Introduction to Formal Semantics
Lecture 7: Presupposition

Volha Petukhova & Nicolaie Dominik Dascalu

Spoken Language Systems Group
Saarland Univeristy

13.06.2022

UNIVERSITÄT
DES
SAARLANDES

Overview for today

- Recap: inference Lecture 2
- Presupposition revisited
- Definedness condition
- Pragmatic Theories

Reading:

- Coppock, E., and Champollion, L. (2021). Invitation to formal semantics. Manuscript, Boston University and New York University (Ch.8)

Quizz (last week)

Simplify the following expression step-by-step

$$[\lambda Q.\forall x[Linguist(x) \rightarrow Q(x)]](\lambda v_1.Offended(j, v_1))$$
$$\Downarrow$$

Quizz (last week)

Simplify the following expression step-by-step

$$[\lambda Q.\forall x[Linguist(x) \rightarrow Q(x)]](\lambda v_1.Offended(j, v_1))$$
$$\Downarrow$$
$$\forall x[Linguist(x) \rightarrow \lambda v_1.Offended(j, v_1)(x)]$$

Quizz (last week)

Simplify the following expression step-by-step

$$[\lambda Q.\forall x[Linguist(x) \rightarrow Q(x)]](\lambda v_1.Offended(j, v_1))$$
$$\Downarrow$$
$$\forall x[Linguist(x) \rightarrow \lambda v_1.Offended(j, v_1)(x)]$$
$$\Downarrow$$

Quizz (last week)

Simplify the following expression step-by-step

$$[\lambda Q.\forall x[Linguist(x) \rightarrow Q(x)]](\lambda v_1.Offended(j, v_1))$$
$$\Downarrow$$
$$\forall x[Linguist(x) \rightarrow \lambda v_1.Offended(j, v_1)(x)]$$
$$\Downarrow$$
$$\forall x[Linguist(x) \rightarrow Offended(j, x)]$$

Recap: summary

Semantic Theories:

- contrast between assertion and presupposition (of an expression)
- presupposition as a different type of inference than logical implication or entailment
- presupposition as relation between sentences vs. statements (or even between speakers and statements)
- ambiguity of negation: structural (Russell) vs. lexical (Frege, Strawson)

Presupposition Projection (recap)

Presuppositions differ from semantic entailments because:

- presuppositions **survive** in contexts where entailments disappear (e.g. negation, modals, attitude verbs)
- presuppositions are **defeasible** e.g. they can disappear in contexts where entailments survive

$\Rightarrow$ **Presupposition projection**

Presupposition Defeasibility/Cancellation (recap)

Contextual defeasibility: the presupposition can be cancelled by the linguistic or non-linguistic context within the context of the same sentence or beyond the sentence, in the larger discourse context.

Surface-structure defeasibility: the presupposition is cancelled by a given surface-structure context (e.g. if-then, or) –*presupposition projection* problem

Presupposition Defeasibility/Cancellation (recap)

A presupposition can be cancelled by the linguistic or non-linguistic context within the context of the same sentence or beyond the sentence, in the larger discourse context.

1. When the linguistic context makes the presupposition inconsistent.
2. When is it common knowledge that the presupposition is false
3. When what is said, taken together with background assumptions makes the presupposition inconsistent.
4. When evidence for truth of the presupposition is being weighed and rejected

The projection problem has been dealt with using dynamic semantics, where the denotation of a sentence is a **"context change potential"**: a function that can update a discourse context.

Surface-Structure Defeasibility (recap)

There are cases of **intra-sentential cancellation or suspension** of presuppositions.

- A presupposition can "survive" i.e. project. We saw cases of this when the intra-sentential context contains *a negation*, *a modal*, *a disjunction and a conditional*.
- A presupposition can be "overtly cancelled or suspended" by the intra-sentential context.
- A presupposition can be "filtered" (i.e. partially let through) by intra-sentential contexts such as *and*, *if ... then*, *but*, *suppose that*

John left work earlier again.

