Review

• PC vs. IPC?
• How to find $[[P(j)]]^{M,w,i,g,c}$?
• How to find $[[\forall xP(x)]]^{M,w,i,g,c}$?
• How to find $[[\exists xP(x)]]^{M,w,i,g,c}$?
• How to find $[[\Box P(x)]]^{M,w,i,g,c}$?
• How to find $[[\Diamond P(x)]]^{M,w,i,g,c}$?
• How to find $[[FP(x)]]^{M,w,i,g,c}$?
• How to find $[[PP(x)]]^{M,w,i,g,c}$?

• Predicate expressions: formalizing semantic relations and their representation
  • Predicate names: lexical item; entities: variables (e.g. x, y, z)
    Dave eats a clam.: dave(x) & clam(y) & eats(x, y)
What’s the difference?

- fred
- \( \lambda x. \text{fred}(x) \)
- \( \lambda P.P(x)[\text{fred}] \)
LAMBDA OPERATIONS
Motivation

• Sentence: subject + predicate
  • Something’s shared between them
  • John is hungry: hungry(John)
    John sneezed: \( P \text{sneeze}(John) \)
    John might sneeze: \( \Box F \text{sneeze}(John) \)

• Complex NP: head noun + relative clause
  • Something’s shared between them
  • the man who sneezed…: \( \text{man}(x) \& P \text{sneeze}(x) \)
    the man who Fred saw…: \( \text{man}(x) \& P \text{see}(Fred,x) \)
    the man who was given a ticket…: \( \text{man}(x) \& \text{ticket}(y) \& \text{give}(z, y, x) \)

• Lambdas allow us to specify what’s shared where, and to keep available slots “open” until we’re ready to fill them.
What is a lambda?

• \( \lambda \)

• Basic building block in several linguistic theories
  • We’ll use it in two different theories this term

• Operator that associates, combines semantic items compositionally
  • Predicates, entities
  • Variables

• If \( \psi \) is a well-formed formula and \( x \) a variable, \( \lambda x[\psi] \) is a Pred\(_1\).
  • \( \lambda x[\neg \text{married}(x) \& \text{male}(x) \& \text{adult}(x)] \)
Lexical items and predication

• ...sneezed ➔ \( \lambda x. [\text{sneeze}(x)] \)
• ...saw... ➔ \( \lambda y. \lambda x. [\text{see}(x, y)] \)
• ... laughed and is not a woman ➔
  \( \lambda x. [\text{laugh}(x) \& \neg \text{woman}(x)] \)
• ... respects himself ➔
  \( \lambda x. \text{respect}(x, x) \)
• ...respects and is respected by... ➔
  \( \lambda y. \lambda x. [\text{respect}(x, y) \& \text{respect}(y, x)] \)
Why and what?

- Mechanism for spreading/collapsing information
- \( \lambda \)-conversion operation
  - Syntax: \( \lambda x \) prefixed to any wff, where \( x \) is a variable
  - Semantics: \( \lambda x[\varphi](t) \leftrightarrow \varphi[t/x] \)
    - i.e. substitute \( t \) for all occurrences of \( x \)
- \( \lambda \)-reduction: left-to-right
- \( \lambda \)-abstraction: right-to-left

\[
[[\lambda x[\psi]]]^{M,w,i,g} = \{ u \in U : [[\psi]]^{M,w,i,g}[u/x] = 1 \}
\]
- Builds the set that constitutes the extension of the predicate
How lambdas operate

• Lambdas fill open predicates’ variables with content
• John sneezed.
  john’, λx.[P[sneeze’(x)]
  λx.[P[sneeze’(x)] (john’)

✗ P[sneeze’(john’)]
The basic op: \( \lambda \)-conversion

• In an expression \((\lambda x. W)(z)\), replace all occurrences of the variable \(x\) in the expression \(W\) with \(z\).
  
  • \((\lambda x. \text{hungry}(x))(\text{John}) \rightarrow \text{hungry(John)}\)

  • \((\lambda x. [\neg \text{married}(x) \& \text{male}(x) \& \text{adult}(x)])(\text{John}) \rightarrow \neg \text{married(John)} \& \text{male(John)} \& \text{adult(John)}\)
Hints

• Work one step at a time
• Pay particular attention to scope in $\lambda$-expressions
• Special considerations (but don’t worry about them for this class)
  • Intensionality is messy unless rigid designators are being considered

• Starting at p. 400 or so, the text uses tick/prime marks to differentiate semantic objects (predicate names, constants, etc.) from lexical objects
  • hungry'(fred')
Reduce this

\[(4) \, \exists y[\exists z[\exists x[B(x) \rightarrow \exists w[R(x, w)](j)] \land \exists x[B(x) \lor Q(x)](z)](y)]\]
Let $M_8$ be such that the following hold:

(a) $U_8 = \{\text{Pavarotti, Bond, Loren}\}$
(b) $V_8(P) = \{\text{Pavarotti, Bond}\}$
(c) $V_8(m) = \text{Pavarotti}$
(d) $V_8(j) = \text{Bond}$

And let $g$ be an arbitrary value assignment.

