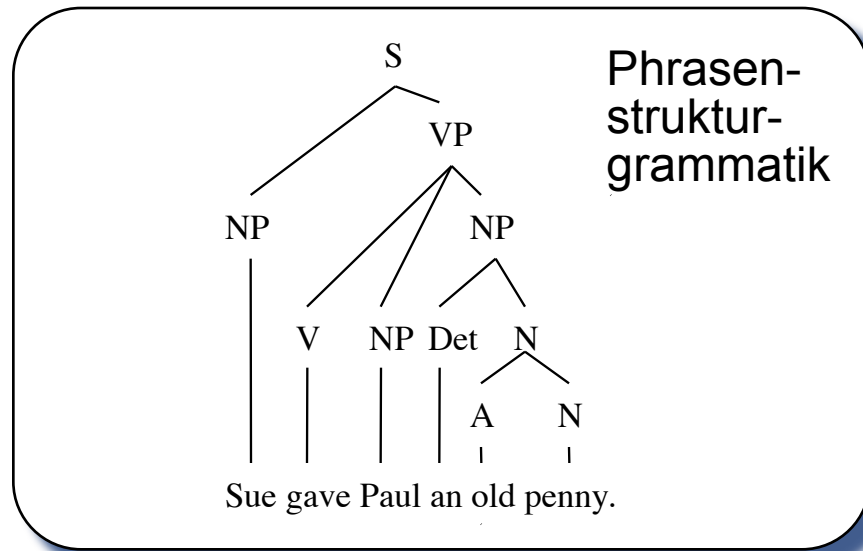


activation of concepts

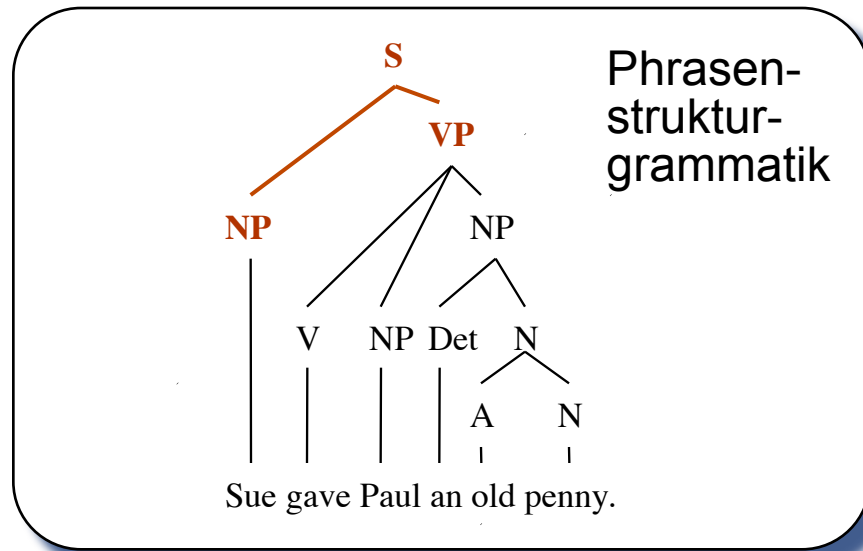
# WHAT HAPPENS IN BETWEEN?



