

# Semantikkonstruktion

Vorlesung “Computerlinguistische Techniken”

Alexander Koller

8. Januar 2016

# Semantische Verarbeitung

- Uraltes Problem: *Inferenz*.
  - ▶ Woher wissen wir, dass ein Satz aus einem anderen folgt?
  - ▶ Können wir das automatisch berechnen?

Alle Menschen sind sterblich.

Sokrates ist ein Mensch.

---

Also ist Sokrates sterblich.

# Anwendungen

- Inferenz wäre für viele Anwendungen der CL nützlich, z.B. automatisches Zusammenfassen und Question Answering.

Which genetically caused connective tissue disorder has severe symptoms and complications regarding the aorta and skeletal features, and, very characteristically, ophthalmologic subluxation?

Marfan's is created by a defect of the gene that determines the structure of Fibrillin-11. One of the symptoms is displacement of one or both of the eyes' lenses. The most serious complications affect the cardiovascular system, especially heart valves and the aorta.

# Formale Repräsentationen

- Grundansatz in Semantik und Computerlinguistik:
  - ▶ Berechne *Bedeutungsrepräsentation* in einer geeigneten formalen Sprache (z.B. Prädikatenlogik),
  - ▶ so dass die formale Repräsentation relevante Informationen über die Satzbedeutung erfasst (z.B. Wahrheitsbedingungen).

Alle Menschen sind sterblich.

Sokrates ist ein Mensch.

---

Also ist Sokrates sterblich.

$\forall x. \text{mensch}(x) \rightarrow \text{sterblich}(x)$

$\text{mensch}(s)$

---

$\text{sterblich}(s)$

# Standardmodell

- In den 1990ern entwickelt sich “Standardmodell” der semantischen Verarbeitung:
  - ▶ Kompositionale Semantikkonstruktion, z.B. mit Lambda-Kalkül oder Unifikationsgrammatiken.
  - ▶ Ergebnis ist eine Formel in Prädikatenlogik 1. Stufe, die die Wahrheitsbedingungen erfassen soll.
  - ▶ Inferenzproblem auf Prädikatenlogik mit *Theorembeweisern* lösen.
- Siehe z.B. das Lehrbuch von Blackburn & Bos.

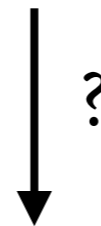
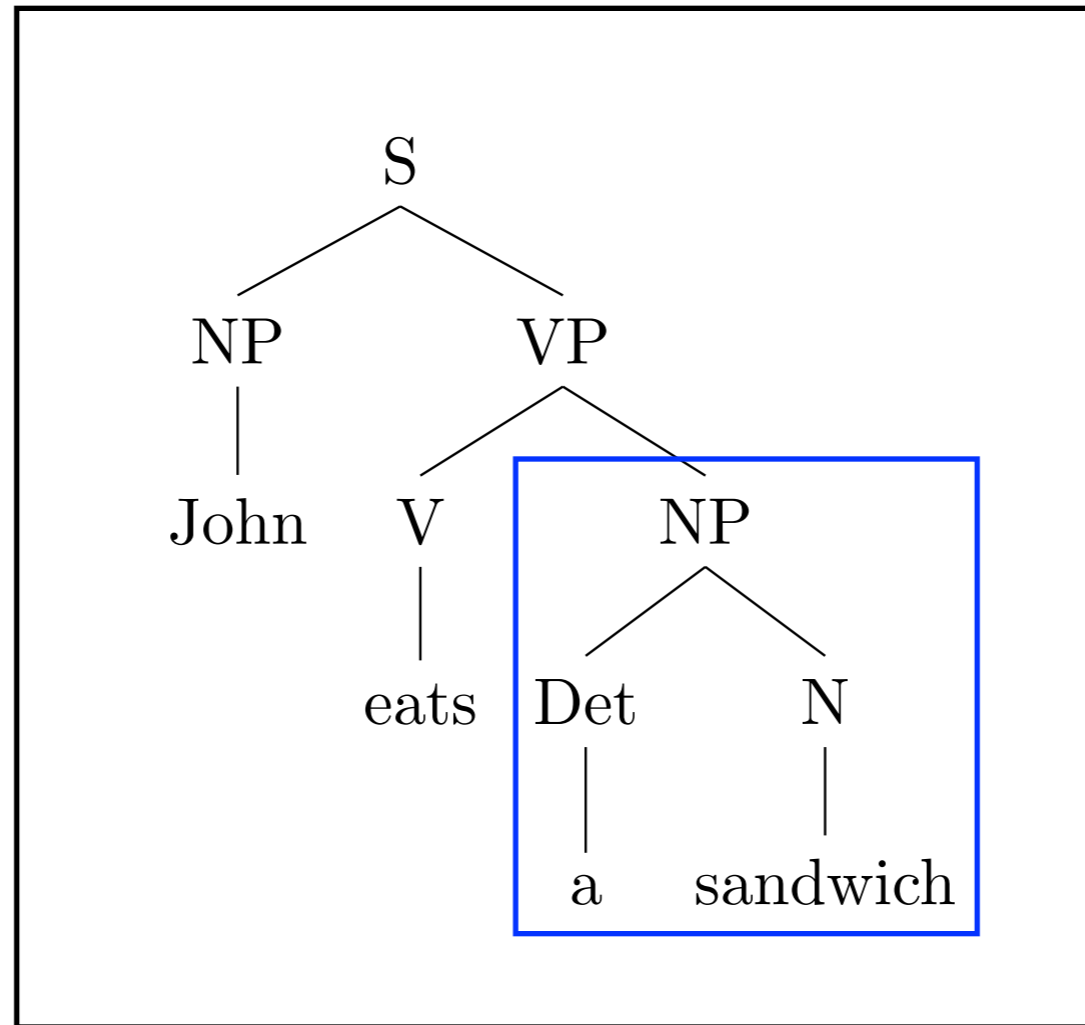
# Hauptfragen

- Wie können wir für einen gegebenen Satz eine formale Bedeutungsrepräsentation berechnen?
- Wie können wir das Wissen formalisieren, das der Theorembeweiser für Inferenzen braucht?
- Wie können wir Semantikkonstruktion und Theorembeweisen effizient machen?

# Semantikkonstruktion

- *Semantikkonstruktion* ist das Problem, aus einem Syntaxbaum eine Bedeutungsrepräsentation zu berechnen.
- Breiter Konsens, dass Semantikkonstruktion *kompositionell* vorgehen sollte.
  - ▶ Heißt: Bedeutung einer Phrase bestimmt sich aus Bedeutungen ihrer Teile sowie der Grammatikregel.
  - ▶ Erlaubt Bottom-Up-Auswertung des Syntaxbaums, was in der CL sehr bequem ist.
  - ▶ Manche semantischen Phänomene allerdings schwierig kompositionell zu machen, z.B. Koreferenz.

