Logic and lambda calculus

Simon Dobnik
CLASP & FloV
University of Gothenburg

CS VT-2023: March 27, 2023

Model theoretic semantics

- linguistic expressions $\rightsquigarrow$ logic expressions
- logic expressions $\rightsquigarrow$ objects in a model
- the model provides
  - a set of basic objects
  - an assignment of objects to the basic expressions of the (logic) language
- for each rule combining expressions in the natural language there is a rule in the logic language for which there is a corresponding rule operating on the model-theoretic objects

Model theoretic semantics

- linguistic expressions $\rightsquigarrow$ logic expressions
- logic expressions $\rightsquigarrow$ objects in a model
- the model provides
  - a set of basic objects
  - an assignment of objects to the basic expressions of the (logic) language
- for each rule combining expressions in the natural language there is a rule in the logic language for which there is a corresponding rule operating on the model-theoretic objects
- Formal languages:
  - Propositional logic
  - First order logic
  - Lambda calculus

Propositional logic

propositional-logic.ipynb

First-order logic

…predicate logic or predicate calculus

first-order-logic.ipynb

From linguistic structure to logic representations

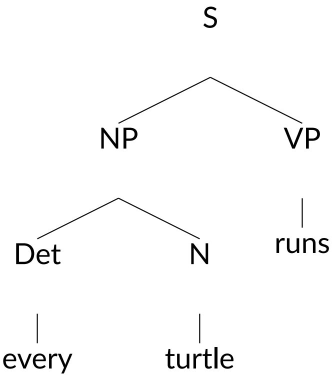
- a turtle runs
- $\exists x[\mathrm{turtle}(x) \wedge \mathrm{run}(x)]$
- a: $\exists x[?(x) \wedge ?(x)]$

CLASP centre for linguistic theory and studies in probability

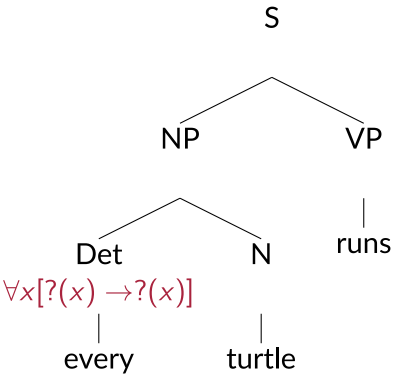
From linguistic structure to logic representations

- a turtle runs
- $\exists x[\mathrm{turtle}(x) \land \mathrm{run}(x)]$
- a: $\exists x[?(x) \land ?(x)]$
- every turtle runs
- $\forall x[\mathrm{turtle}(x) \rightarrow \mathrm{run}(x)]$
- every: $\forall x[?(x) \rightarrow ?(x)]$