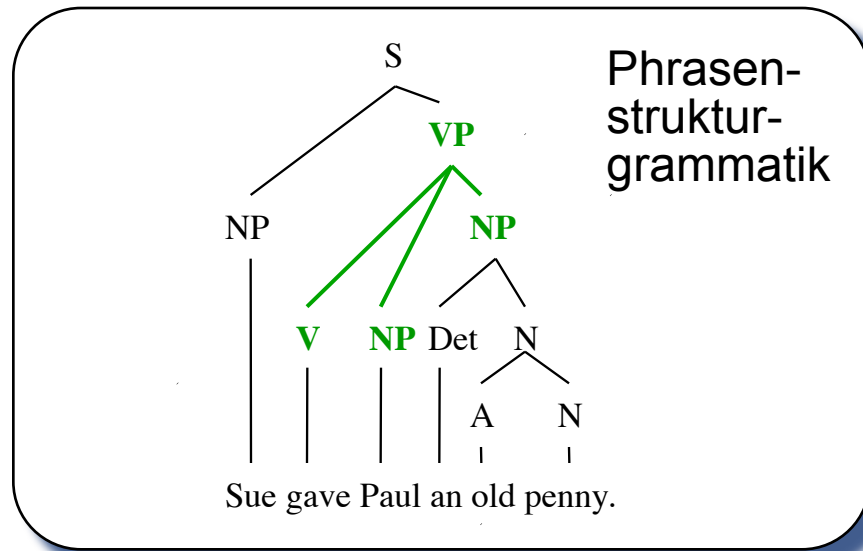




$S \rightarrow NP VP$

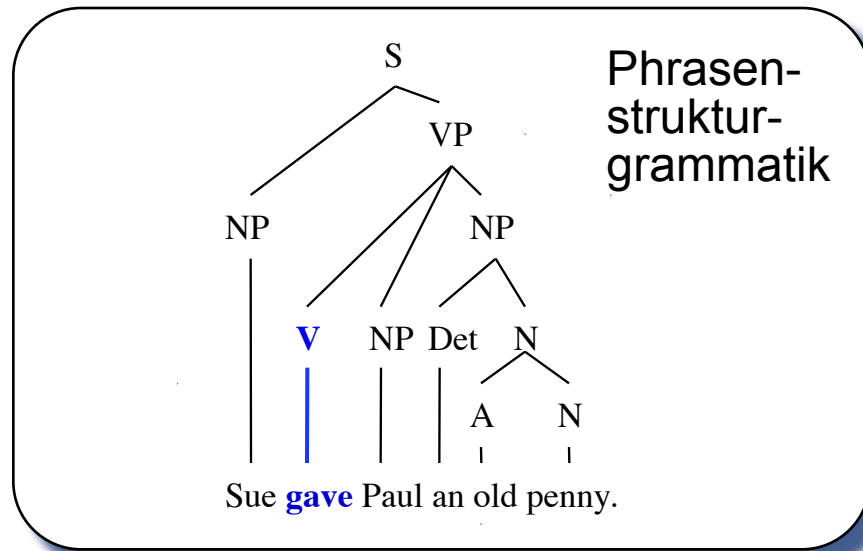


$S \rightarrow NP VP$



$S \rightarrow NP VP$

$VP \rightarrow V NP NP$



$S \rightarrow NP VP$

$VP \rightarrow V NP NP$

$V \rightarrow \text{gave}$



Grammar Theory  
Grammatiktheorie

Grammar Formalism  
Grammatikformalismus

Formal Grammar  
Formale Grammatik

Implementation of a GF  
GF-Implementierung



	Einzelsymbole	Ketten
nichtterminale	A, B, C, ...	... , X, Y, Z
terminale	a, b, c, ...	... , x, y, z
unspezifiziert	$\alpha$ , $\beta$ , $\gamma$ , ...	... , $\phi$ , $\chi$ , $\psi$ , $\omega$
Startsymbol	S	
die leere Kette	e	
Zahlen	..., i, j, k, l, m, n, ...	



Eine Sprache  $L$  über einem Alphabet (Vokabular)  $\Sigma$  ist eine Teilmenge von  $\Sigma^*$ .

Eine formale Grammatik  $G_L$  für eine Sprache  $L$  ist ein  
Quadrupel  $(V_N, V_T, \{S\}, P)$ .

$V_N$  - nichtterminales Vokabular, Hilfsalphabet

$V_T$  - terminales Vokabular

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

$\{S\}$  - Einermenge mit dem Startsymbol, Axiomenmenge

$P$  - Menge der Produktionen, Regelmenge

Menge von Regeln der Form  $\omega_1 \phi \omega_2 \rightarrow \omega_1 \psi \omega_2$

meist geschrieben als  $\phi \rightarrow \psi$



## Relation "folgt":

Wenn  $G = (V_N, V_T, \{S\}, P)$ , dann folgt  $\psi$  aus  $\phi$  nach  $G$  genau dann, wenn es Ketten  $\phi_1, \phi_2, \chi, \omega$  gibt, so daß  $\phi = \phi_1 \chi \phi_2$  und  $\psi = \phi_1 \omega \phi_2$  und  $\chi \rightarrow \omega \in P$ .