# Ein Beispiel

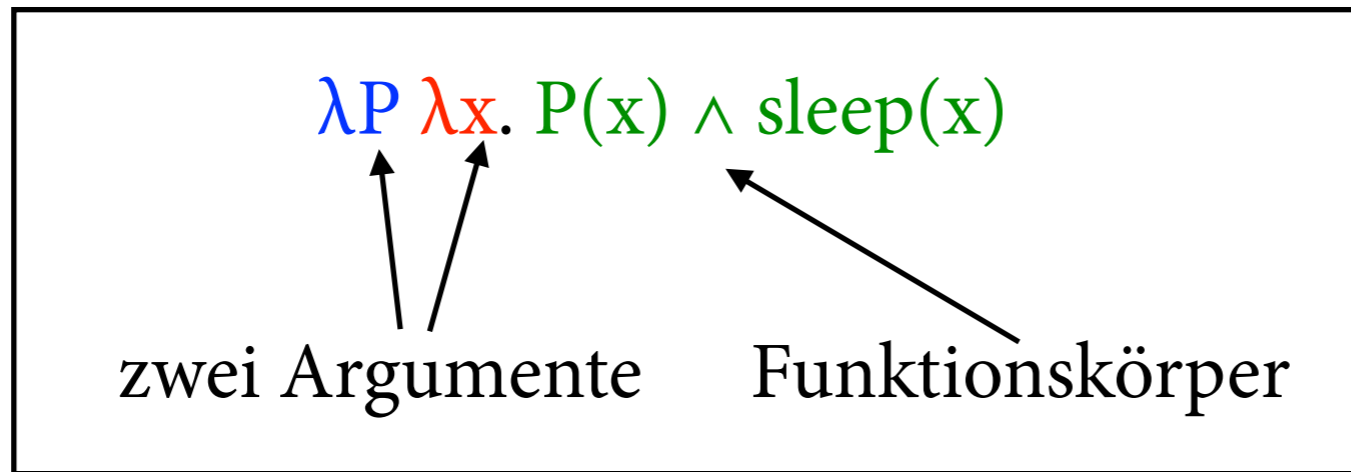


Zielrepräsentation:  
 $\exists x. \text{sandwich}(x) \wedge \text{eat}(x)(j)$



# Der Lambda-Kalkül

- Term-Notation für Funktionen:



- Applikationen der Form  $(\lambda x M(x))(N)$  kann man mit  $\beta$ -Reduktion vereinfachen.

$$(\lambda P \lambda x. P(x) \wedge \text{sleep}(x)) (\text{boy}) \rightarrow_{\beta} \lambda x. \text{boy}(x) \wedge \text{sleep}(x)$$

$$(\lambda x. \text{boy}(x) \wedge \text{sleep}(x)) (\text{a}) \rightarrow_{\beta} \text{boy}(\text{a}) \wedge \text{sleep}(x)$$

(im vollen Lambda-Kalkül gibt es auch  $\alpha$ - und  $\eta$ -Äquivalenz)

# Montague-Grammatik

$S \rightarrow NP VP$

$\langle S \rangle = \langle NP \rangle(\langle VP \rangle)$

$VP \rightarrow V NP$

$\langle VP \rangle = \lambda y \langle NP \rangle(\langle V \rangle(y))$

$NP \rightarrow Det N$

$\langle NP \rangle = \langle Det \rangle(\langle N \rangle)$

$NP \rightarrow John$

$\langle NP \rangle = \lambda P P(j')$

$V \rightarrow eats$

$\langle V \rangle = eat'$

$Det \rightarrow a$

$\langle Det \rangle = \lambda P \lambda Q \exists x P(x) \wedge Q(x)$

$N \rightarrow sandwich$

$\langle N \rangle = sw'$

↑  
wenn man diese  
Grammatikregel  
anwendet ...

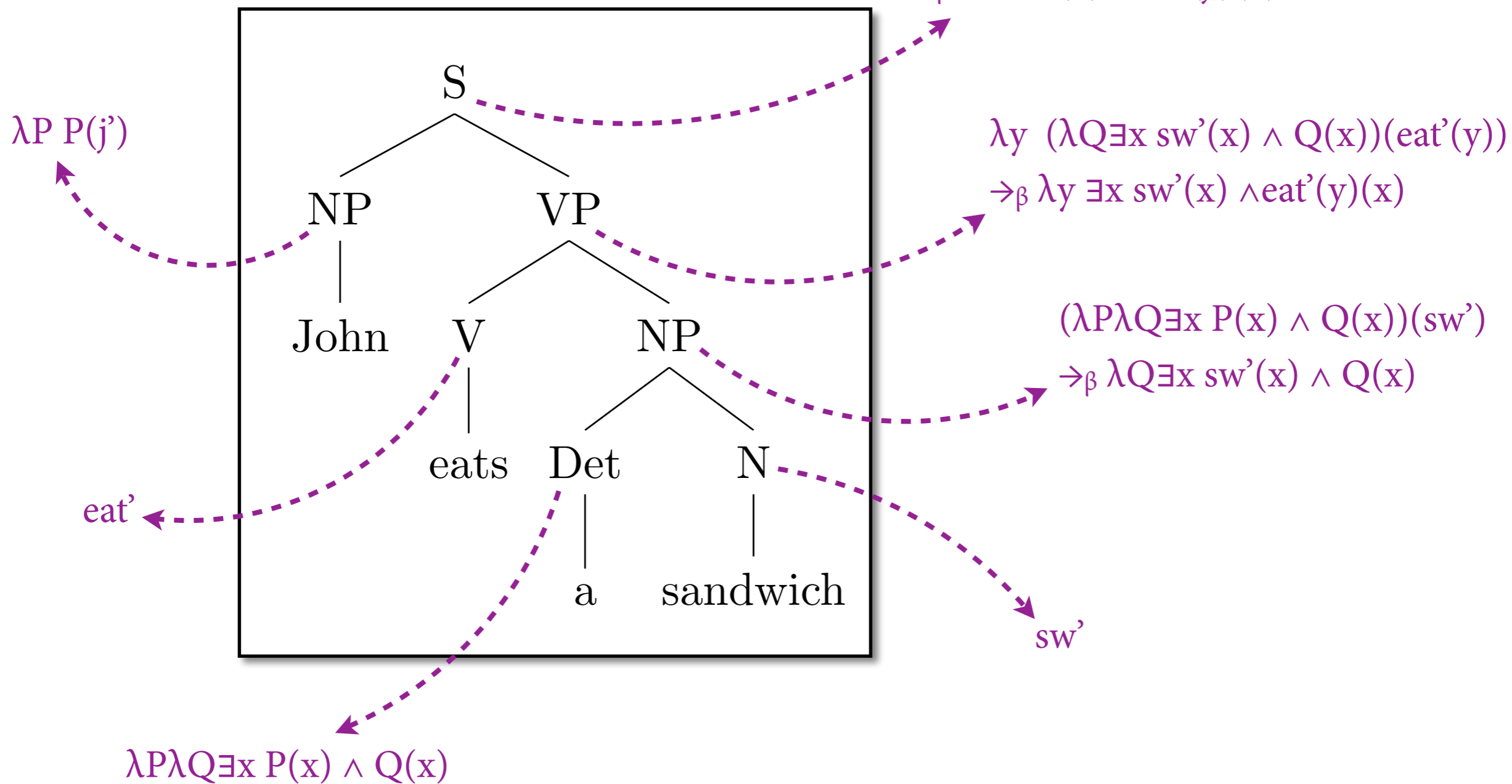
↙  
... konstruiere so einen  $\lambda$ -Term für Vater  
aus  $\lambda$ -Termen für die Kinder