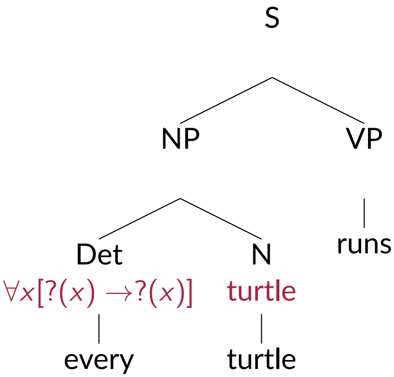
A mismatch in structure



A mismatch in structure



A mismatch in structure



S

NP

VP

Det

$\forall x[?(x) \rightarrow ?(x)]$

every

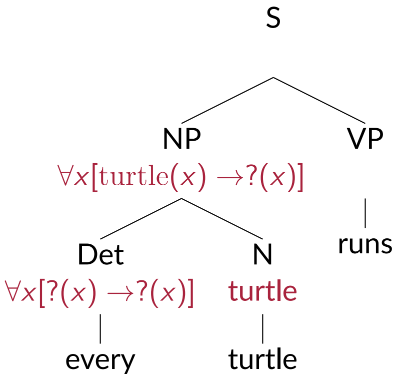
N

turtle

turtle

runs

A mismatch in structure



S

NP

$\forall x[\mathrm{turtle}(x) \rightarrow ?(x)]$

VP

runs

Det

$\forall x[?(x) \rightarrow ?(x)]$

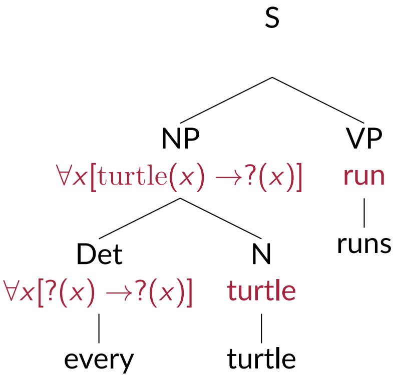
every

N

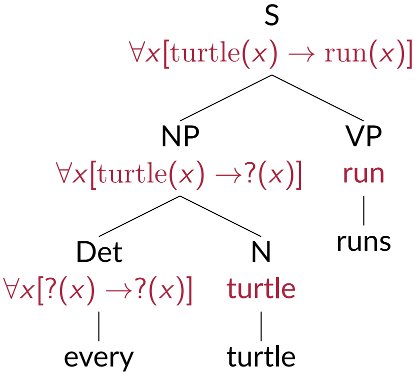
turtle

turtle

A mismatch in structure


S
NP
$\forall x[\mathrm{turtle}(x) \rightarrow ?(x)]$
VP
run
runs
Det
$\forall x[?(x) \rightarrow ?(x)]$
N
turtle
every
turtle

A mismatch in structure



S

$\forall x[\text{turtle}(x) \rightarrow \text{run}(x)]$

NP

$\forall x[\text{turtle}(x) \rightarrow ?(x)]$

VP

run

runs

Det

$\forall x[?(x) \rightarrow ?(x)]$

every

N

turtle

turtle

Solution: typed functions and lambda calculus

- $f(x) = 2x$

Solution: typed functions and lambda calculus

- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$

Solution: typed functions and lambda calculus

- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$
- Applying arguments and β-equivalence/conversion:
  $\lambda x[2 \times x](3)$

Solution: typed functions and lambda calculus

- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$
- Applying arguments and β-equivalence/conversion:
  $\lambda x[2 \times x](3)$
  2×3
  6

Solution: typed functions and lambda calculus

- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$
- Applying arguments and β-equivalence/conversion:
  $\lambda x[2 \times x](3)$
  2×3
  6
- Arguments can be assigned (basic) types:
  $x : r$ where r is the type of real numbers

Solution: typed functions and lambda calculus

- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$
- Applying arguments and β-equivalence/conversion:
  $\lambda x[2 \times x](3)$
  2×3
  6
- Arguments can be assigned (basic) types:
  $x : r$ where r is the type of real numbers
- Functions are also typed: $\langle r, r \rangle$
- Functions can have other functions as domains and targets:
  $\lambda y[\lambda x[2x + y]](3)$

Solution: typed functions and lambda calculus

- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$
- Applying arguments and β-equivalence/conversion:
  $\lambda x[2 \times x](3)$
  2×3
  6
- Arguments can be assigned (basic) types:
  $x : r$ where r is the type of real numbers
- Functions are also typed: $\langle r, r\rangle$
- Functions can have other functions as domains and targets:
  $\lambda y[\lambda x[2x + y]](3)$
  $\lambda x[2x + 3](4)$

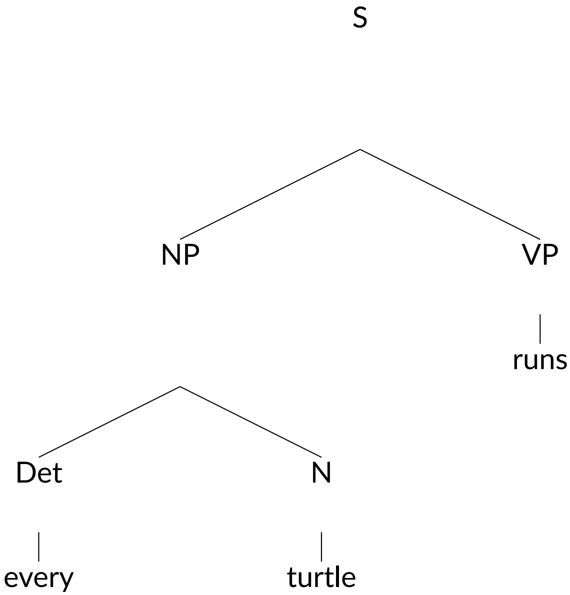
CLASP centre for linguistic theory and studies in probability