(e) $\left[\lambda x[P(x) \land \neg[x = m]]\right]^{M_8.g} = \{u : \left[\left[ P(x) \land \neg[x = m]\right]^{M_8.g[u/x]} = 1\right\}$ By (11)

(f) $\left[P(x) \land \neg[x = m]\right]^{M_8.g[\text{Pavarotti}/x]} = 0$ By (a) to (d) and the semantics of PC

(g) $\left[P(x) \land \neg[x = m]\right]^{M_8.g[\text{Bond}/x]} = 1$ By (a) to (d) and the semantics of PC

(h) $\left[P(x) \land \neg[x = m]\right]^{M_8.g[\text{Loren}/x]} = 0$ By (a) to (d) and the semantics of PC

(i) $\left[\lambda x[P(x) \land \neg[x = m]]\right]^{M_8.g} = \{\text{Bond}\}$ By (e) to (h)
The new semantics

- Two components
  - Compositional translation into logical calculus (representation)
  - Truth-conditional and model-theoretic evaluation
- $\lambda$ operator brings them together
- Treatment is systematic, modular, compositional, principled
Syntax of $F_3$

a. i. $TP \rightarrow NP \bar{T}$
   ii. $\bar{T} \rightarrow T \ VP$

b. i. $TP \rightarrow TP \ conj \ TP$
   ii. $TP \rightarrow \ neg \ TP$

c. $VP \rightarrow V_t \ NP$

d. $VP \rightarrow V_i$

e. $VP \rightarrow V_{dt} \ NP \ PP[to]$

f. $T \rightarrow \ PAST, \ PRES, \ FUT$

g. $NP \rightarrow \ Det \ N_c$

h. $PP[to] \rightarrow \ to \ NP$

i. $Det \rightarrow \ the, \ a, \ every$

j. $N_p \rightarrow \ Pavarotti, \ Loren, \ Bond, \ldots, \ he_n, \ldots$

k. $N_c \rightarrow \ book, \ fish, \ man, \ woman, \ldots$

l. $V_i \rightarrow \ be \ boring, \ be \ hungry, \ walk, \ talk, \ldots$

m. $V_t \rightarrow \ like, \ hate, \ kiss, \ldots$

n. $V_{dt} \rightarrow \ give, \ show, \ldots$

o. $\ conj \rightarrow \ and, \ or$

p. $\ neg \rightarrow \ not$

q. $NP \rightarrow N_p$

r. $CP \rightarrow C \ TP$

s. $VP \rightarrow V_s \ CP$

t. $V_s \rightarrow \ believe, \ know, \ regret, \ldots$

u. $C \rightarrow \ that$

a. i. $[A \ B]' = B'$
   ii. $[A \ to \ B] = B'$

example: $[to \ Lee]' = \text{Lee'}$

b. $[TP \ NP \ T]' = T'(NP')$

e. $[V \ NP \ PP]' = \lambda x[V'(x, NP')]

example: $[VP \ like \ Kim]' = \lambda x[\text{like'}(x, Kim')]

d. $[VP \ V \ NP]' = \lambda x[V'(x, NP')]$

e. $[V \ NP \ PP]' = \lambda x[V'(x, NP', PP')]

example: $[VP \ introduce \ him, \ to \ Lee ]'$

f. $[CP \ C \ TP]' = C' \ TP'$

example: $[that \ Lee \ smokes]' = \wedge \text{smoke'}(\text{Lee}')$

g. $[X \ TP]' = X' \ TP'$

examples: $[PAST \ Lee \ smoke]' = P \text{smoke'}(\text{Lee}')$

h. $[NP_i \ TP]$ structures

i. if $NP_i = [every \beta]$, then $[NP_i \ TP]' = \forall x_i[\beta'(x_i) \rightarrow TP']$

example: $[every \ dog, \ t_i \ barks]'$

j. $= \forall x_i[\text{dog'}(x_i) \rightarrow \text{bark'}(x_i)]$

ii. if $NP_i = [a \ beta]$, then $[NP_i \ TP]' = \exists x_i[\beta'(x_i) \wedge TP']$

example: $[a \ dog_i[t_i \ barks]' = \exists x_i[\text{dog'}(x_i) \wedge \text{bark'}(x_i)]$

iii. if $NP_i = [the \ beta]$, then

$[NP_i \ TP]' = \exists x_i[\beta'(x_i) \wedge \forall y[\beta'(y) \rightarrow x_i = y] \wedge TP']$

example: $[the \ dog_i[t_i \ barks]'$

$= \exists x_i[\text{dog'}(x_i) \wedge \forall y[\text{dog'}(y) \rightarrow x_i = y] \wedge \text{bark'}(x_i)]$
(a) S-structure

\[ TP \left[ NP \left[ \text{the fish} \right] \right] \left[ \neg \left[ VP \left[ \text{not introduce Pavarotti to Loren} \right] \right] \right] \]