# Beispiel

$(\lambda P P(j')) (\lambda y \exists x sw'(x) \wedge eat'(y)(x))$

$\rightarrow_{\beta} (\lambda y \exists x sw'(x) \wedge eat'(y)(x))(j')$

$\rightarrow_{\beta} \exists x sw'(x) \wedge eat'(j')(x)$



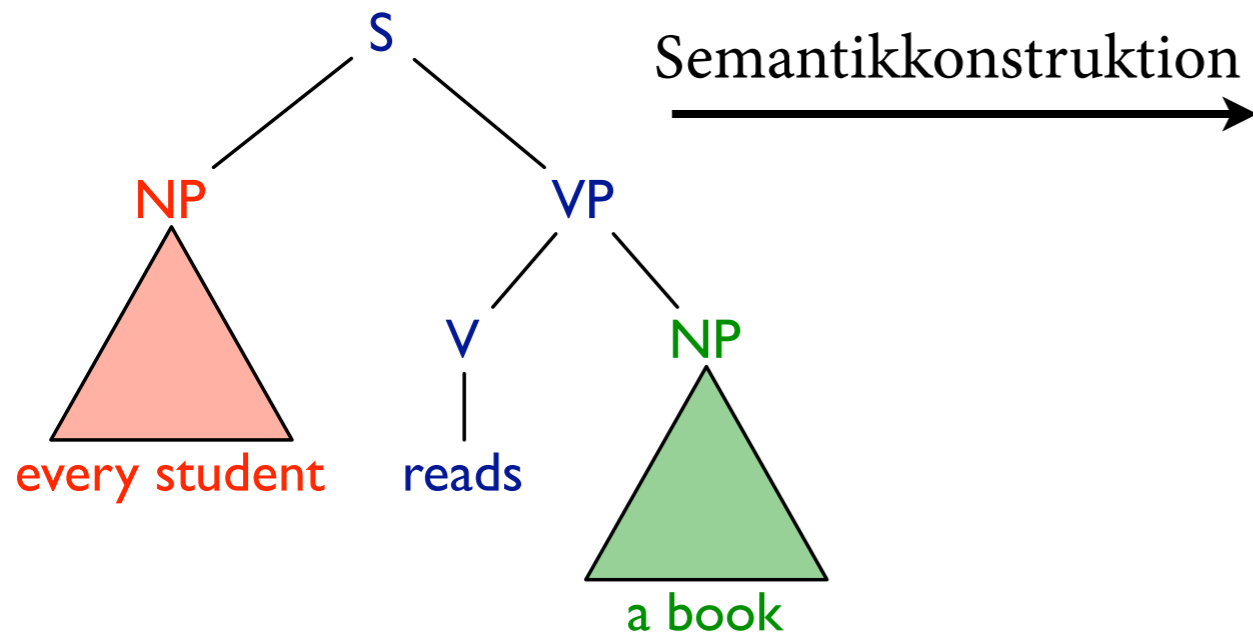
# Skopusambiguitäten

- Eine große Herausforderung bei der Semantikkonstruktion sind *semantische Ambiguitäten*.
- Prototypisches Beispiel: Skopusambiguität.
  - ▶ “Every student reads a book”
  - ▶  $\forall x \text{ student}(x) \rightarrow (\exists y \text{ book}(y) \wedge \text{read}(x,y))$
  - ▶  $\exists y \text{ book}(y) \wedge (\forall x \text{ student}(x) \rightarrow \text{read}(x,y))$
  - ▶ Funktioniert auf Deutsch nicht so gut, wegen Wortstellung.

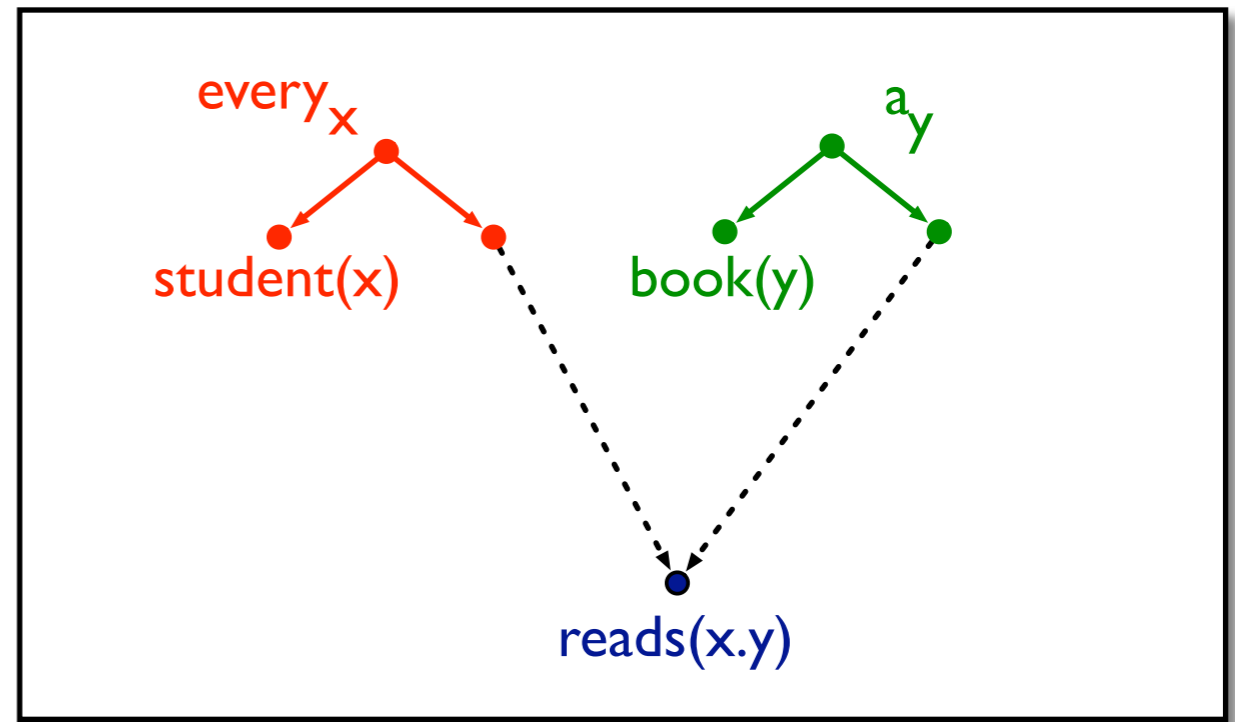
# Ansätze

- Montague-Grammatik: “Quantifying In”
  - ▶ NP wird in der Syntax angehoben, Skopusambiguität wird zu syntaktischer Ambiguität.
- Cooper-Storage
  - ▶ Während der Semantikkonstruktion kann man Quantor in einen “Speicher” legen und später wieder auslesen. SK-Prozess wird damit nichtdeterministisch.
- Unterspezifikation
  - ▶ Ergebnis der SK ist nur eine *Beschreibung* aller möglichen semantischen Repräsentationen.

# Unterspezifikation



Semantikonstruktion

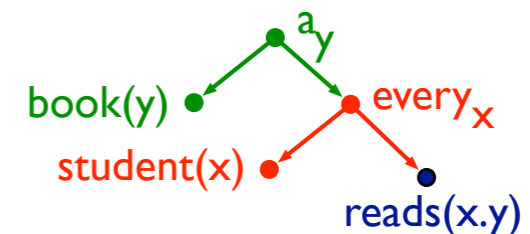
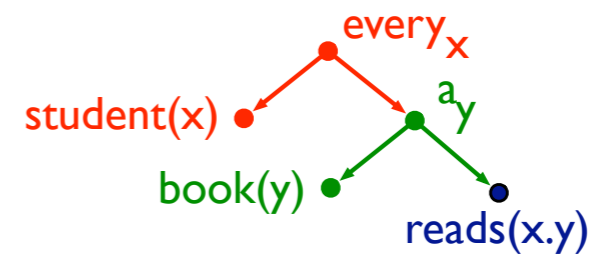


Dominanzgraph

Lösen

$every_x(student'(x), a_y(book'(y), read'(x,y)))$   
 $a_y(book'(y), every_x(student'(x), read'(x,y)))$