- John doesn't regret leaving work early again because in fact he never did.
- John left work early again. What you mean again? He never did this before.
- John left work early again if in fact he ever did.
- John would leave work early again if he had a job.
- I don't know whether John left work early again.
- John died before leaving work early again.

Semantic Presupposition: problems

Problem 1 Presupposition failure (= the presupposition is false in context)

(1) King of France is bald.
When uttered on May 13 2005, the presupposition is false

Problem 2 Presupposition cancellation (= the presupposition is "removed" in context)

(2) If John has a wife, she (John's wife) likes gardening.
If John is married, his wife likes gardening.
Either John got a divorce or his wife is helping him with work.
$\not\gg$ John has a wife.

Classical logic cannot handle presupposition failure; nor can it explain why sentences whose presuppositions are not satisfied are odd. To remedy this, semantic theories of presuppositions use **multi-valued logics**, which include true, false and neither-true-nor-false as possible truth-values.

Classical logic cannot account for the cancelling of presuppositions due to information available in the context. A possible remedy is to use a **non-monotonic logic**.

Semantic Presupposition: problems (cont.)

Moreover, many cases of what one would want to call presupposition are **not truth-conditional effects**, and are also strongly context-dependent. Therefore, the distinction between **semantic** and **pragmatic** presupposition is untenable and has been abandoned.

- Peter Frederick Strawson (1919-2006)
- Introduces an important distinction namely the distinction between **sentences** and **use of sentences** i.e. **statements**.
- Sentences aren't true or false; Statements, i.e. $\langle$ Sentence, Context $\rangle$ pairs, are.

Example

The King of France is wise.

- True in 1670.
- False in 1770.
- Neither true nor false in 1970.

Presuppositions are conventions about **use** of referring expressions: **a statement A presupposes a statement B iff B is a precondition for the truth or falsity of A.**

Pragmatic Theories of Presuppositions (recap)

Besides the (mostly abandoned) semantic attempts of modelling the projection problem, there are two main types of pragmatic theories:

(i) Theories based on a "static" semantics: Gazdar (1979), Karttunen (1973)

(ii) Theories based on dynamic semantics: Karttunen (1974), Heim (1982), Van der Sandt (1988, 1992), Beaver (1995), Geurts (1997), etc.

- Presuppositions are neither viewed as referring expressions nor as semantic entailments but as context-dependent (i.e. pragmatic) phenomena.
- When a presupposition conflicts with previous information, this presupposition
  - does not give rise to inconsistency
  - is lifted (i.e. cancelled) or altered (i.e. filtered) to resolve the conflict.

Presupposition Problems: summary

Problems with Semantic Theories:

- Cannot account for presupposition defeasibility.
  Proposed solution: Defeasibility is captured through binding or accommodation to a sub-level of the DRS (v.d. Sandt)

Presupposition Problems: summary

Problems with Semantic Theories:

- Cannot account for presupposition defeasibility.
  Proposed solution: Defeasibility is captured through binding or accommodation to a sub-level of the DRS (v.d. Sandt)

Problems with Static Pragmatic Theories

- Semantic and presuppositional information are represented separately which yields wrong predictions concerning the communicated meaning. Proposed solution: Semantic and presuppositional information are represented in a uniform way. Problem does not occur (v.d. Sandt)

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.

Neither theory can explain why sentences whose presuppositions are not satisfied are odd. >> There are exactly two theories. (*quantifiers*)

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.
Neither theory can explain why sentences whose presuppositions are not satisfied are odd. >> There are exactly two theories. (*quantifiers*)
John's daughter will come. >> John has a daughter. (*possessives*)

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.
Neither theory can explain why sentences whose presuppositions are not satisfied are odd. >> There are exactly two theories. (*quantifiers*)
John's daughter will come. >> John has a daughter. (*possessives*)
John knows that Mary hates Bill. >> Mary hates Bill (*cognitive factive verbs*).