(b) LF

\[
\begin{array}{c}
\text{NP}_2 & \text{TP} [2] \\
\text{Det} & \text{N [3]} & \text{NEG [4]} & \text{TP [5]} \\
\text{the} & \text{fish} & \text{not} & \text{T [6]} & \text{TP [7]} \\
\text{PAST} & \text{NP [8]} & \text{T [9]} \\
\theta_2 & \text{VP [10]} \\
\text{V [11]} & \text{NP [12]} & \text{PP [13]} \\
\text{introduce} & \text{Pavarotti} & \text{P} & \text{NP [14]} \\
\text{to} & \text{Loren} \\
\end{array}
\]

(c) Node-by-node translation (from bottom up)

[11] \Rightarrow \text{introduce}' \text{ By (25a)}

[12] \Rightarrow \text{Pavarotti}' \text{ By (25a)}

[14] \Rightarrow \text{Loren}' \text{ By (25a)}

[13] \Rightarrow \text{Loren}' \text{ By (25a)}

[10] \Rightarrow \lambda x \left[ \text{introduce}' \left( x, \text{Pavarotti}', \text{Loren}' \right) \right] \text{ By (25c)}

[9] \Rightarrow \lambda x \left[ \text{introduce}' \left( x, \text{Pavarotti}', \text{Loren}' \right) \right] \text{ By (25a)}

[8] \Rightarrow x_2 \text{ By (25a)}

[7] \Rightarrow \lambda x \left[ \text{introduce}' \left( x, \text{Pavarotti}', \text{Loren}' \right) \right] (x_2) \text{ By (25b)}

\Rightarrow \lambda x \left[ \text{introduce}' \left( x_2, \text{Pavarotti}', \text{Loren}' \right) \right] \text{ By $\lambda$-conv.}

[6] \Rightarrow P \text{ By (25a)}

[5] \Rightarrow P \text{ introduce}' \left( x_2, \text{Pavarotti}', \text{Loren}' \right) \text{ By (25g)}

[4] \Rightarrow \neg P \text{ By (25a)}

[2] \Rightarrow \neg P \text{ introduce}' \left( x_2, \text{Pavarotti}', \text{Loren}' \right) \text{ By (25g)}

[3] \Rightarrow \text{fish}' \text{ By (25a)}

[1] \Rightarrow \exists x_2 \left[ \text{fish}' \left( x_2 \right) \land \forall y \left[ \text{fish}' \left( y \right) \rightarrow y = x_2 \right] \right] \text{ By (25h.i)}

\Rightarrow \neg P \text{ introduce}' \left( x_2, \text{Pavarotti}', \text{Loren}' \right) \text{ By (25h.ii)}
Translating syntax to semantics

Analysis of (28b), “Every man is hungry or is boring”

(a) S-structure

\[ S \left[ \text{NP every man, VP [VP is hungry \text{ or } VP is boring]} \right] \]

(b) LF

\[
\begin{array}{c}
S \left[ 1 \right] \\
\downarrow \\
NP_1 \quad S \left[ 2 \right] \\
\downarrow \\
\text{Det} \quad \text{N} \quad \text{NP} \left[ 3 \right] \quad \text{VP} \left[ 4 \right] \\
\downarrow \quad \downarrow \quad \downarrow \\
\text{every} \quad \text{man} \quad e_1 \quad \text{VP} \left[ 5 \right] \quad \text{conj} \quad \text{VP} \left[ 6 \right] \\
\downarrow \quad \downarrow \\
is \text{hungry} \quad \text{or} \quad is \text{boring}
\end{array}
\]

(c) Compositional interpretation (each numbered node is associated with its translation)

i. [5] ⇒ hungry’
ii. [6] ⇒ boring’
iii. [3] ⇒ x₁
iv. [4] ⇒ [hungry’ v boring’] By (32)
v. [2] ⇒ [hungry’ v boring’](x₁) By (25b)
vi. [1] ⇒ ∀x₁[man’(x₁) → [hungry’ v boring’](x₁)] By (25h,i)
    = ∀x₁[man’(x₁) → \lambda y[hungry’(y) v boring’(y)](x₁)]
    By (31b)
    = ∀x₁[man’(x₁) → [hungry’(x₁) v boring’(x₁)]] By \lambda \text{-conv.}
Refining previous notions

