Semantic Theory

Summer 2005

Dynamic Semantics and Compositionality

M. Pinkal / A. Koller

Is discourse semantics compositional?

- We approximate the meaning of sentences and discourses by their truth conditions.
- But there are truth-conditionally equivalent sentences that behave differently in discourses.
  One of the ten balls is not in the bag. It is under the sofa.
  ? Nine of the ten balls are in the bag. It is under the sofa.
- Conclusion: Discourse semantics can't be compositional.

The representationality debate

- A key feature of type theory/Montague grammar is that it is non-representational:
  - semantics construction is compositional
  - interpretation of semantic representations is compositional
  - Hence, we could in principle map sentences directly to meanings without semantic representations.

The representationality debate

- If we give up compositional interpretation, we can't eliminate semantic representations like this; such an approach is called representational.
- The point about representational approaches is that meaning isn't all there is to a sentence.
- Psychological reality of semantic representations?

- DRT is not interpreted compositionally, and therefore it is a representational approach.

Verifying embeddings for conditionals (final)

- An embedding f of K into M verifies K in M:
  $f \models_M K$ iff f verifies every condition $\alpha \in C_K$.
- f verifies condition α in M ($f \models_M \alpha$):
  - (i) $f \models_M R(x_1, \ldots, x_n)$ iff $\langle f(x_1), \ldots, f(x_n) \rangle \in V_M(R)$
  - (ii) $f \models_M x = a$ iff $f(x) = V_M(a)$
  - (iii) $f \models_M x = y$ iff $f(x) = f(y)$
  - (iv) $f \models_M K_1 \Rightarrow K_2$ iff for all $g \supseteq_{U_{K_1}} f$ s.t. $g \models_M K_1$ there is a $h \supseteq_{U_{K_2}} g$ s.t. $h \models_M K_2$

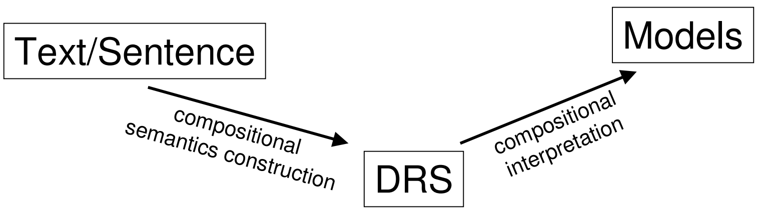
Is discourse semantics compositional?

- We approximate the meaning of sentences and discourses by their truth conditions.
- But there are truth-conditionally equivalent sentences that behave differently in discourses.
  One of the ten balls is not in the bag. It is under the sofa.
  ? Nine of the ten balls are in the bag. It is under the sofa.
- Conclusion: Discourse semantics can't be compositional.

Is discourse semantics compositional?

- We approximate the meaning of sentences and discourses by their truth conditions.
- But there are truth-conditionally equivalent sentences that behave differently in discourses.
  One of the ten balls is not in the bag. It is under the sofa.
  ? Nine of the ten balls are in the bag. It is under the sofa.
- Conclusion: Discourse semantics can't be compositional.
- Alternative conclusion: Truth conditions are not a sufficient approximation for discourse semantics!

Putting the compositionality into DRT



λ-DRT etc.

Dynamic Predicate Logic

DRT and Dynamic Predicate Logic (DPL)

- DPL is a dynamic theory of meaning, just like DRT: The meaning of a sentence is its potential for changing the context.
- In contrast to DRT, DPL is admits compositional interpretation and is non-representational.
- The DRT approach:
  - Alternative representations (DRSs)
  - Interpretation not fully compositional
- The DPL approach:
  - Conventional representations (predicate logic)
  - Interpretations are compositional, but more complex

Semantics of programming languages

- DPL was inspired by concepts from program verification (denotational semantics of programming languages)
- A program π denotes a set of pairs of start and end configurations: $\langle f,g \rangle \in [[\pi]]^M$ iff g is an end configuration that can be reached from the start configuration f by running the program π.
- Semantics of complex programs can be determined compositionally: e.g. $\langle f,g \rangle \in [[\pi_1;\pi_2]]^M$ iff there is an intermediate configuration h that can be reached from f by running π_1 ($\langle f,h \rangle \in [[\pi_1]]^M$) and from which g can be reached by running π_2 ($\langle h,g \rangle \in [[\pi_2]]^M$).