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. $>>$ There is at least one man and he kissed the woman who loved him.
Neither theory can explain why sentences whose presuppositions are not satisfied are odd. $>>$ There are exactly two theories. (*quantifiers*)
John's daughter will come. $>>$ John has a daughter. (*possessives*)
John knows that Mary hates Bill. $>>$ Mary hates Bill (*cognitive factive verbs*).
John is happy that Mary agrees to marry him. $>>$ Mary agrees to marry John. (*emotive factive verbs*)

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.
Neither theory can explain why sentences whose presuppositions are not satisfied are odd. >> There are exactly two theories. (*quantifiers*)
John's daughter will come. >> John has a daughter. (*possessives*)
John knows that Mary hates Bill. >> Mary hates Bill (*cognitive factive verbs*).
John is happy that Mary agrees to marry him. >> Mary agrees to marry John. (*emotive factive verbs*)
Mary bakes cookies again. >> Mary has baked cookies before. (*additive adverbs*)

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.

Neither theory can explain why sentences whose presuppositions are not satisfied are odd. >> There are exactly two theories. (*quantifiers*)

John's daughter will come. >> John has a daughter. (*possessives*)

John knows that Mary hates Bill. >> Mary hates Bill (*cognitive factive verbs*).

John is happy that Mary agrees to marry him. >> Mary agrees to marry John. (*emotive factive verbs*)

Mary bakes cookies again. >> Mary has baked cookies before. (*additive adverbs*)

It was John who broke the computer. >> Someone broke the computer. (*Clefts*)

Presupposition Triggers

There are certain words/constructions that signal presupposition.

Example

Every man kissed the woman who loved him. >> There is at least one man and he kissed the woman who loved him.
Neither theory can explain why sentences whose presuppositions are not satisfied are odd. >> There are exactly two theories. (*quantifiers*)
John's daughter will come. >> John has a daughter. (*possessives*)
John knows that Mary hates Bill. >> Mary hates Bill (*cognitive factive verbs*).
John is happy that Mary agrees to marry him. >> Mary agrees to marry John. (*emotive factive verbs*)
Mary bakes cookies again. >> Mary has baked cookies before. (*additive adverbs*)
It was John who broke the computer. >> Someone broke the computer. (*Clefts*)
The student is smart. >> There is an unique student in the context. (*definite determiners*)

Definite Descriptions

Take predicates and return the unique individual, thus of type

Definite Descriptions

Take predicates and return the unique individual, thus of type $\langle\langle e, t\rangle, e\rangle$

Definite Descriptions

Take predicates and return the unique individual, thus of type $\langle\langle e, t\rangle, e\rangle$

Definite descriptions convey EXISTENCE and UNIQUENESS

Russell example

The princess smokes.
$\exists x.[Princess(x) \wedge \forall y.[Princess(y) \rightarrow x = y] \wedge Smokes(x)])$
Problems?

Definite Descriptions

Take predicates and return the unique individual, thus of type $\langle\langle e, t\rangle, e\rangle$

Definite descriptions convey EXISTENCE and UNIQUENESS

Russell example

The princess smokes.
$\exists x.[Princess(x) \wedge \forall y.[Princess(y) \rightarrow x = y] \wedge Smokes(x)])$
Problems?

Strawson: EMPTY DESCRIPTIONS are definite descriptions in which nothing satisfies the descriptive content

Definite Descriptions

Take predicates and return the unique individual, thus of type $\langle\langle e, t\rangle, e\rangle$

Definite descriptions convey EXISTENCE and UNIQUENESS

Russell example

The princess smokes.
$\exists x.[Princess(x) \land \forall y.[Princess(y) \rightarrow x = y] \land Smokes(x)])$
Problems?