• LF: syntactically-interpreted semantic form
• lf: logical interpretation of propositional content
• Model-theoretic
  • Syntactic framework: as before
  • Semantic framework: as before, except that VP’s become λ-expressions
Linguistic applications of $\lambda$

- Relative clauses
  - Assume DS/SS dichotomy, transformational mapping, constraints
  - Fronted relative pronoun is (in some sense) the $\lambda$ term, derivation proceeds as usual

- Disjunction and conjunction
  - Combining subject and compound predicate
Linguistic applications of $\lambda$

- VP anaphora
  - Often ambiguous: sloppy/strict identity
  - Deletion vs. generated empty category
  - Generality: require semantic identity
  - Empty “placeholder predicate” substituted with $\lambda$-expression
  - Very interesting scopal interactions, fine-grained predictions for acceptability
VISH and coordinated VP’s

- Assume subject originates in spec-VP in DS
- It moves to spec-TP (overtly, i.e. before SS)

- Every student is tired and didn’t enjoy the show.

- Lambda opens, “saves” a slot in each conjunct
Relative clauses

- Head, complement
  - Gap in complement refers to head
  - Can be subject, object, oblique
- Assume wh- element in appropriate slot in DS
Relative clauses

- Overt wh- movement to spec-CP
  - Leaves a trace behind
  - Refers to head
  - (or, as here, Chomsky adjoin)

- Lambda can bind head to trace

\[
[\text{whom}_2 [\text{Mary likes } e_2]] \\
\Rightarrow \lambda x_2 \text{like}'(\text{Mary}', x_2)
\]
Syntax and semantics of “Pavarotti likes a fish that Loren hates”

(a) D-structure
\[ S [\text{Pavarotti} [\text{VP likes} [\text{NP a} [\text{fish} [\text{CP that} [S \text{Loren hates which}_1]]]]]] \]

(b) S-structure
\[ S [\text{Pavarotti} [\text{VP likes} [\text{NP a} [\text{fish} [\text{CP which}_1 [\text{CP that} [S \text{Loren hates} e_1]]]]]]]] \]
From (a) via wh-movement

(c) LF
\[ [\text{a} [\text{fish} [\text{CP which}_1 [\text{CP that} [S \text{Loren hates} e_1]]]]]]_2 \]
\[ S [\text{Pavarotti} [\text{VP likes} [\text{NP} e_2]]] \]
From (b) via QR

\[ S [1] \]
\[ NP_2 \]
\[ S [2] \]
\[ \text{Det} \]
\[ \text{N} [3] \]
\[ \text{NP} \]
\[ \text{VP} \]
\[ \text{DP} \]
\[ \text{Det} \]
\[ \text{N} [4] \]
\[ \text{CP} [4] \]
\[ \text{Pavarotti} \]
\[ \text{V} \]
\[ \text{NP} \]
\[ \text{C} \]
\[ \text{S} [5] \]
\[ \text{NP} \]
\[ \text{VP} \]
\[ \text{V} \]
\[ \text{NP} \]

\)

Compositional interpretation

i. \[ [5] \Rightarrow \text{hate}'(\text{Loren}', x_1) \] By (25a, b, d)

ii. \[ [4] \Rightarrow \lambda x_1[\text{hate}'(\text{Loren}', x_1)] \] By (55a)

iii. \[ [3] \Rightarrow \text{fish}' \land \lambda x_1[\text{hate}'(\text{Loren}', x_1)] \] By (55b)
\[ = \lambda y[\text{fish}'(y) \land \lambda x_1[\text{hate}'(\text{Loren}', x_1)](y)] \] By (31a)
\[ = \lambda y[\text{fish}'(y) \land \text{hate}'(\text{Loren}', y)] \] By \( \lambda \)-conversion

iv. \[ [2] \Rightarrow \text{like}'(\text{Pavarotti}', x_2) \] By (25a, b, d)

v. \[ [1] \Rightarrow \exists x_2[\lambda y[\text{fish}'(y) \land \text{hate}'(\text{Loren}', y)](x_2) \land \text{like}'(\text{Pavarotti}', x_2)] \] By (25h.ii)
\[ = \exists x_2[\text{fish}'(x_2) \land \text{hate}'(\text{Loren}', x_2) \land \text{like}'(\text{Pavarotti}', x_2)] \] By \( \lambda \)-conversion