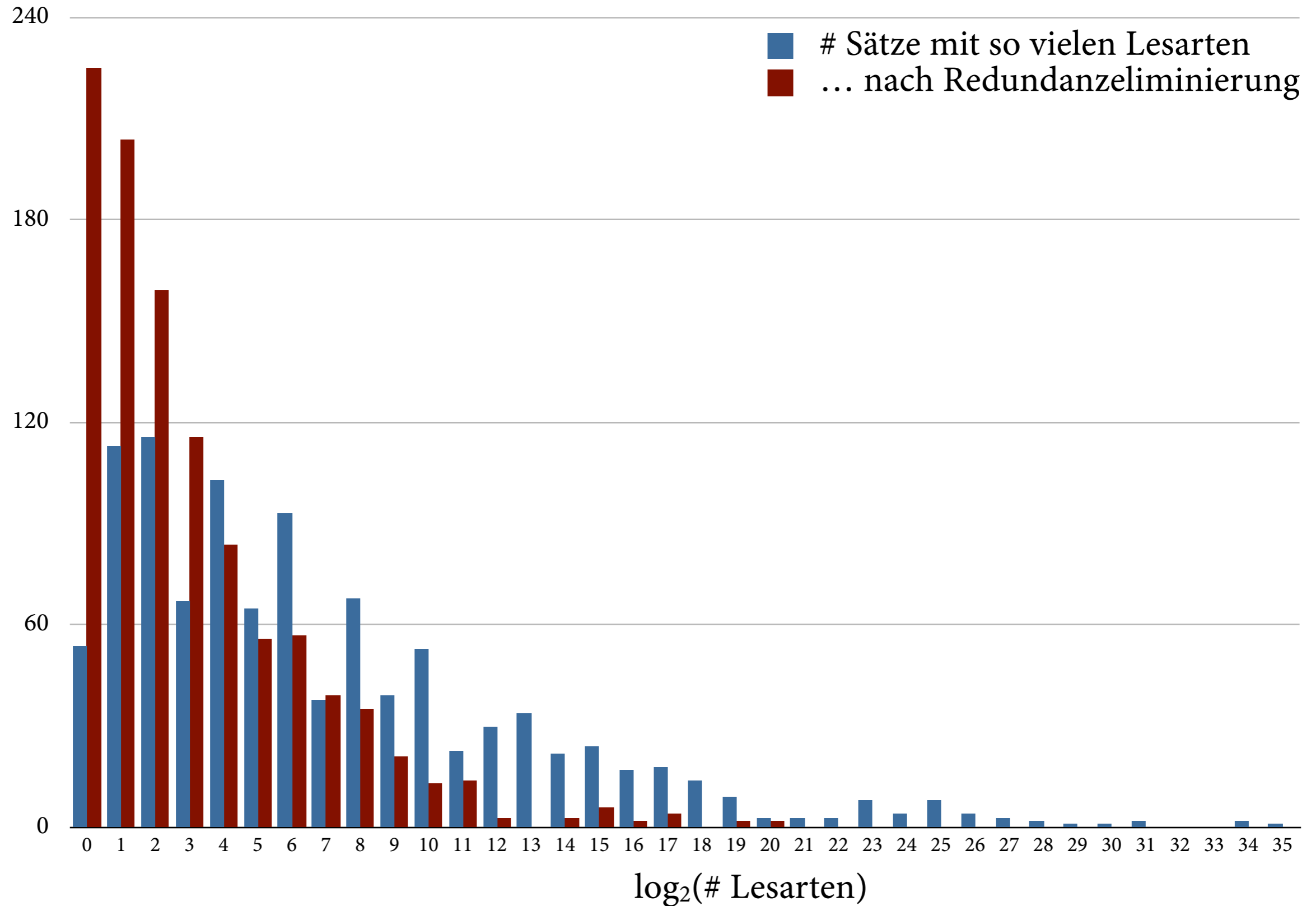
semantische Repräsentationen



# Unterspezifikation

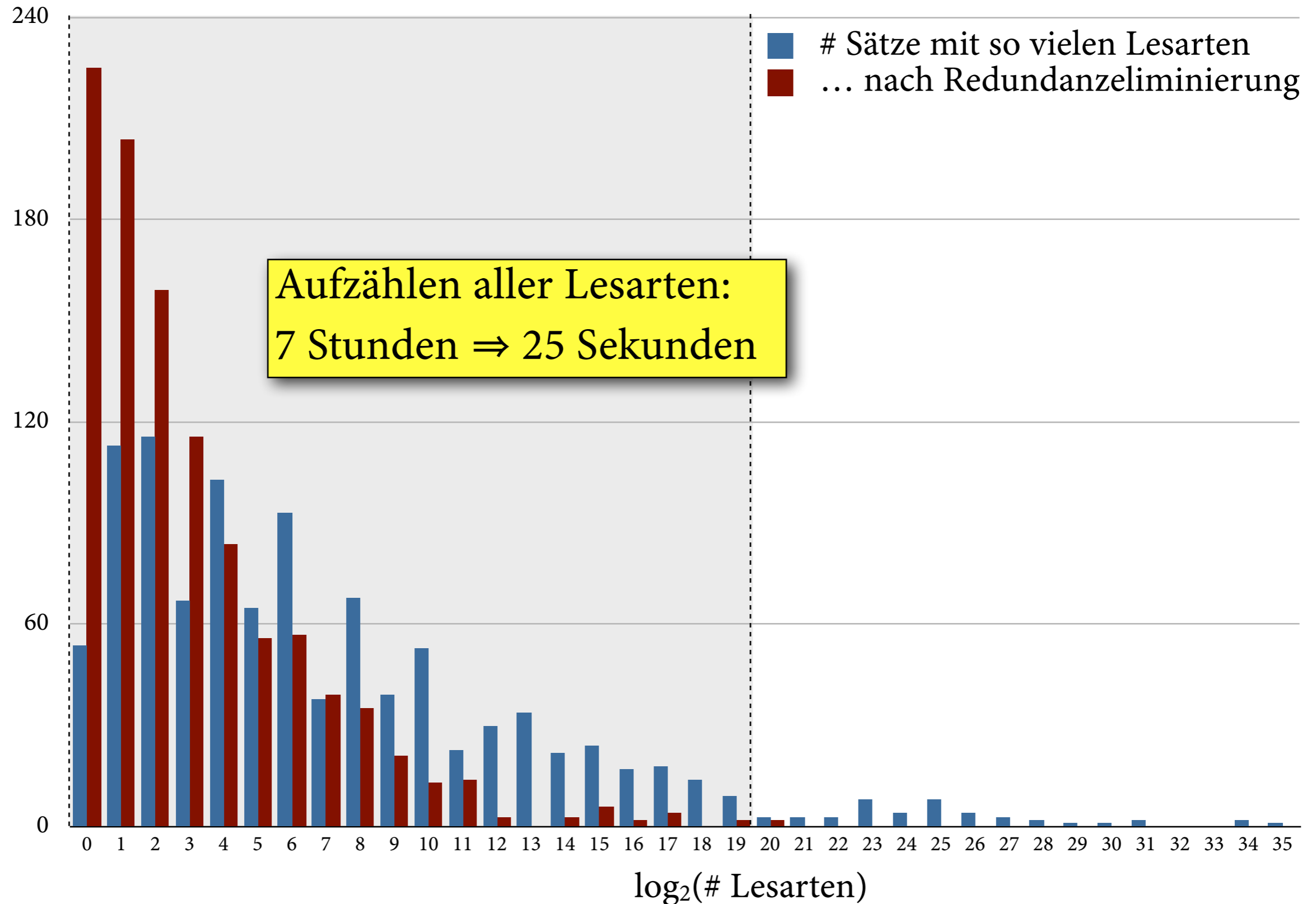
- Vorteil von Unterspezifikation ist, dass man die usp. Repräsentation weiterverarbeiten kann.
  - ▶ so verstärken, dass unerwünschte Lesarten verschwinden, bevor man sie aufzählt
  - ▶ z.B. Redundanzeliminierung: beide Lesarten von “a student reads a book” sind logisch äquivalent; eine davon reicht (siehe Koller & Thater 2010).
- Unterspezifikation ist heute Standardansatz für Semantikkonstruktion mit großen handgeschriebenen Grammatiken.