Notation:  $\phi \underset{G}{\Rightarrow} \psi$

## Ableitung:

Eine Folge von Ketten  $\phi_1, \phi_1, \dots, \phi_n$  ist eine Ableitung nach  $G$  g.d.w.  $\phi_i \underset{G}{\Rightarrow} \phi_{i+1}$  für alle  $i, 1 \leq i \leq n$ .

Wenn es eine Ableitung nach  $G$  von  $\phi$  nach  $\psi$  gibt, können wir das so schreiben:

$$\phi \underset{G}{\overset{*}{\Rightarrow}} \psi$$

Die Relation ableitbar ist transitiv und wird als reflexiv definiert.



Die Sprache L: Eine Kette  $\omega$  ist nach  $G_L$  in der Sprache L g.d.w. die folgenden drei Bedingungen erfüllt sind:

1.  $\omega \in V_T^*$

2. 
$$S \underset{G}{\overset{*}{\Rightarrow}} \omega$$

3. Es gibt kein  $\chi$ , so daß 
$$\omega \underset{G}{\overset{*}{\Rightarrow}} \chi \quad \text{und } \omega \neq \chi.$$

Man sagt auch  $G_L$  erzeugt die Sprache L. Die von G erzeugte Sprache L wird auch als  $L(G)$  geschrieben.

Schwache Äquivalenz: Zwei Grammatiken  $G_1$  und  $G_2$  sind schwach äquivalent, wenn sie dieselbe Sprache erzeugen.



## Typ 0 (unbeschränkte Ersetzungsregelsysteme):

Jede Grammatik, die die Definition einer formalen Grammatik erfüllt, ist vom Typ 0.

## Typ 1 (kontextsensitive Grammatiken):

Jede Produktion hat die Form  $\phi A \psi \rightarrow \phi \omega \psi$ , wobei  $A \in V_N$ ,  $\omega \neq \varepsilon$ .

## Typ 2 (kontextfreie Grammatiken):

Jede Produktion hat die Form  $A \rightarrow \omega$ , wobei  $\omega \neq \varepsilon$ .

## Typ 3 (reguläre Grammatiken):

Jede Produktion hat die Form  $A \rightarrow x B$  oder  $A \rightarrow x$ , wobei  $x \neq \varepsilon$ .



S  
NP VP  
DET ADJ N VP  
DET ADJ N V NP  
DET ADJ N V DET ADJ N

.  
. .  
. .

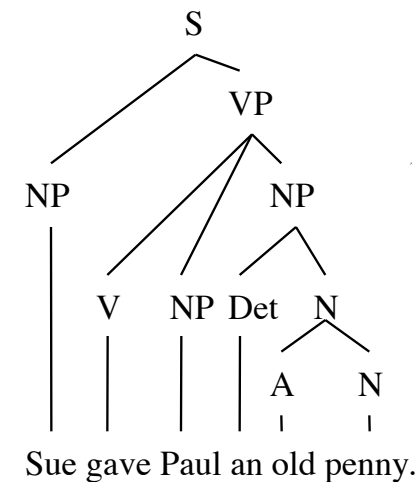
ein kleines Mädchen sucht ein kleines Mädchen



## der Begriff des syntaktischen Strukturbaums

kodierte Information

1. Die hierarchische Organisation der Teile eines Satzes in Konstituenten.
2. Der Zugehörigkeit jeder Konstituente in eine syntaktische (linguistische) Klasse (Kategorie).
3. Die lineare Abfolge der Konstituenten.





**Relationen:** unmittelbare Dominanz - Dominanz  
unmittelbare Präzedenz - Präzedenz

**Konstituentenstrukturbaum: Quintupel** (N, Q, D, P, L)

N - endliche Menge von Knoten

Q - endliche Menge von Etiketten

D - schwache Teilordnung in  $N \times N$ , die Dominanzrelation  
(reflexiv, transitiv und antisymmetrisch)

P - starke Teilordnung in  $N \times N$ , die Präzedenzrelation  
(irreflexiv, transitiv und asymmetrisch)