Strawson: EMPTY DESCRIPTIONS are definite descriptions in which nothing satisfies the descriptive content
Trivalent Strawsonian semantics: true, false and undefined

Definite Descriptions: syntax and semantics

We introduce a special 'undefined individual' of type e and use the symbol $\#_e$ to denote this individual in our meta-language

Syntax Rule: Iota

If ϕ is an expression of type t , and u is a variable of type e, then $\iota u.\phi$ is an expression of type e

Semantic Rule: Iota

$$[\![\iota u.\phi]\!]^{M,g} = \begin{cases} d \ \ if \ \ \{k | [\![\phi]\!]^{M,g^{[u \mapsto k]}} = 1\} = \{d\} \\ \#_e otherwise \end{cases}$$

$the \rightsquigarrow \lambda P.\iota x.P(x)$

Definite Descriptions (cont.)

Example

the president

- DP, e, $\iota x.President(x)$
  - D, $\langle\langle e,t\rangle,e\rangle$, $\lambda P.\iota x.P(x)$
    - the
  - NP, $\langle e,t\rangle$, $\lambda x.President(x)$
    - NP, $\langle e,t\rangle$, $\lambda x.President(x)$
      - president

Definite Descriptions (cont.)

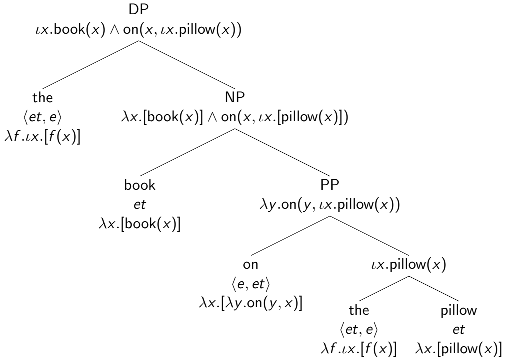
$$[\![\iota x.President(x)]\!]^{M,g} = \begin{cases} d \ \ if \ \ \{k | [\![President(x)]\!]^{M,g^{[x \mapsto k]}} = 1\} = \{d\} \\ \#_e otherwise \end{cases}$$

why undefined value?

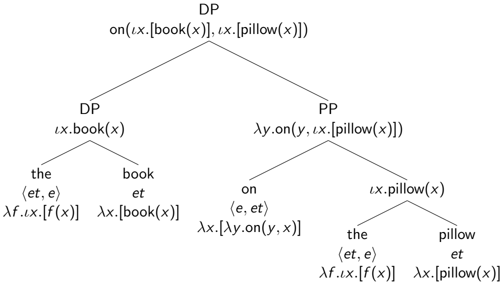
The King of France is bald.
The opera of Mozart is Italian.

Modifiers in Definite Descriptions

(1)



(2)



Which of the structures lead to uninterpretability of a sentence like ‘Ann likes the book on the pillow’ and why?

Defineteness Conditions

Definite determiner ‘the’ is one of the presupposition trigers, what about others? Both, neither, every, etc.

Defineteness Conditions

Definite determiner ‘the’ is one of the presupposition trigers, what about others? Both, neither, every, etc.

∂ operator is defined like $\partial(|P| = 2)$ reads ‘presupposing that there are exactly two Ps’.

Defineteness Conditions

Definite determiner ‘the’ is one of the presupposition trigers, what about others? Both, neither, every, etc.

∂ operator is defined like $\partial(|P| = 2)$ reads ‘presupposing that there are exactly two Ps’.

neither $\rightsquigarrow \lambda P \lambda Q.[\partial(|P| = 2) \wedge \neg \exists x.[P(x) \wedge Q(x)]]$

Defineteness Conditions

Definite determiner ‘the’ is one of the presupposition trigers, what about others? Both, neither, every, etc.

∂ operator is defined like $\partial(|P| = 2)$ reads ‘presupposing that there are exactly two Ps’.

neither $\rightsquigarrow \lambda P \lambda Q.[\partial(|P| = 2) \wedge \neg \exists x.[P(x) \wedge Q(x)]]$
every $\rightsquigarrow \lambda P \lambda Q.[\partial(\exists x.P(x)) \wedge \forall x.[P(x) \rightarrow Q(x)]]$

Defineteness Conditions

Definite determiner 'the' is one of the presupposition trigers, what about others? Both, neither, every, etc.