# Redundanzeliminierung

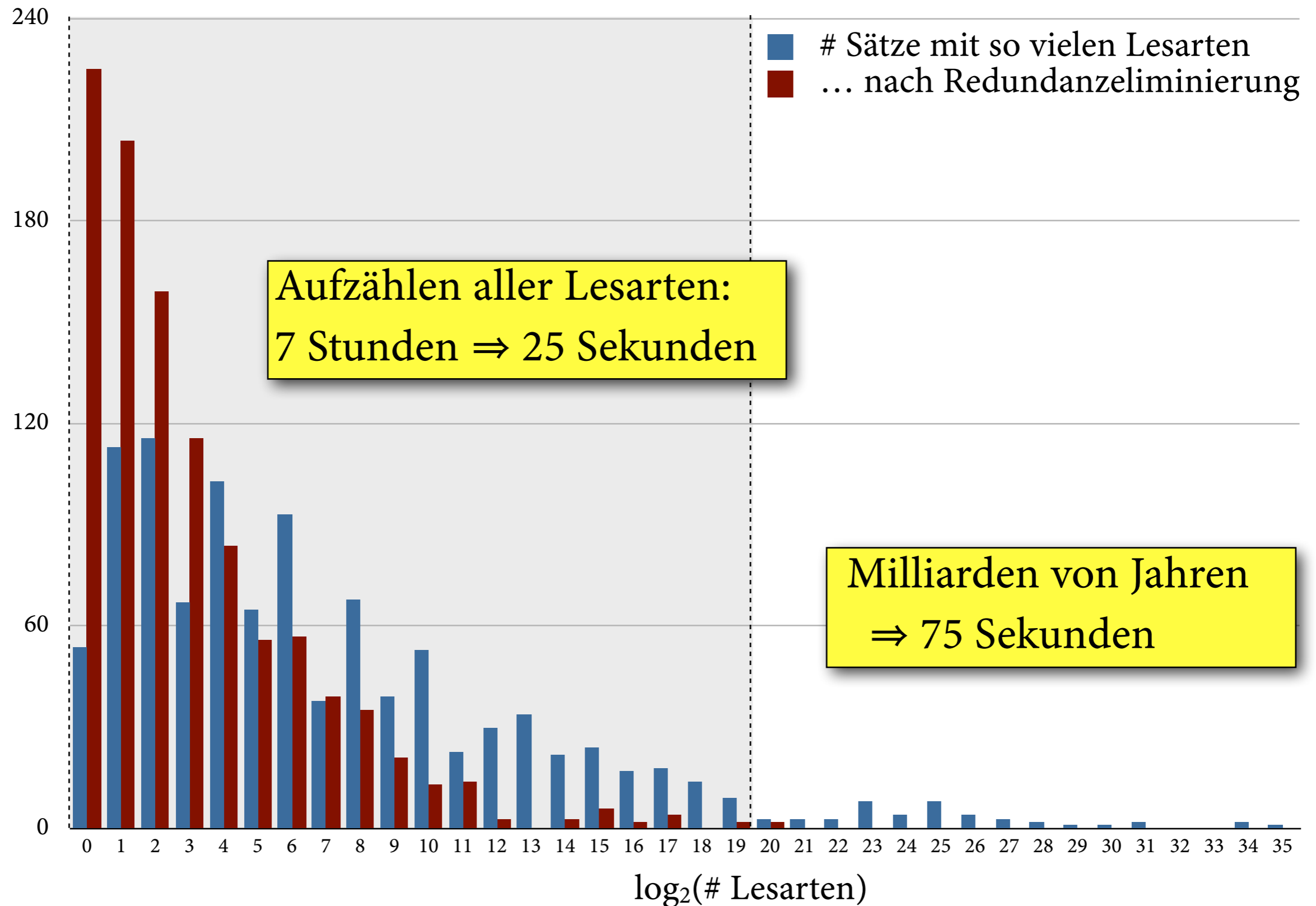




# Redundanzeliminierung



# Redundanzeliminierung



# Stand der Kunst

- Klassische Semantikkonstruktion ist ausgereift:
  - ▶ es gibt große handgeschriebene Grammatiken
  - ▶ Unterspezifikation ist in der Praxis nützlich
  - ▶ Theorembeweiser sind schnell genug
- Trotzdem heute Randthema in der CL:
  - ▶ Manuelle Grammatikentwicklung äußerst aufwändig
  - ▶ Viele Phänomene mit Prädikatenlogik nicht darstellbar
  - ▶ Breite Formalisierung von Weltwissen unpraktikabel?

# Semantic Parsing

- Einige aktuelle Forschung zu *semantic parsing*:  
Semantikkonstruktion von sprachlichem Ausdruck  
in formale Repräsentation direkt lernen.
- Zum Beispiel aus dem Geoquery-Korpus (880 Sätze):

What is the smallest state by area?

```
answer(x1, smallest(x2, state(x1), area(x1, x2)))
```

# Synchrone Grammatiken

- Eine *synchrone kontextfreie Grammatik (SCFG)* besteht aus zwei kfGen, deren Ableitungen verknüpft sind.

$S \rightarrow \text{that NP VP}$

$VP \rightarrow V \text{ NP}$

$NP \rightarrow \text{John} \mid \text{Mary}$

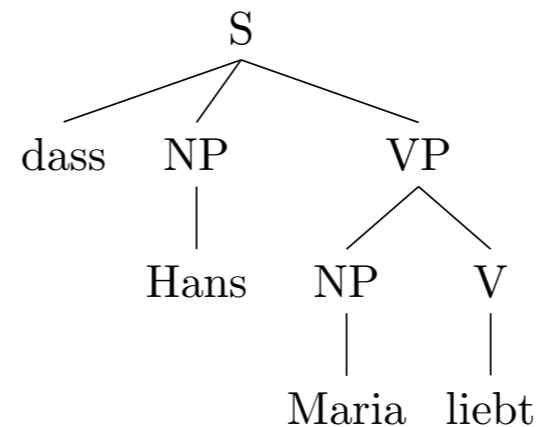
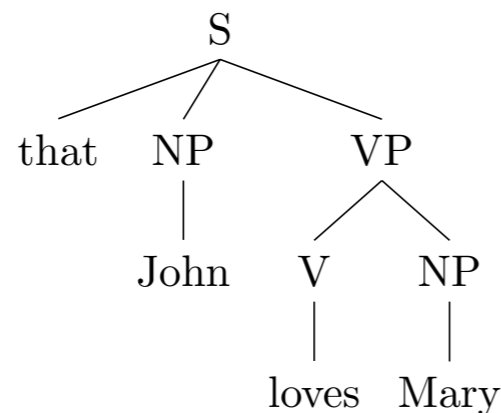
$V \rightarrow \text{loves}$

$S \rightarrow \text{dass NP VP}$

$VP \rightarrow \text{NP V}$

$NP \rightarrow \text{Hans} \mid \text{Maria}$

$V \rightarrow \text{liebt}$



# Im Semantic Parsing

- Eine  $\lambda$ -SCFG leitet gleichzeitig einen Syntaxbaum und einen  $\lambda$ -Term ab (Wong & Mooney 07).

$Q \rightarrow \text{what is the } F$

$F \rightarrow \text{smallest } F F$

$F \rightarrow \text{state}$

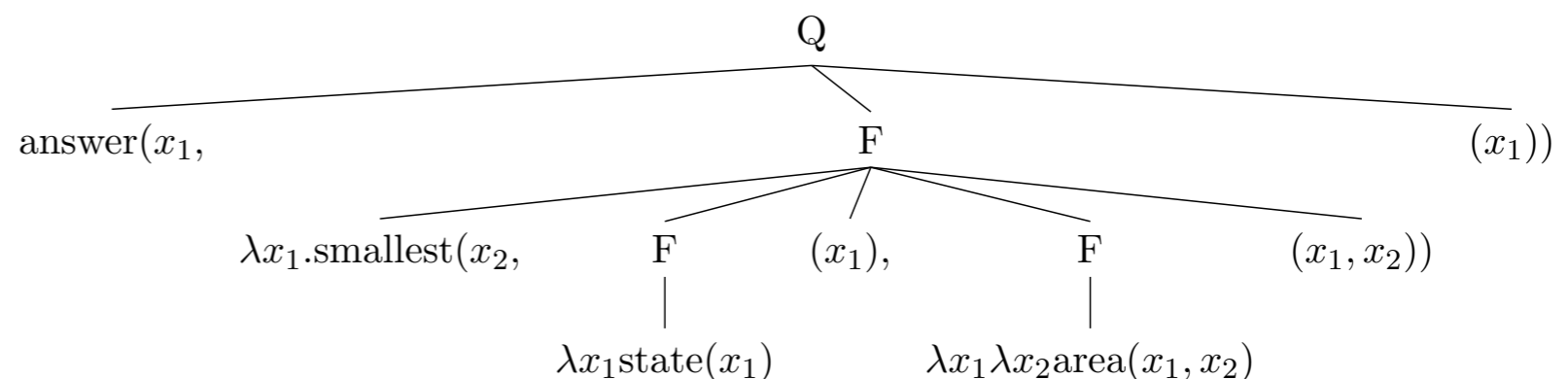
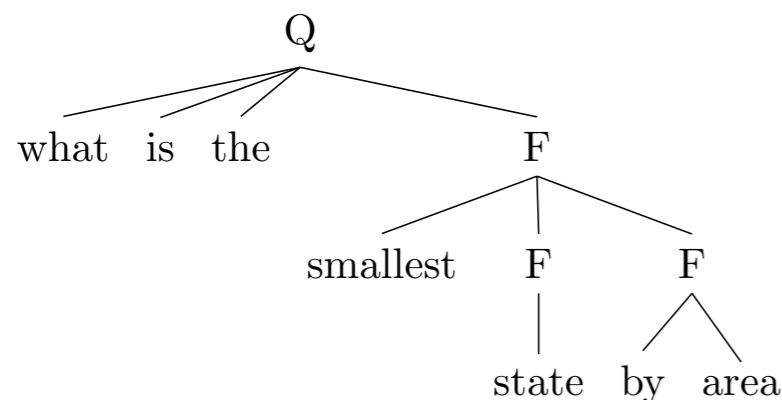
$F \rightarrow \text{by area}$