Solution: typed functions and lambda calculus

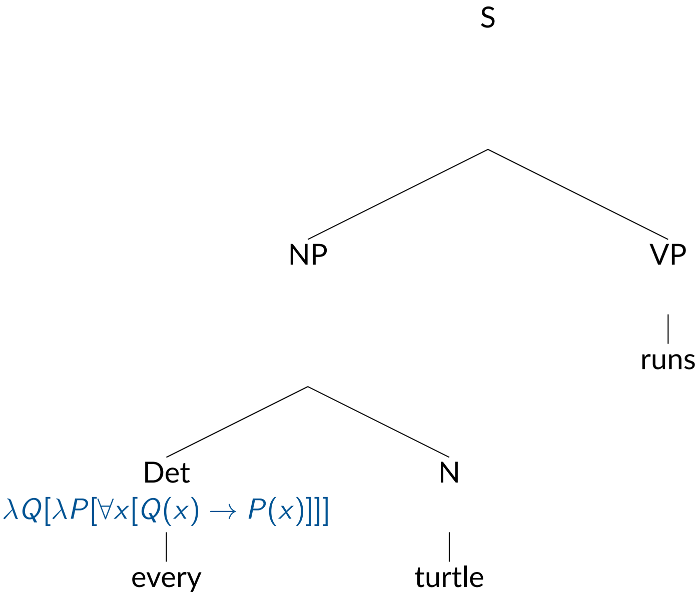
- $f(x) = 2x$
- $\lambda x[2x]$
  $\lambda x.(2x)$
  $\lambda x.2x$
- Applying arguments and β-equivalence/conversion:
  $\lambda x[2 \times x](3)$
  2×3
  6
- Arguments can be assigned (basic) types:
  $x : r$ where r is the type of real numbers
- Functions are also typed: $\langle r, r \rangle$
- Functions can have other functions as domains and targets:
  $\lambda y[\lambda x[2x + y]](3)$
  $\lambda x[2x + 3](4)$
  $2 \times 4 + 3$
  11

CLASP centre for linguistic theory and studies in probability

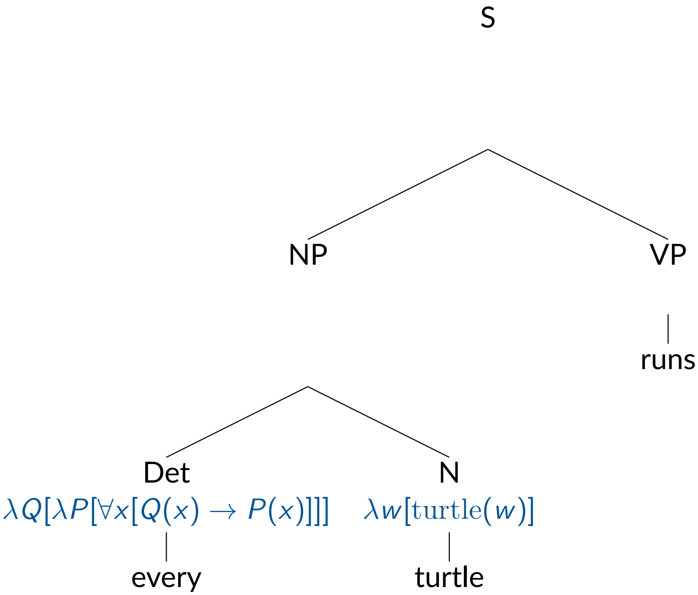
Solution for ? in the linguistic example