DPL: Formulas as programs

- Logical formulas are programs.
- Contexts are configurations.
- Represent them as variable assignments.
- A formula denotes a set of pairs of start and end configurations (input and output assignments).
- Certain formulas and connectives are instructions for changing the assignments. E.g. "∃x" modifies the value of x by overwriting it with an arbitrary individual from the universe.
- Other formulas are tests: "Fx" checks whether the value of x in the current assignment has the property F.

DPL: Representations

- The syntax of DPL is the syntax of first-order predicate logic.
- Translation of NL expressions into DPL: "a" and "every" into the respective quantifier, pronouns into (possibly free) variables.

Example:

Somebody works. She is successful.

$(\exists x\ \mathrm{work}(x)) \wedge \mathrm{successful}(x)$

Note: We won't say how to do semantics construction (or anaphora resolution) for DPL. This is as problematic as for standard FOL.

DPL: Interpretation

- The model structures for DPL are the model structures for first-order predicate logic.
- The only thing that changes is the interpretation of formulas: Denotations are sets of pairs of input and output assignments.
- A formula is true in a model structure M for a given input assignment if the formula can be "processed" completely and leads to an output assignment.

DPL Interpretation: An informal example

Let's determine whether "(∃x work(x)) ∧ successful(x)“ is true relative to an input assignment g and a model structure M=< U,V>:

- We process the first conjunct first. The "∃x" instructs us to change the value of g(x) to an arbitrary individual; let's call the resulting assignment h. (We write "h[x]g": you get h from g by overwriting the value of x, i.e. g and h differ at most in x.)
- We test whether h(x) satisfies the predicate "work".
- We hand the current assignment h over to the second conjunct and test whether the value of the (free!) variable x satisfies the predicate "successful". The variable still has the same value that h assigns to it.
- If both tests were positive for at least one possible h[x]g, then the formula is true.

DPL: Interpretation (formal)

- Terms are interpreted as in standard FOL (relative to a model structure and a variable assignment):

  $[[x]]^{M,h} = h(x)$ if x is a variable

  $[[a]]^{M,h} = V_M(a)$ if a is an individual constant

- Formulas are interpreted as binary relations between assignments:

  $$[[A]]^{M} = \{\langle g,h \rangle \mid \ldots \}$$

- This has analogies to the interpretation of standard FOL:

  $[[A]]^{M,h} = 1$ iff ... / $[[A]]^{M} = \{h \mid \ldots\}$

DPL: Interpretation (connectives)

- Terms: $[[x]]^{M,h} = h(x)$ if x is a variable

  $[[a]]^{M,h} = V_M(a)$ if a is an individual constant

- Formulas:

  $[[R(t_1,\ldots, t_n)]]^M = \{ \langle g,h \rangle \mid h = g \wedge \langle [[t_1]]_h \ldots [[t_n]]_h \rangle \in V_M(R)\}$

  $[[t_1 = t_2]]^M = \{ \langle g,h \rangle \mid h = g \wedge [[t_1]]_h = [[t_2]]_h\}$

  $[[\phi \wedge \psi]]^M = \{ \langle g,h \rangle \mid \exists k: \langle g,k \rangle \in [[\phi]]^M \wedge \langle k,h \rangle \in [[\psi]]^M \}$

  $[[\exists x\phi]]^M = \{ \langle g,h \rangle \mid \exists k: k[x]g \wedge \langle k,h \rangle \in [[\phi]]^M\}$

  $[[\phi \rightarrow \psi]]^M = \{ \langle g,h \rangle \mid h = g \wedge \forall k: \langle h,k \rangle \in [[\phi]]^M \Rightarrow \exists j: \langle k,j \rangle \in [[\psi]]^M\}$

  $[[\neg\phi]]^M = \{ \langle g,h \rangle \mid h = g \wedge \neg\exists k: \langle h,k \rangle \in [[\phi]]^M\}$

  $[[\phi \vee \psi]]^M = \{ \langle g,h \rangle \mid h = g \wedge \exists k: \langle h,k \rangle \in [[\phi]]^M \vee \langle h,k \rangle \in [[\psi]]^M\}$