L - Funktion von N in Q, die Etikettierfunktion

**Bedingungen:**

Wurzelbedingung

Exklusivitätsbedingung

Kreuzungsfreiheit



## Wurzelbedingung

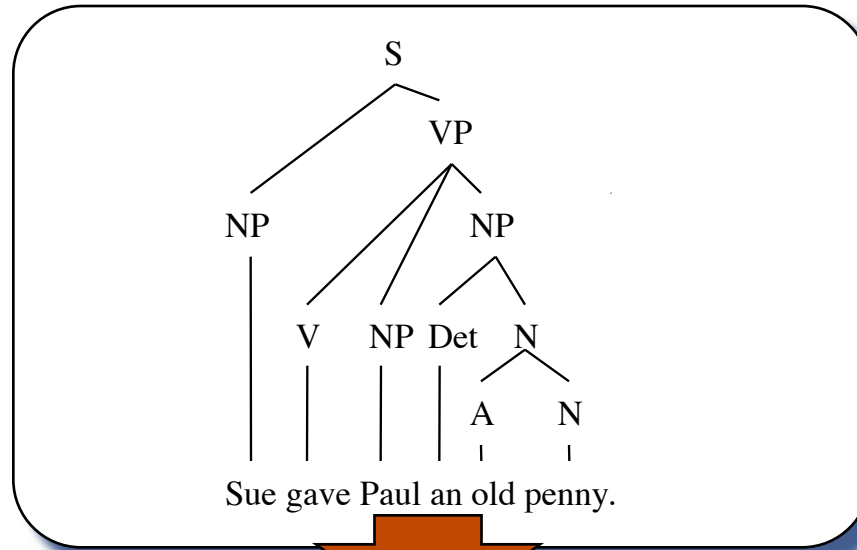
Es gibt genau einen Knoten, der alle Knoten des Baumes dominiert.

## Exklusivitätsbedingung

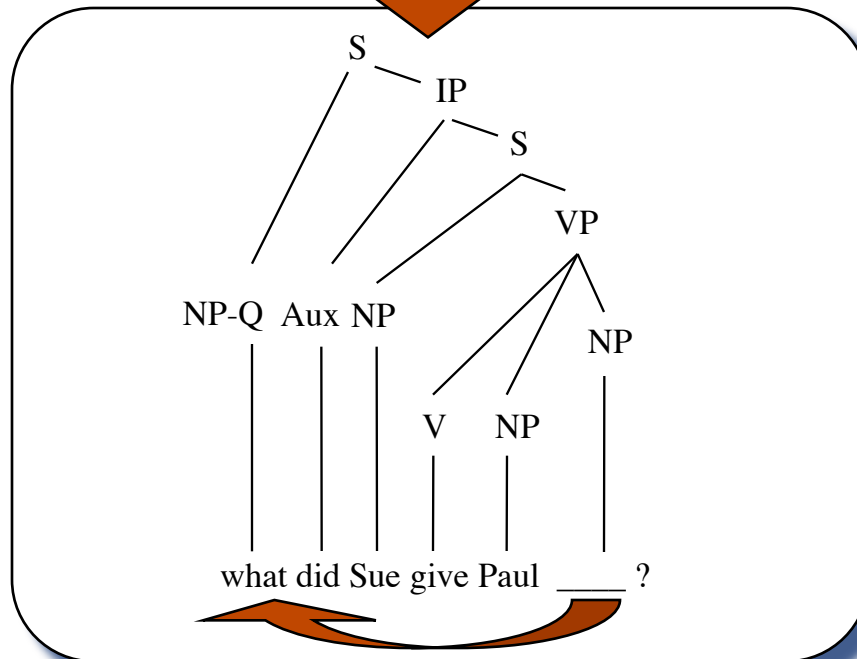
Für jegliche zwei Knoten  $x$  und  $y$  gilt, entweder  $D(x,y)$  oder  $D(y,x)$   
oder  $P(x,y)$  oder  $P(y,x)$

## Kreuzungsfreiheit

Wenn  $P(x,y)$  dann gilt für alle  $x'$ , die von  $x$  dominiert werden  $D(x,x')$  und für alle  $y'$ , die von  $y$  dominiert werden  $D(y, y')$  dass auch  $x'$   $y'$  vorausgeht [ $P(x', y')$ ]