Solution for ? in the linguistic example



Solution for ? in the linguistic example



S

NP

VP

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]]$

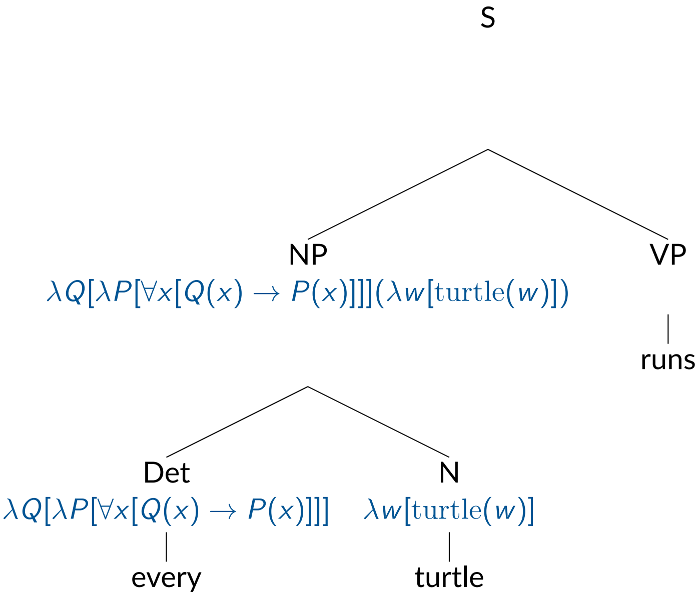
every

N

$\lambda w[\mathrm{turtle}(w)]$

turtle

Solution for ? in the linguistic example



S

NP

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]](\lambda w[\mathrm{turtle}(w)])$

VP

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]]$

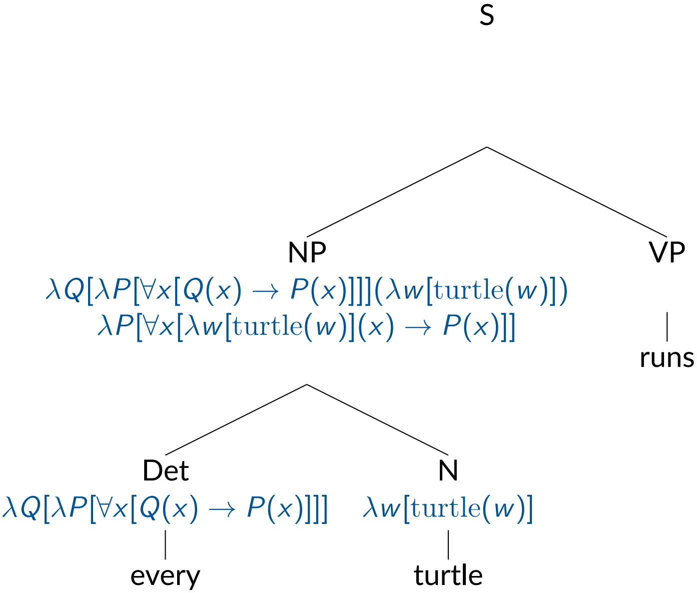
every

N

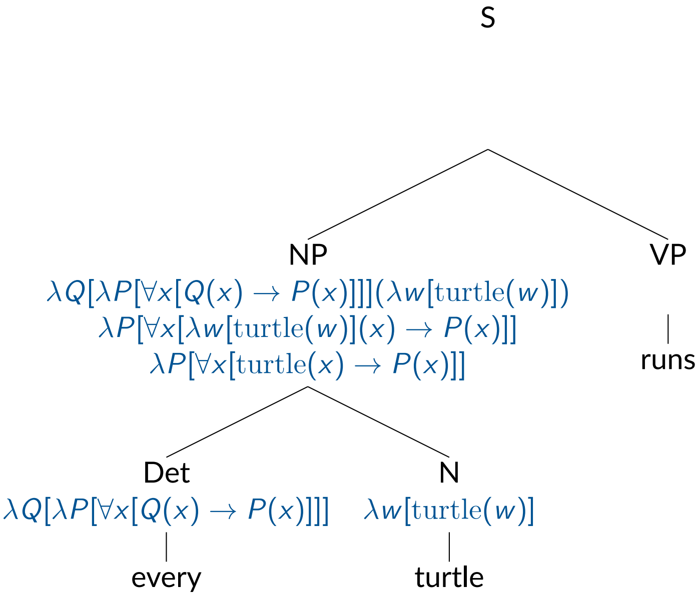
$\lambda w[\mathrm{turtle}(w)]$

turtle

Solution for ? in the linguistic example


S
NP
$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]](\lambda w[\mathrm{turtle}(w)])$
$\lambda P[\forall x[\lambda w[\mathrm{turtle}(w)](x) \rightarrow P(x)]]$
VP
runs
Det
$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]]$
every
N
$\lambda w[\mathrm{turtle}(w)]$
turtle