  $[[\forall x\phi]]^M = \{ \langle g,h \rangle \mid h = g \wedge \forall k: k[x]h \Rightarrow \exists m: \langle k,m \rangle \in [[\phi]]^M\}$

Existential quantifier and conjunction

- *Somebody works. She is successful.*
  $(\exists x\ work(x)) \wedge successful(x)$
- DPL interpretation:
  $\langle g, h\rangle \in [[\ (\exists x\ work(x)) \wedge successful(x)\ ]]^M$
  iff there is a k such that
  $\langle g, k\rangle \in [[\ \exists x\ work(x)\ ]]^M$ and
  $\langle k, h\rangle \in [[\ successful(x)\ ]]^M$
  iff there is a k such that
  $k[x]g$ and $k(x) \in V_M(work)$ and
  $k = h$ and $k(x) \in V_M(successful)$
- $[[(\exists x\ work(x)) \wedge successful(x)]]^M =$
  $\{\langle g, h\rangle |\ h[x]g \text{ and } h(x) \in V_M(work) \text{ and } h(x) \in V_M(successful)\}$

Existential quantifier and implication

- *If somebody works, she is successful.*
  $(\exists x\ work(x)) \rightarrow successful(x)$
- DPL interpretation:
  $\langle g, h\rangle \in [[\ (\exists x\ work(x)) \rightarrow successful(x)\ ]]^M$
  iff $g = h$ and for all k: if $\langle h, k\rangle \in [[\ \exists x\ work(x)\ ]]^M$,
  then there is a j such that $\langle k, j\rangle \in [[\ successful(x)\ ]]$
  iff $g = h$ and for all k: if $k[x]h$ and $k(x) \in V_M(work)$,
  then there is a j such that $k = j$ and $j(x) \in V_M(successful)$
  iff $g = h$ and for all k: if $k[x]h$ and $k(x) \in V_M(work)$,
  then $k(x) \in V_M(successful)$
- $[[\ (\exists x\ work(x)) \rightarrow successful(x)\ ]]^M =$
  $\{\ \langle g, g\rangle\ |$ for all k: if $k[x]g$ and $k(x) \in V_M(work)$, then $k(x) \in V_M(succ.)\ \}$

DPL Interpretation: Alternative Notation

- Alternative Notation: „$g[[\ \phi\ ]]h$" for „$\langle g,h\rangle \in [[\ \phi\ ]]$"

$g[[R(t_1,\dots, t_n)]]h$	iff	$h = g \wedge \langle [[t_1]]_h \dots [[t_n]]_h\rangle \in V(R)$
$g[[t_1 = t_2]]h$	iff	$h = g \wedge [[t_1]] = [[t_2]]$
$g[[\neg\phi]]h$	iff	$h = g \wedge \neg\exists k: h[[\phi]]\, k$
$g[[\phi \wedge \psi]]h$	iff	$\exists k: g[[\phi]]k \wedge k[[\psi]]h$
$g[[\phi \vee \psi]]h$	iff	$h = g \wedge \exists k: h[[\phi]]k \vee h[[\psi]]k$
$g[[\phi \rightarrow \psi]]h$	iff	$h = g \wedge \forall k: h[[\phi]]k \Rightarrow \exists j: k[[\psi]]j$
$g[[\exists x\phi]]h$	iff	$\exists k: k[x]g \wedge k[[\phi]]h$
$g[[\forall x\phi]]h$	iff	$h = g \wedge \forall k: k[x]h \Rightarrow \exists m: k[[\phi]]m$

Truth and validity

- A formula φ is true in M with respect to an input assignment g iff there is a h such that $\langle g, h\rangle \in [[\varphi]]^M$
- A formula φ is true in M iff φ is true in M wrt. every input assignment g.
- A formula φ is valid iff φ is true in every model structure M.