∂ operator is defined like $\partial(|P| = 2)$ reads 'presupposing that there are exactly two Ps'.

neither $\rightsquigarrow \lambda P \lambda Q.[\partial(|P| = 2) \wedge \neg\exists x.[P(x) \wedge Q(x)]]$
every $\rightsquigarrow \lambda P \lambda Q.[\partial(\exists x.P(x)) \wedge \forall x.[P(x) \rightarrow Q(x)]]$

Syntax Rule: Definedeness Conditions

If ϕ is an expression of type t , then $\partial(\phi)$ is an expression of type t

Defineteness Conditions

Definite determiner ‘the’ is one of the presupposition trigers, what about others? Both, neither, every, etc.

∂ operator is defined like $\partial(|P| = 2)$ reads ‘presupposing that there are exactly two Ps’.

neither $\rightsquigarrow \lambda P \lambda Q.[\partial(|P| = 2) \wedge \neg\exists x.[P(x) \wedge Q(x)]]$
every $\rightsquigarrow \lambda P \lambda Q.[\partial(\exists x.P(x)) \wedge \forall x.[P(x) \rightarrow Q(x)]]$

Syntax Rule: Definedeness Conditions

If ϕ is an expression of type t , then $\partial(\phi)$ is an expression of type t

Semantic Rule: Defineteness Conditions

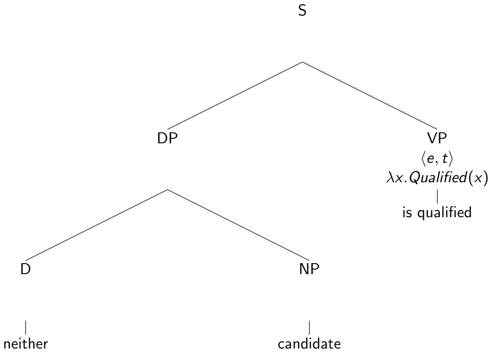
If ϕ is an expression of type t , then:

$$[\![\partial(\phi)]\!]^{M,g} = \begin{cases} 1 \; if \; [\![\phi]\!]^{M,g} = 1 \\ \#_e otherwise \end{cases}$$

Defineteness Condition (cont.)

Example

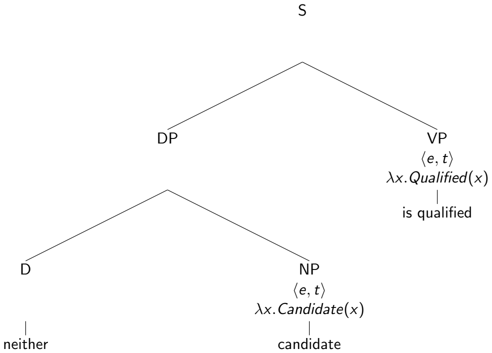
Neither candidate is qualified.



Defineteness Condition (cont.)

Example

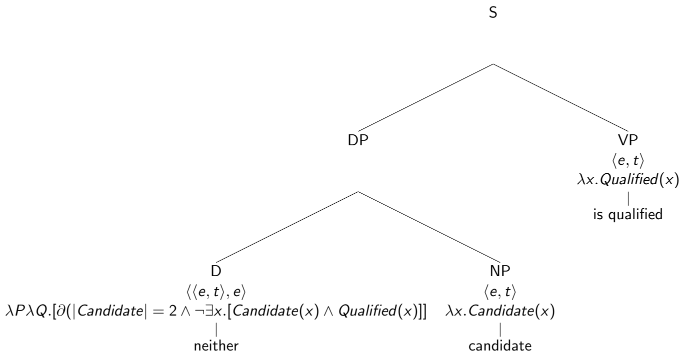
Neither candidate is qualified.