$Q \rightarrow \text{answer}(x_1, F(x_1))$

$F \rightarrow \lambda x_1 \text{smallest}(x_2, F(x_1), F(x_1, x_2))$

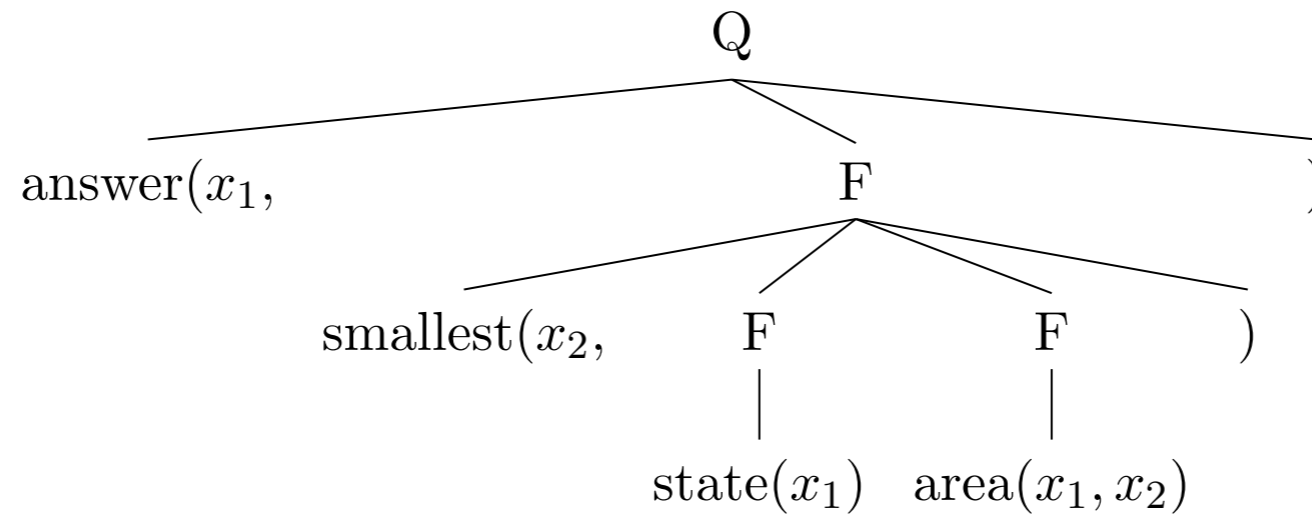
$F \rightarrow \lambda x_1 \text{state}(x_1)$