Solution for ? in the linguistic example



S

NP

$\lambda Q[\lambda P[\forall x[Q(x) \to P(x)]]](\lambda w[\mathrm{turtle}(w)])$

$\lambda P[\forall x[\lambda w[\mathrm{turtle}(w)](x) \to P(x)]]$

$\lambda P[\forall x[\mathrm{turtle}(x) \to P(x)]]$

VP

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \to P(x)]]]$

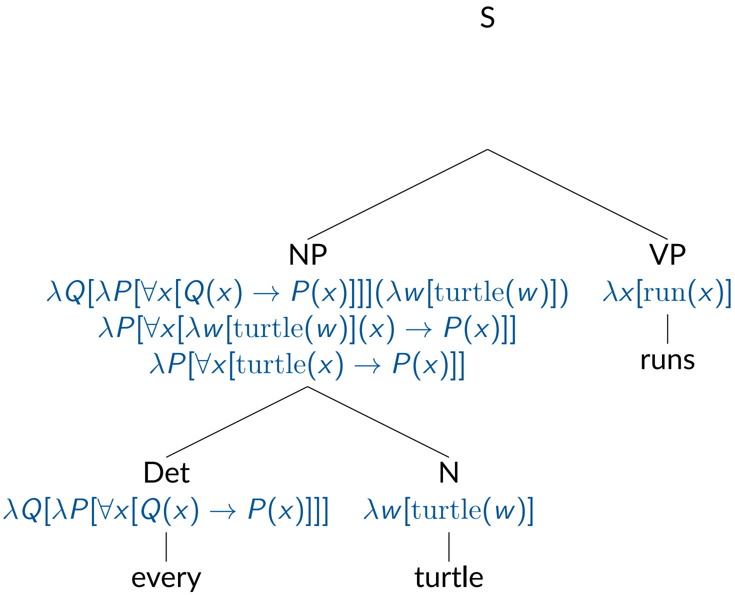
every

N

$\lambda w[\mathrm{turtle}(w)]$

turtle

Solution for ? in the linguistic example



S

NP

$\lambda Q[\lambda P[\forall x[Q(x) \to P(x)]]](\lambda w[\mathrm{turtle}(w)])$

$\lambda P[\forall x[\lambda w[\mathrm{turtle}(w)](x) \to P(x)]]$

$\lambda P[\forall x[\mathrm{turtle}(x) \to P(x)]]$

VP

$\lambda x[\mathrm{run}(x)]$

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \to P(x)]]]$

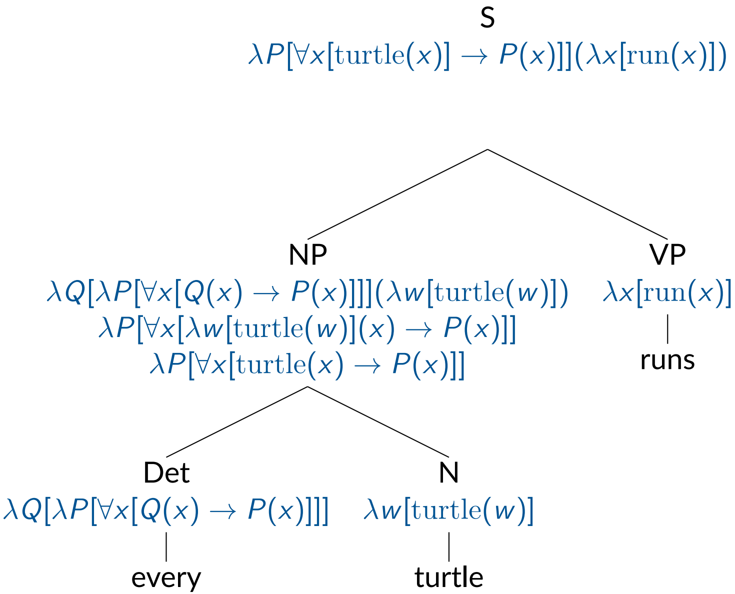
every

N

$\lambda w[\mathrm{turtle}(w)]$

turtle

Solution for ? in the linguistic example



S

$\lambda P[\forall x[\mathrm{turtle}(x)] \rightarrow P(x)]](\lambda x[\mathrm{run}(x)])$

NP

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]](\lambda w[\mathrm{turtle}(w)])$