Defineteness Condition (cont.)

Example

Neither candidate is qualified.



Defineteness Condition (cont.)

Example

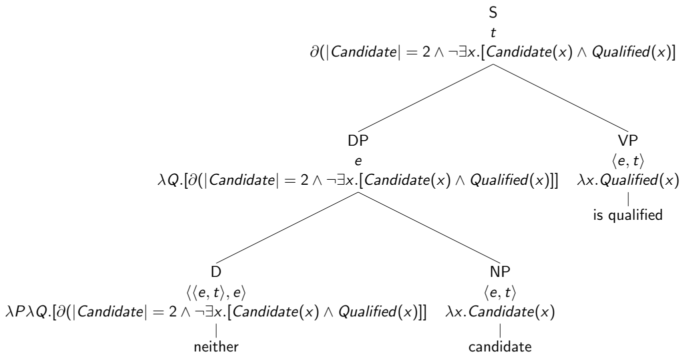
Neither candidate is qualified.

- S
  - DP: e, $\lambda Q.[\partial(|Candidate| = 2 \wedge \neg\exists x.[Candidate(x) \wedge Qualified(x)]]$
    - D: $\langle\langle e,t\rangle, e\rangle$, $\lambda P \lambda Q.[\partial(|Candidate| = 2 \wedge \neg\exists x.[Candidate(x) \wedge Qualified(x)]]$
      - neither
    - NP: $\langle e,t\rangle$, $\lambda x.Candidate(x)$
      - candidate
  - VP: $\langle e,t\rangle$, $\lambda x.Qualified(x)$
    - is qualified

Defineteness Condition (cont.)

Example

Neither candidate is qualified.



Weak Kleene Connectives

$\wedge$	T	F	#
T	T	F	#
F	F	F	#
#	#	#	#

$\vee$	T	F	#
T	T	T	#
F	T	F	#
#	#	#	#

	$\neg$
T	F
F	T
#	#

	∂
T	T
F	#
#	#

Idea: We see as a $\#$ "contaminating" (or nonsense) value, which does not allow us to deduce anything if there is a presupposition failure somewhere.
In Weak Kleene, any local presupposition failure leads to a global failure. If $[\![S]\!] = \#$, then any sentence that contains S denotes $\#$.

Strong Kleene Connectives

∧	T	F	#
T	T	F	#
F	**F**	F	**F**
#	#	**F**	#

∨	T	F	#
T	T	**T**	**T**
F	T	F	#
#	**T**	#	#

	¬
T	**F**
F	T
#	#

	∂
T	**T**
F	#
#	#

Idea: Idea: We see # in one argument as "ignorance" (unknown) - it still allows us to deduce the result from the value of the other argument.

Universal & Existential Quantifiers

Every boy loves his cat.
$\forall x.[Boy(x) \rightarrow Loves(x.\iota y.[Cat(y) \wedge Has(x,y)])]$ (UNIVERSAL PROPOSITION)
every element of D_e is $[Boy(x) \rightarrow Loves(x.\iota y.[Cat(y) \wedge Has(x,y)])]$
x loves his cat (SCOPE PROPOSITION) where x is 1 or # should UNIVERSAL PROPOSITION be 1 or #?

Muskens (1995) sees universal claim as a big conjunction

$$[\![\forall x.\phi]\!]^{M,g} = \begin{cases} 1 \textit{ if } [\![\phi]\!]^{M,g^{[x\mapsto k]}} = 1 \textit{ for all } k \in D \\ \# \textit{ if } [\![\phi]\!]^{M,g^{[x\mapsto k]}} = \# \textit{ for some } k \in D \\ 0 \textit{ otherwise} \end{cases}$$

and existential claim as big disjunction

$$[\![\exists x.\phi]\!]^{M,g} = \begin{cases} 0 \textit{ if } [\![\phi]\!]^{M,g^{[x\mapsto k]}} = 0 \textit{ for all } k \in D \\ \# \textit{ if } [\![\phi]\!]^{M,g^{[x\mapsto k]}} = \# \textit{ for some } k \in D \\ 1 \textit{ otherwise} \end{cases}$$

Identity

The King of France is the Grand Sultan of Germany.