$F \rightarrow \lambda x_1 \lambda x_2 \text{area}(x_1, x_2)$



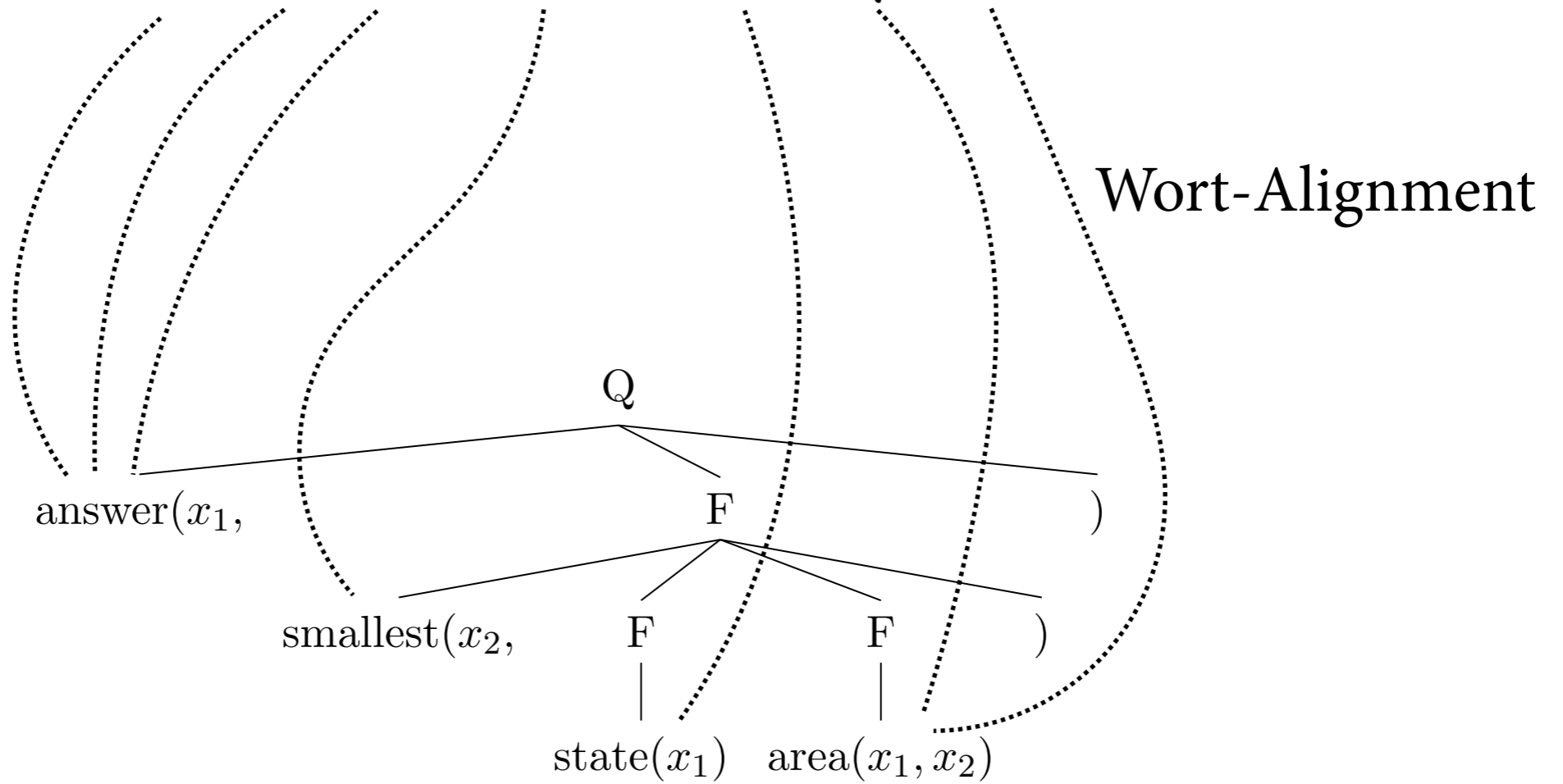
# SCFG lernen

what is the smallest state by area



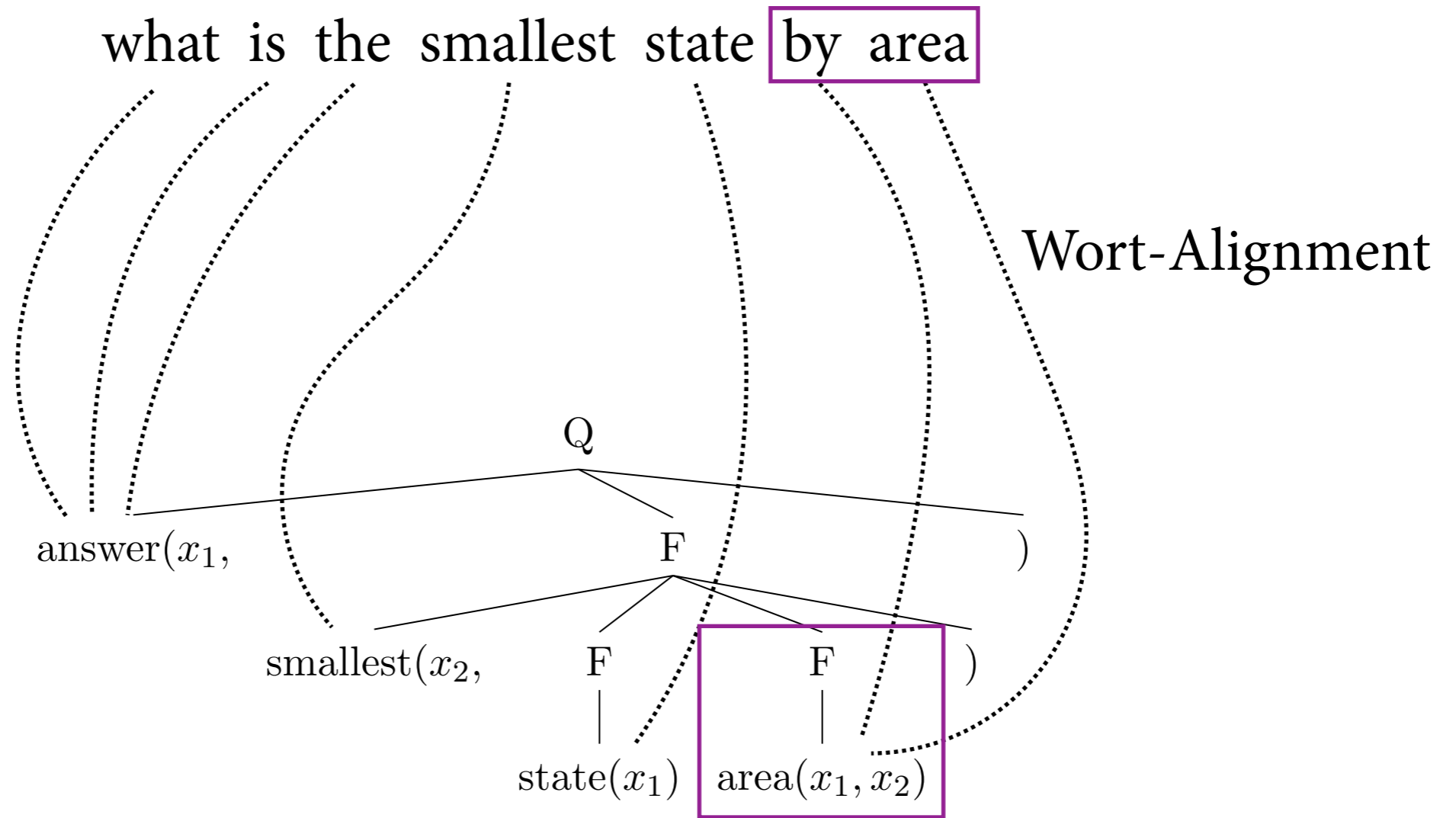
# SCFG lernen

what is the smallest state by area

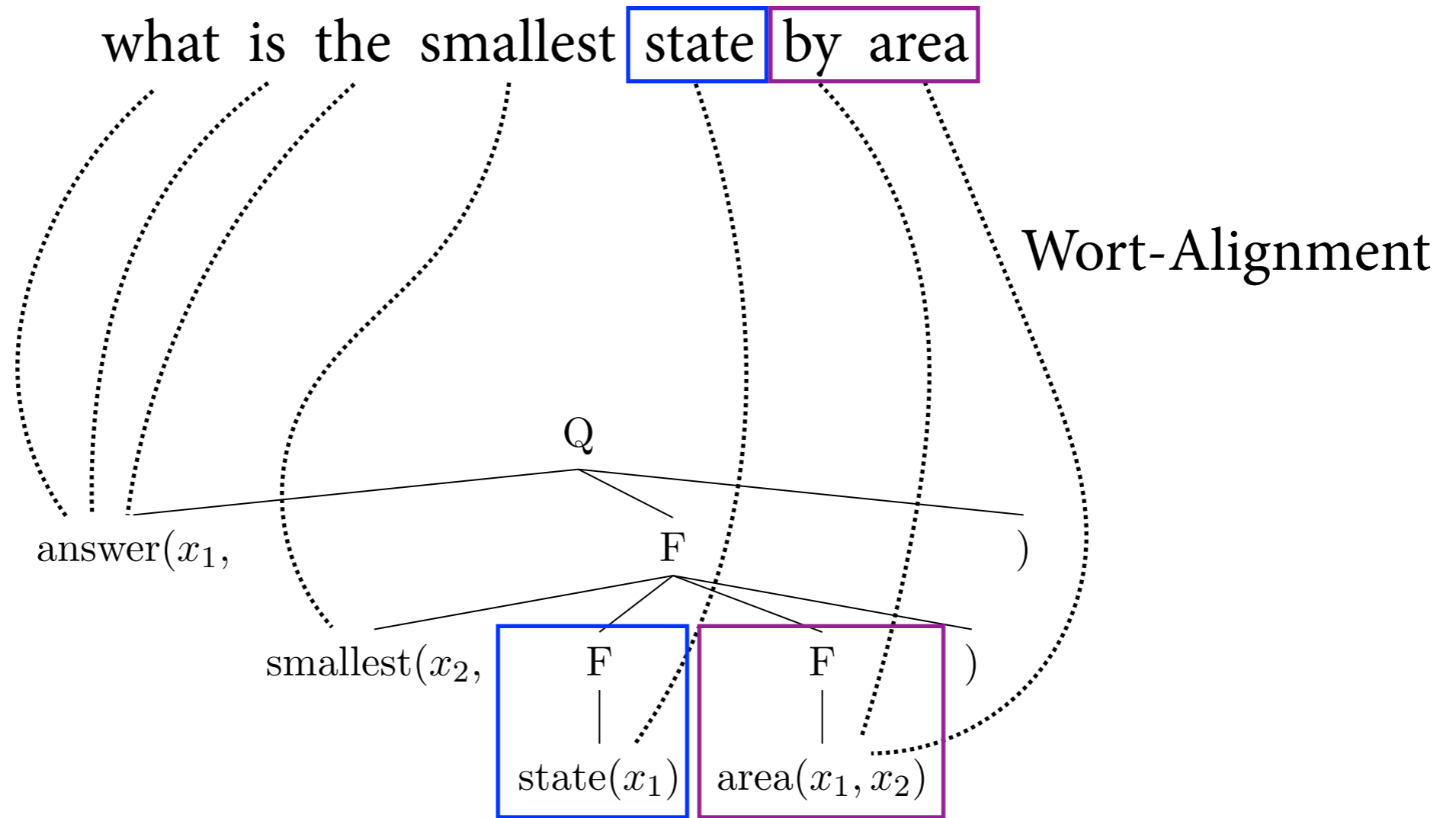




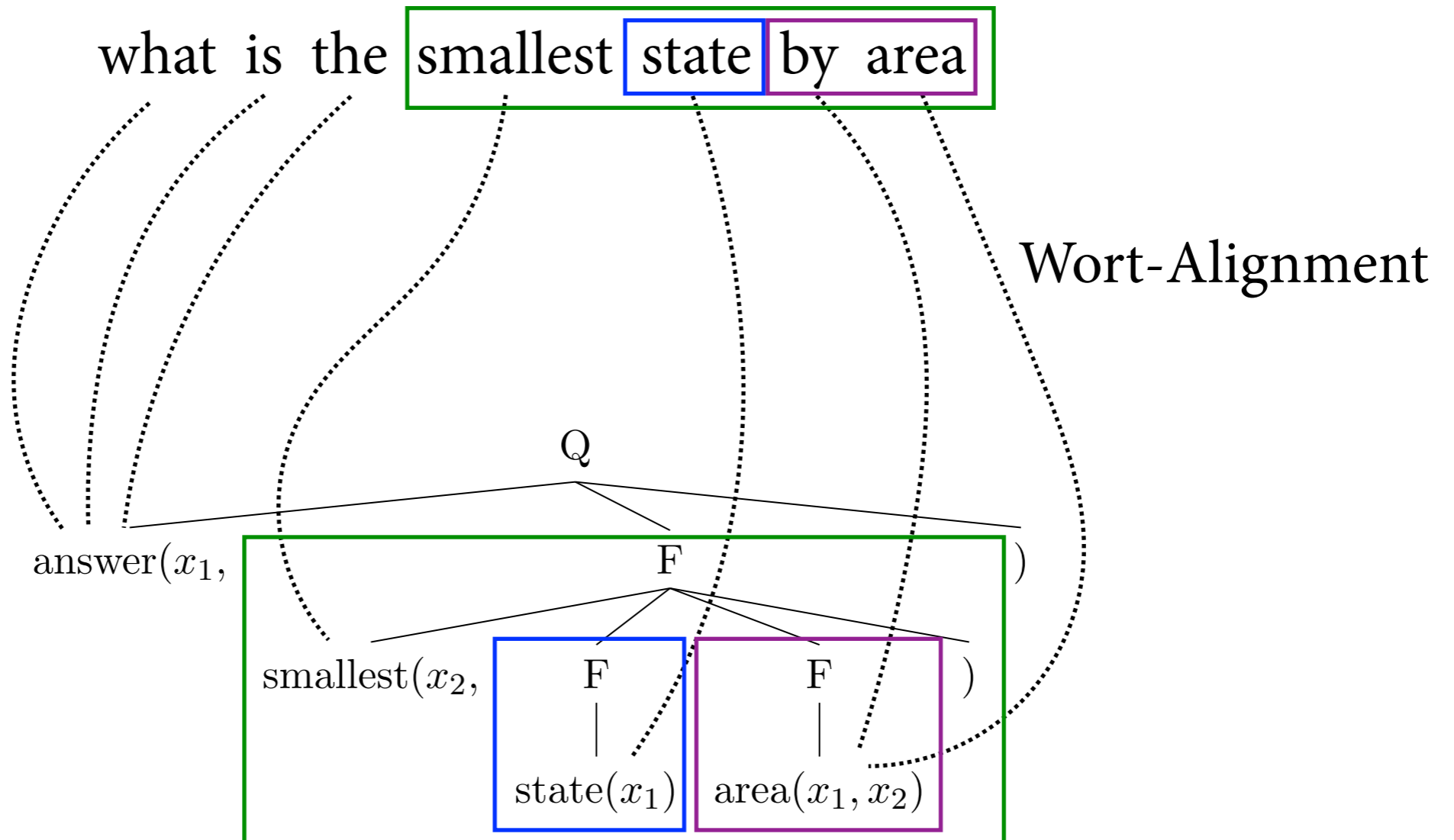
# SCFG lernen



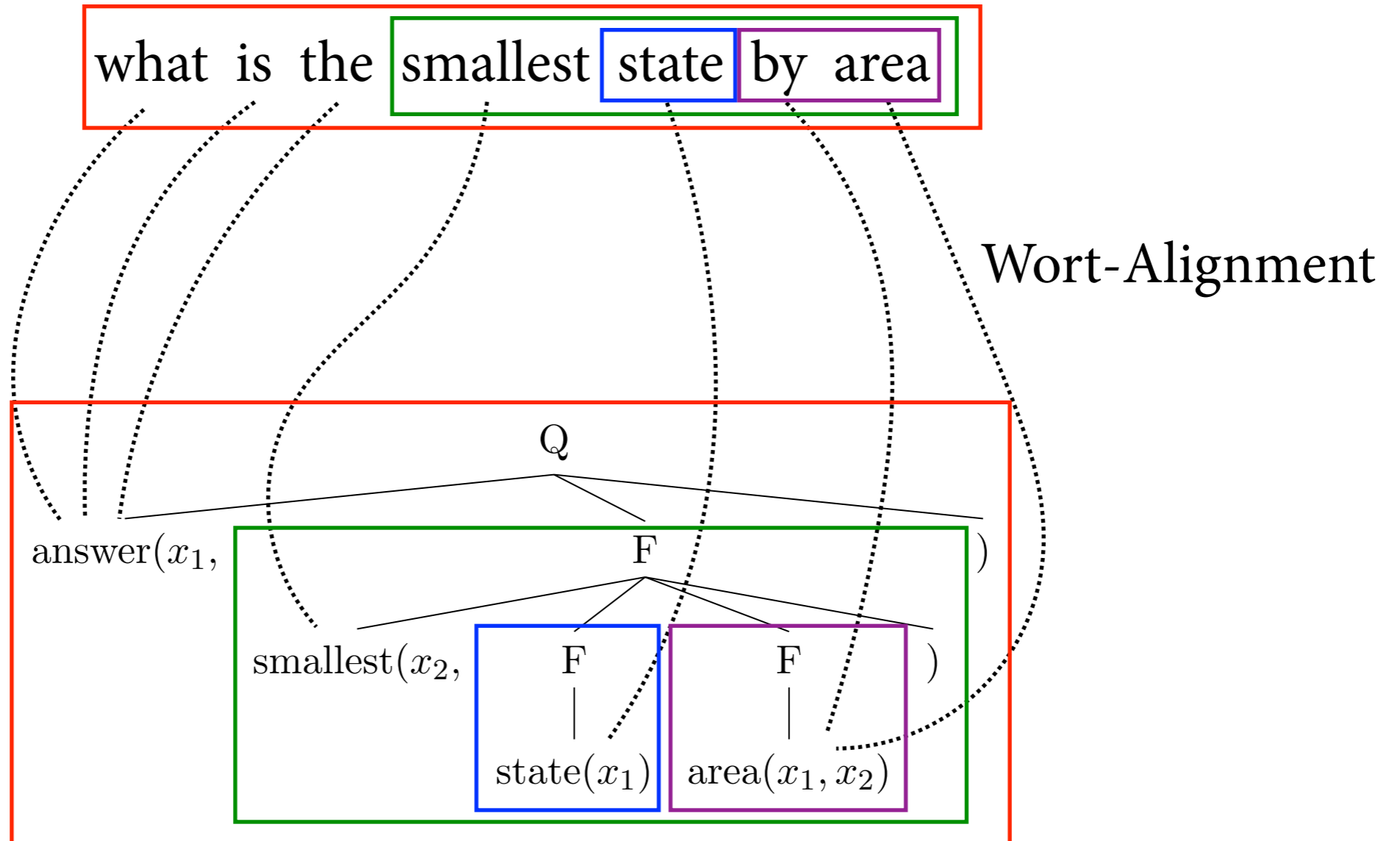
# SCFG lernen



# SCFG lernen



# SCFG lernen



# Zusammenfassung

- Inferenz mit natürlicher Sprache erfordert Berechnung von formalen Bedeutungsrepräsentationen.
- Kompositionelle Semantikkonstruktion.
  - ▶ Behandlung von Ambiguitäten mit Unterspezifikation
- Aktuelle Forschung zu Semantic Parsing:
  - ▶ direkte Abbildung von Satz zu Repräsentation lernen
  - ▶ zum Beispiel mit synchronen Grammatiken
  - ▶ momentan äußerst aktives Forschungsgebiet