Syntax-Semantics (Syn-Sem) Interface

Lambda (λ) Calculus

Lambda calculus is a representation of logical relationships as functions, f(x), and function compositions, f(g(x)).

f(x) == λx. 2 + x, f(1) = 3

g(x) == λx. x*x, g(2) = 4

f(g(x)) == λx. 2 + x*x f(g(3)) = 11

λ Calculus and Language

Each POS can be represented as a different lambda expression.

For example,

DT {λP[λQ [∃ x P(x) ∧ Q(x)]]} → a | an

N {λy.pokemon(y)}→ pokemon

IV {λz.appeared(z)} → appeared

λ Calculus and Language

Sentences and Y/N questions can be thought of as expressions.

Most wh-questions can be thought of as lambda expressions.

λ Calculus and Language

Compare:

A pokemon appeared.

$\exists x[\ pokemon(x) \wedge appeared(x)]$

And

What appeared?

$\lambda.P[\ \exists x[\ P(x) \wedge appeared(x)]]$

λ Calculus and Language

How do we get from

DT {λP[λQ [∃x P(x) ∧ Q(x)]]} → a | an

N {λy.pokemon(y)}→ pokemon

IV {λz.appeared(z)}→ appeared

to

∃x[pokemon(x) ∧ appeared(x)] ?

Function application

$$\lambda \mathbf{P}[\lambda Q\ [\exists x\ \mathbf{P}(x) \wedge Q(x)]]\ \mathbf{\{\lambda y.pokemon(y)\}}$$

$$\lambda Q\ [\exists x\ \mathbf{\lambda y.pokemon(y)}(x) \wedge Q(x)]$$

Reduction

$$\lambda Q\,[\,\exists\, x\ \lambda y.\text{pokemon}(y)\mathbf{(x)} \wedge Q(x)]$$

$$\lambda Q\,[\,\exists\, x\ \text{pokemon}\mathbf{(x)} \wedge Q(x)]$$

Finishing up the sentence

$\lambda Q\ [\exists x\ \text{pokemon}(x) \wedge \mathbf{Q}(x)]\ \mathbf{\{\lambda z.\text{appeared}(z)\}}$

$\exists x\ \text{pokemon}(x) \wedge \lambda z.\text{appeared}(z)\mathbf{(x)}$

$\exists x\ \text{pokemon}(x) \wedge \text{appeared}(x)$

Your turn

A wild pokemon appeared.

DT {λP[λQ [∃x P(x) ∧ Q(x)]]} → a | an

Adj {λR[λw [Z(w) ∧ wild(w)]}

N {λy.pokemon(y)}→ pokemon

IV {λz.appeared(z)}→ appeared

λ Calculus in PROLOG

```
lexeme\2 % lexeme(semantics, token)
lexeme(n(X^pokemon(X)),pokemon).
lex(dt((X^P)^(X^Q)^exists(X,and(P,Q))),a).

% Augment rules with semantics
rule(np(B),[dt(A^B),n(A)]).
```