Static and dynamic connectives

- A connective C is internally dynamic iff the left-hand subformula can change the input assignment for the right-hand subformula (i.e. can affect variables there).
- A connective C is externally dynamic iff the output assignment of a formula with main connective C can be different than the input assignment (i.e. can affect the later context).
- Formulas whose main connective is externally static are called tests: From $\langle g, h \rangle \in [[\varphi]]^M$ follows $g = h$

Overview of DPL connectives

connective	externally	internally
$\neg$	s	--
$\wedge$	d	d
$\vee$	s	s
$\rightarrow$	s	d
$\forall$	s	d
$\exists$	d	d

Equivalence

- Satisfaction set of a formula φ in M:

  $\backslash\varphi\backslash_M = \{ g \mid \exists h: \langle g, h\rangle \in [[\varphi]]^M \}$
- s-equivalence (static equivalence):

  $\varphi \Leftrightarrow_s \psi$ iff for all M: $\backslash\varphi\backslash_M = \backslash\psi\backslash_M$
- Equivalence (dynamic/full equivalence):

  $\varphi \Leftrightarrow \psi$ iff for all M: $[[\varphi]]^M = [[\psi]]^M$

- Equivalent formulas are always statically equivalent too.

Logical properties of DPL

- The following equivalences hold:
  - $(\exists xA) \wedge B \Leftrightarrow \exists x(A \wedge B)$
  - $(\exists xA) \rightarrow B \Leftrightarrow \forall x(A \rightarrow B)$
  - $(A \wedge B) \wedge C \Leftrightarrow A \wedge (B \wedge C)$
  - $A \rightarrow (B \rightarrow C) \Leftrightarrow (A \wedge B) \rightarrow C$
  - $A \vee B \Leftrightarrow B \vee A$
- The following equivalences don't hold:
  - $\neg\forall xA \Leftrightarrow \exists x\neg A$
  - $A \wedge B \Leftrightarrow_s B \wedge A$
  - $A \Leftrightarrow_s A \wedge A$

Definability of connectives

- $\vee$, $\rightarrow$ and $\forall$ can be defined from $\neg$, $\wedge$ und $\exists$.
- But not vice versa!

- Equivalences:

  $A \rightarrow B \Leftrightarrow \neg(A \wedge \neg B)$

  $A \vee B \Leftrightarrow \neg(\neg A \wedge \neg B)$

  $A \vee B \Leftrightarrow (\neg A) \rightarrow B$

  $\forall x\, A \Leftrightarrow \neg\exists x \neg A$

- Non-equivalences:

  $A \wedge B \Leftrightarrow \neg(A \rightarrow \neg B)$

  $A \wedge B \Leftrightarrow_s \neg(\neg A \vee \neg B)$

  $A \rightarrow B \Leftrightarrow_s \neg A \vee B$

  $\exists x A \Leftrightarrow \neg\forall x \neg A$

Entailment

- Static entailment:

  $\varphi \mid=_s \psi$ iff for all M, g:
  If φ is true in M for g, then ψ is true in M for g.

- Meaning Inclusion:

  $\varphi \leq \psi$ iff $[[\varphi]]^M \leq [[\psi]]^M$

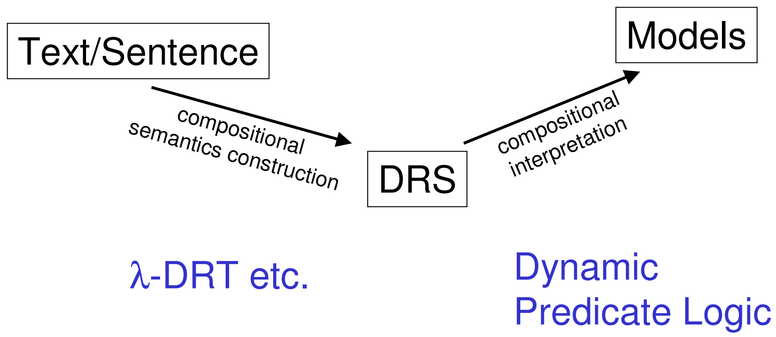
- Dynamic entailment:

  $\varphi \mid= \psi$ iff for all M,g,h:
  If $\langle g, h\rangle \in [[\varphi]]$, then there is a k s.t. $\langle h, k\rangle \in [[\psi]]$.

DPL: Summary

- We can give a compositional interpretation to a theory of dynamic semantics: relation between variable assignments.
- DPL uses standard syntax of predicate logic, but the different interpretation makes for interesting logical properties; e.g., some equivalences break.
- We can translate DRSs into DPL formulas and further into static PL formulas (see exercise).
- DRT can be equipped directly with a DPL-style interpretation.