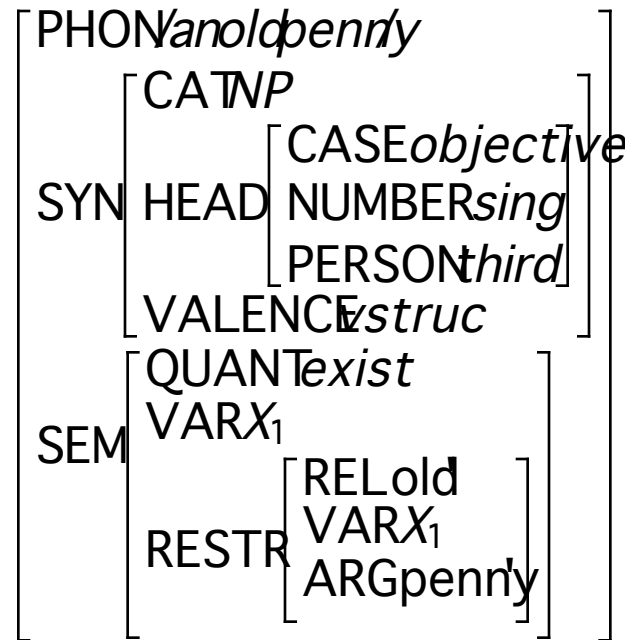
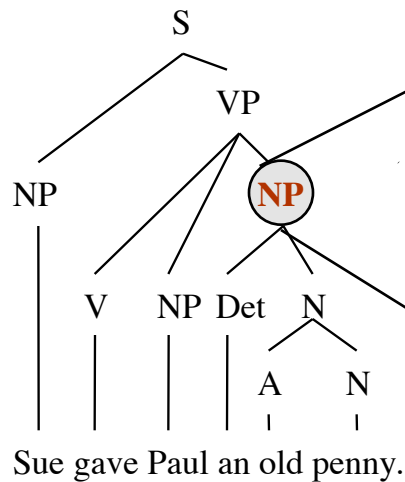


Transformations-  
grammatik





## Unifikationsgrammatik





## Formal grammars, formal languages and their corresponding automata

### Chomsky hierarchy

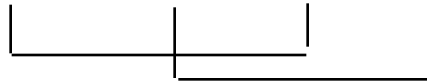
Grammars	Languages	Minimal Automaton
Type-0 Unrestricted	Recursively enumerable	Turing machine
n/a (no common name)	Recursive	Decider
Type-1 Context-sensitive	Context-sensitive	Linear-bounded Automaton
Type-2 Context-free	Context-free	Pushdown Automaton
Type-3 Regular	Regular	Finite-State Automaton

## Eine kurze Chronologie der Kontroverse

- 57-70: Chomsky, Postal u.a. behaupten, daß KFGs are unzureichend für natürliche Sprachen
- 63: Gilbert Harrman verteidigt Phrasenstrukturgrammatik
- 71&73: Peters und Ritchie zeigen, daß TG alle Sprachen vom Typ 0 erzeugt
- 79: Gazdar schlägt eine Zwei-Stufen PSG (GPSG) vor, die adäquat für das Englische ist
- 82: Gazdar & Pullum zeigen, daß alle bisherigen "Beweise" für die Nichtkontextfreiheit fehlerhaft sind
- 82: Uszkoreit und Peters beweisen, daß die damalige GPSG alle Sprachen vom Typ 0 erzeugt dieser Umstand wird von GKPS durch den finite-closure constraint beseitigt
- 85: Shieber beweist, daß Zürcher Deutsch nicht KF ist, Culy zeigt das gleiche für Bambara
- 85: ähnliche Behauptungen von Higginbotham and Langendoen & Postal für das Englische wurden durch Pullum erfolgreich widerlegt