$\lambda P[\forall x[\lambda w[\mathrm{turtle}(w)](x) \rightarrow P(x)]]$

$\lambda P[\forall x[\mathrm{turtle}(x) \rightarrow P(x)]]$

VP

$\lambda x[\mathrm{run}(x)]$

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]]$

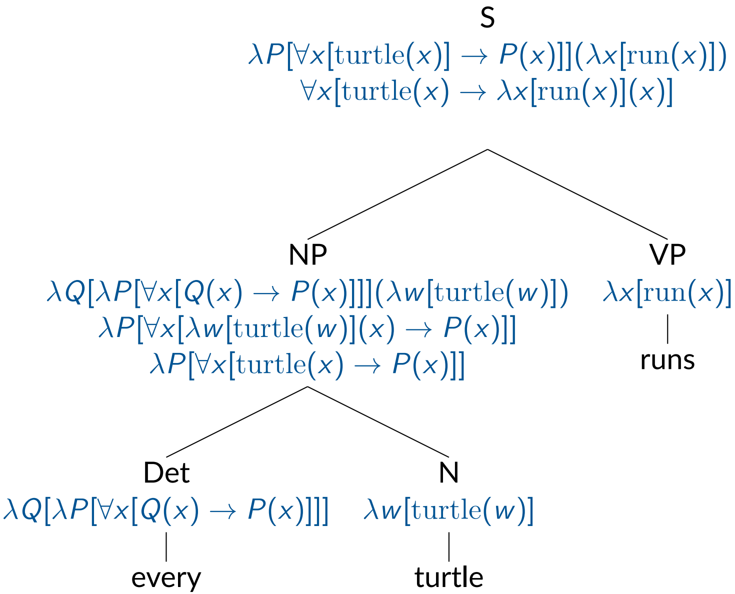
every

N

$\lambda w[\mathrm{turtle}(w)]$

turtle

Solution for ? in the linguistic example



S

$\lambda P[\forall x[\mathrm{turtle}(x)] \rightarrow P(x)]](\lambda x[\mathrm{run}(x)])$

$\forall x[\mathrm{turtle}(x) \rightarrow \lambda x[\mathrm{run}(x)](x)]$

NP

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]](\lambda w[\mathrm{turtle}(w)])$

$\lambda P[\forall x[\lambda w[\mathrm{turtle}(w)](x) \rightarrow P(x)]]$

$\lambda P[\forall x[\mathrm{turtle}(x) \rightarrow P(x)]]$

VP

$\lambda x[\mathrm{run}(x)]$

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]]$

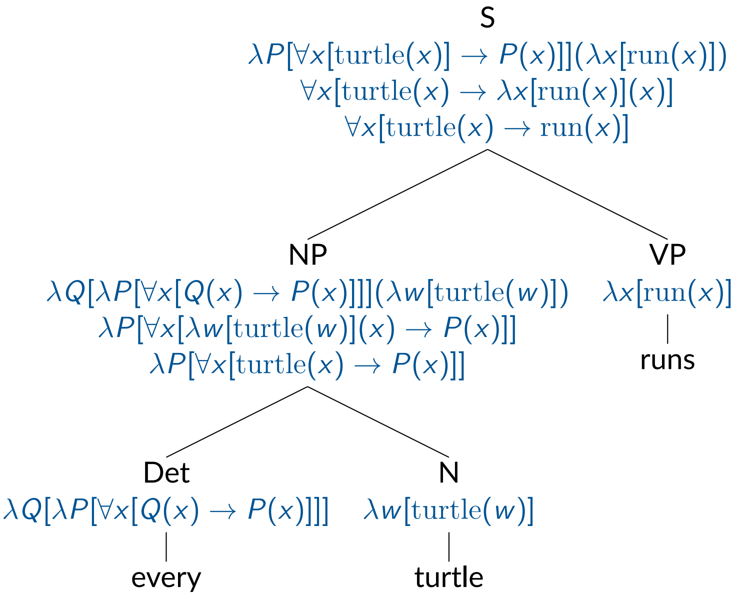
every

N

$\lambda w[\mathrm{turtle}(w)]$

turtle

Solution for ? in the linguistic example



S

$\lambda P[\forall x[\mathrm{turtle}(x)] \rightarrow P(x)]](\lambda x[\mathrm{run}(x)])$

$\forall x[\mathrm{turtle}(x) \rightarrow \lambda x[\mathrm{run}(x)](x)]$

$\forall x[\mathrm{turtle}(x) \rightarrow \mathrm{run}(x)]$

NP

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]](\lambda w[\mathrm{turtle}(w)])$