Putting the compositionality into DRT



Dynamic semantics and semantics construction

- DPL is compositional: The denotations of DPL formulas can be determined solely from the denotations of the subexpressions.
- DPL is a first-order logic, and so doesn't say anything about semantics construction.
- Question: Can we get compositional semantics construction for dynamic theories of meaning?
- Try to combine
  - type theory (higher-order logic / λ-calculus) and
  - first-order dynamic semantics (e.g. DRT or DPL)

Higher-order dynamic semantics

- Our goal now: Get an idea of why getting a clean higher-order dynamic semantics formalism is not trivial.
- Differences between variables and discourse referents.
- Some formalisms:
  - Dynamic Montague Grammar (Groenendijk & Stokhof 1990)
  - Lambda-DRT (Bos et al. 1993, Kuschert et al. 1996)
  - Compositional DRT (Muskens 1996)
  - Dynamic Lambda Calculus (Kuschert 1998)

Naive λ-DRT: just allow λ-abstraction over DRSs

- *every student* ⇒ λG

z	⇒ G(x)
student(z)	

  alternative notation: λG [∅ | [z | student(z)] ⇒ G(z)]

- *works* ⇒ λx [∅ | work(x)]

  An expression consists of a lambda prefix and a (partially instantiated) DRS.

Naive λ-DRT: β-reduction of λ-DRSs

- *every student works*

  ⇒ λG[∅ | [z | student(z)] ⇒ G(z)]](λx.[∅ | work(x)])

  ⇔ [∅ | [z | student(z)] ⇒ λx.[∅ | work(x)](z)]

  ⇔ [∅ | [z | student(z)] ⇒ [∅ | work(z)]]

λ -DRT: The "Merge" operation

- *a student* ⇒ λG ([z | student(z)]; G(z))
- *works* ⇒ λx [∅ | work(x)]

- *A student works*

  ⇒ λG ([z | student(z)];G(z))(λx.[∅ | work(x)])

  ⇔ [z | student(z)]; λx.[∅ | work(x)](z)

  ⇔ [z | student(z)]; [∅ | work(z)]

  ⇔ [z | student(z), work(z)]

Merge

- The "merge" operation on DRSs combines two DRSs (conditions and universes).
- It has a similar function as the beta reduction in type theory: Replace a complex formula (the ";"-combination of two DRSs) by an equivalent simpler formula.
- It is also similar to DPL conjunction.
- Let $K_1 = [\, U_1 \mid C_1 \,]$ and $K_2 = [\, U_2 \mid C_2 \,]$.

  We define: $K_1; K_2 = [\, U_1 \cup U_2 \mid C_1 \cup C_2 \,]$

  under the assumption that no discourse referent $u \in U_2$ occurs free in a condition $\gamma \in C_1$.

Naive λ-DRT: The problem

- *A student works. She is successful.*
- Compositional analysis:
- λK λK'(K;K')([z | student(z), work(z)])([|successful(z)])

  ⇔ λK'([z | student(z), work(z)];K')([|successful(z)])

? ⇔ [z | student(z), work(z)];[|successful(z)]

  ⇔ [z | student(z), work(z), successful(z)]

Via the interaction of β-reduction and DRS-binding, discourse referents are "captured"!

Higher-order DRT: The challenge

- Via the interaction of β-reduction and DRS-binding, discourse referents are captured.
- But the β-reduced DRS must still be equivalent to the original DRS!
- This means that we somehow have to encode the potential for capturing discourse referents into the denotation of a λ-DRS. Getting this right is tricky.
- Discourse referents and bound variables behave differently! (Discourse referents may be captured.)

Compositional DRT

- The most transparent formalism of higher-order dynamic semantics is Muskens' Compositional DRT.
- Realise discourse referents as individual constants.
- Encode value assignments for the discourse referents directly into terms of (static) type theory.
- Uses big terms and big types. Representations remain reasonably readable by using notational macros.

Summary

- The quest for compositional dynamic semantics.
- Dynamic Predicate Logic (DPL):
  - Use standard syntax of predicate logic
  - with a compositional dynamic interpretation.
  - This is still first-order, so the usual problems with semantics construction apply.
- Higher-order theories of dynamic semantics:
  - Interaction of β-reduction and DRS-binding "captures" discourse referents.
  - Challenge: Build a formalism that models this properly.