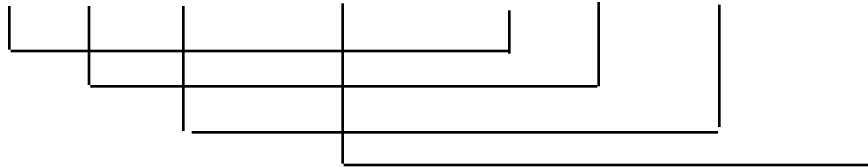


Jan säit das mer em Hans es huus hälfed aastriche



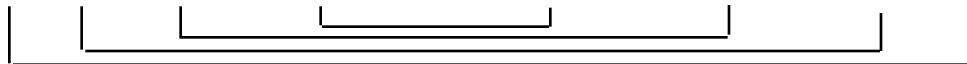
...dat Jan Piet Marie de kinderen zag helpen laten zwemmen

daß Jan Piet Marie den Kindern sah helfen lassen schwimmen

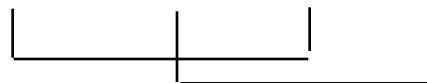


daß Jan Piet Marie helfen sah, die Kinder schwimmen zu lassen

daß Jan Piet Marie die Kinder schwimmen zu lassen helfen sah



Jan säit das mer em Hans es huus hälfed aastriche





- Mirror human performance.
- Improve machine performance.
- Understand human processing.
- Understand why language is as it is.



- Linguistic Competence:

The knowledge a speaker has to possess in order to master a language.

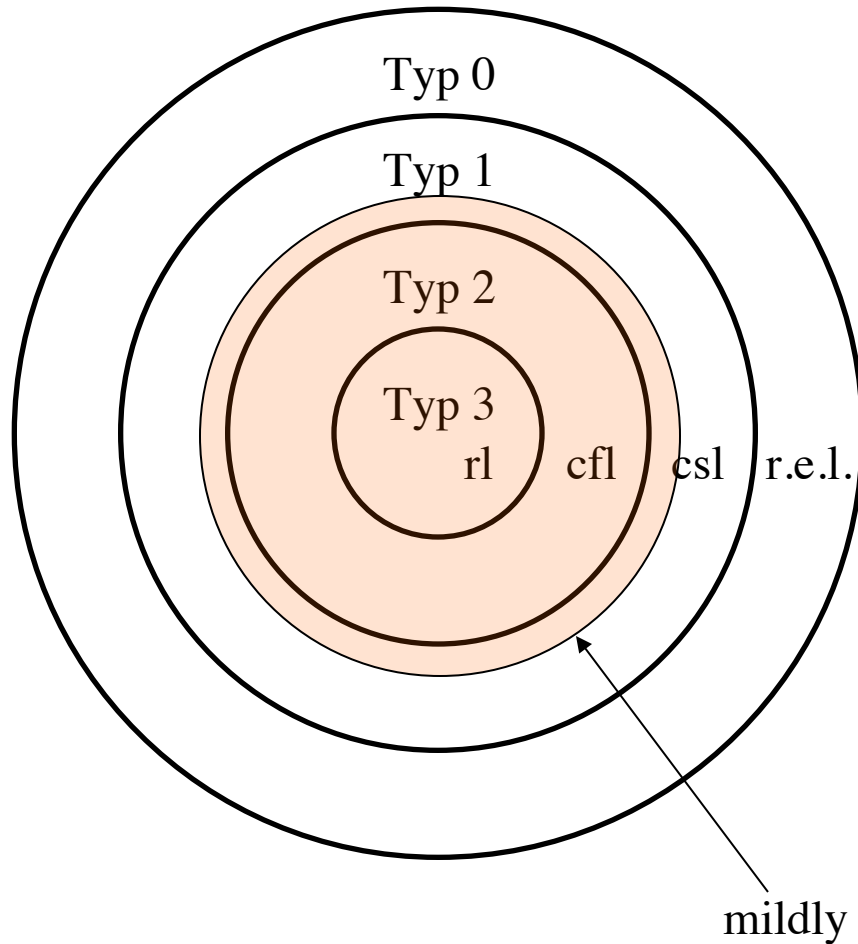
The system of rules, principles and constraints that constitute the grammar of a language

The finite definition of an infinite natural language.

- Linguistic Performance

The mechanisms and processes underlying actual human language use, i.e., sentence production and comprehension.

This includes the influence (assisting or limiting) of other cognitive processes such as reasoning, perception and action as well as other tasks.



- Typ 0: recursively enumerable sets
- Typ 1: contextsensitive languages
- Typ 2: context-free languages
- Typ 3: regular languages