$\lambda P[\forall x[\lambda w[\mathrm{turtle}(w)](x) \rightarrow P(x)]]$

$\lambda P[\forall x[\mathrm{turtle}(x) \rightarrow P(x)]]$

VP

$\lambda x[\mathrm{run}(x)]$

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \rightarrow P(x)]]]$

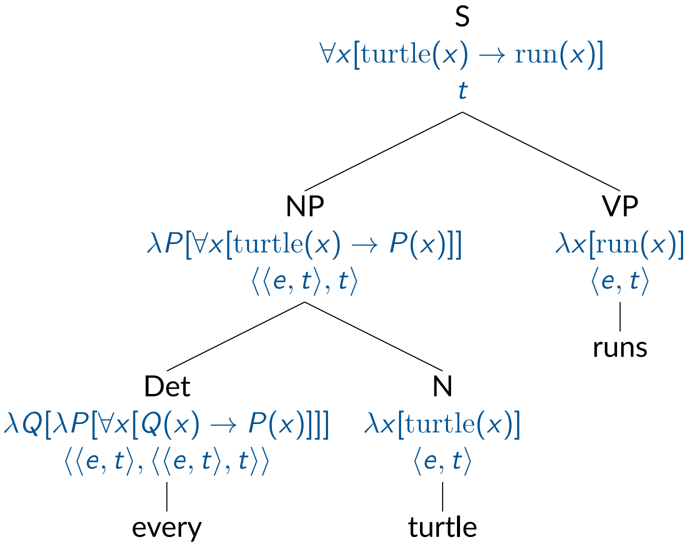
every

N

$\lambda w[\mathrm{turtle}(w)]$

turtle

Balance correct composition AND semantic interpretation



S

$\forall x[\mathrm{turtle}(x) \to \mathrm{run}(x)]$

t

NP

$\lambda P[\forall x[\mathrm{turtle}(x) \to P(x)]]$

$\langle\langle e, t\rangle, t\rangle$

VP

$\lambda x[\mathrm{run}(x)]$

$\langle e, t\rangle$

runs

Det

$\lambda Q[\lambda P[\forall x[Q(x) \to P(x)]]]$

$\langle\langle e, t\rangle, \langle\langle e, t\rangle, t\rangle\rangle$

every

N

$\lambda x[\mathrm{turtle}(x)]$

$\langle e, t\rangle$

turtle

Grammar with first-order logic + lambda calculus

lambda-calculus.ipynb

simple-sem.fcfg

Summary

- Logic-based computational semantics can use model theory to evaluate natural language in a database

Summary

- Logic-based computational semantics can use model theory to evaluate natural language in a database
- This is done by translating natural language into an extension of first order logic employing the λ-calculus

Summary

- Logic-based computational semantics can use model theory to evaluate natural language in a database
- This is done by translating natural language into an extension of first order logic employing the λ-calculus
- These expressions can be then evaluated in the database

Summary

- Logic-based computational semantics can use model theory to evaluate natural language in a database
- This is done by translating natural language into an extension of first order logic employing the λ-calculus
- These expressions can be then evaluated in the database
- Translations to logic and lambda calculus can be done for a significant part of natural language

Summary

- Logic-based computational semantics can use model theory to evaluate natural language in a database
- This is done by translating natural language into an extension of first order logic employing the λ-calculus
- These expressions can be then evaluated in the database
- Translations to logic and lambda calculus can be done for a significant part of natural language
- There are, however, important constructions of natural language which go beyond first order logic

Further reading

- Bird et al. (2009), Chapter 10: Analyzing the meaning of sentences
- Manning (2005)
- Eisenstein (2019), Chapter 13: Logical Semantics
- Jurafsky and Martin (2000): Chapter 17: Representing meaning and Chapter 18: Computational semantics

Advanced

- Blackburn and Bos (2005): Chapter 1: First-order logic and Chapter 2: Lambda calculus
- Eijck and Unger (2010)

Acknowledgement

Some slides based on the slides by Robin Cooper.

References I

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. O'Reilly.

Patrick Blackburn and Johan Bos. 2005. *Representation and inference for natural language. A first course in computational semantics*. CSLI Publications.

J. van Eijck and Christina Unger. 2010. *Computational semantics with functional programming*. Cambridge University Press, Cambridge.

Jacob Eisenstein. 2019. *Introduction to natural language processing*. Adaptive computation and machine learning. The MIT Press: Cambridge, MA and London, England.

Dan Jurafsky and James H. Martin. 2000. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall, Upper Saddle River, N.J.

Christopher D. Manning. 2005. An introduction to formal computational semantics. Lecture notes for CS224N/Ling 280.