LaPierre (1992) defines identity between two terms as follows:

- if neither α nor β denotes the undefined individual, then $[\![\alpha = \beta]\!]^{M,g} = 1$ if $[\![\alpha]\!]^{M,g} = [\![\beta]\!]^{M,g}$, and 0 otherwise.
- If one α or β denotes the undefined individual, then $[\![\alpha = \beta]\!]^{M,g} = 0$
- If both denote the undefined individual, then $[\![\alpha = \beta]\!]^{M,g} = \#$ is undefined (not enough is "known" about the objects to determine that they are the same or distinct).

Predication with Undefined Individuals

Semantic Rule: Existence Predicate

$[\![\mathit{Exists}(\alpha)]\!]^{M,g} = 1$ if $[\![\alpha]\!]^{M,g} \neq \#_e$ and 0 otherwise

Predict truth value of ‘The Golden Mountain does not exist’

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$
Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$
Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$
For functional types $\langle \sigma, \tau \rangle$ there is a domain $D_{\langle \sigma, \tau \rangle}$ consisting of the (total) functions from D_σ to D_τ.

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$
Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$
For functional types $\langle\sigma, \tau\rangle$ there is a domain $D_{\langle\sigma,\tau\rangle}$ consisting of the (total) functions from D_σ to D_τ.
For every type, there is also an 'undefined individual' of that type, which we refer to $\#_e$.

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$
Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$
For functional types $\langle \sigma, \tau \rangle$ there is a domain $D_{\langle \sigma, \tau \rangle}$ consisting of the (total) functions from D_σ to D_τ.
For every type, there is also an 'undefined individual' of that type, which we refer to $\#_e$.

- The domain of individuals D_e contains at least one individual.

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$
Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$
For functional types $\langle \sigma, \tau \rangle$ there is a domain $D_{\langle \sigma, \tau \rangle}$ consisting of the (total) functions from D_σ to D_τ.
For every type, there is also an 'undefined individual' of that type, which we refer to $\#_e$.

- The domain of individuals D_e contains at least one individual.
- I is an interpretation function, assigning a denotation to all of the constants of the language. The denotation of a constant of type τ is a member of D_τ.

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$

Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$

For functional types $\langle\sigma, \tau\rangle$ there is a domain $D_{\langle\sigma,\tau\rangle}$ consisting of the (total) functions from D_σ to D_τ.

For every type, there is also an 'undefined individual' of that type, which we refer to $\#_e$.

- The domain of individuals D_e contains at least one individual.
- I is an interpretation function, assigning a denotation to all of the constants of the language. The denotation of a constant of type τ is a member of D_τ.

An assignment g is a total function whose domain consists of the variables of the language such that if u is a variable of type τ then $g(u) \in D_\tau$.

Semantics of ∂L

Type e is associated with the domain of individuals $D_e = D$
Type t is associated with the domain of truth values $D_t = \{1, 0, \#\}$
For functional types $\langle\sigma, \tau\rangle$ there is a domain $D_{\langle\sigma,\tau\rangle}$ consisting of the (total) functions from D_σ to D_τ.
For every type, there is also an 'undefined individual' of that type, which we refer to $\#_e$.

- The domain of individuals D_e contains at least one individual.
- I is an interpretation function, assigning a denotation to all of the constants of the language. The denotation of a constant of type τ is a member of D_τ.

An assignment g is a total function whose domain consists of the variables of the language such that if u is a variable of type τ then $g(u) \in D_\tau$.
We use $g[x \mapsto d]$ to denote an assignment function which is exactly like g with the possible exception that $g(x) = d$.

Quizz for Today

TBA