Introduction to Formal Semantics
Lecture 3: Meaning Representation

Volha Petukhova & Nicolaie Dominik Dascalu

Spoken Language Systems Group
Saarland Univeristy

09.05.2022

UNIVERSITÄT
DES
SAARLANDES

Overview for today

- Recap: Inferences
- Predicate-Argument Structure
- Syntax of Predicate Logic
- Semantics of Predicated Logic

Reading:

- Coppock, E., and Champollion, L. (2021). Invitation to formal semantics. Manuscript, Boston University and New York University (Ch.4)

Quizz (last week)

John left work early again. (a) generate ordinary entailments and all possible presuppositions

- Entailment: John left work
- Entailment: John left work early

? Entailment: John didn't stay at work till the end

? Entailment: John is no longer at work

? Entailment: John left from some place

- Presupposition: John designates somebody or There is one particular mail person that the speaker knows and his name is John
- Presupposition: John has work
- Presupposition: John left work early before (at least once)

? Presupposition: John was employed

? Presupposition: John went to work

? Presupposition: John was at work

John left work early again (cont.)

(b) generate presupposition projections

- John didn't leave work early again.
- Did John leave work early again?
- John could leave work early again.
- John regrets leaving work early again.
- It was John who left work early again.
- I wish John leaves work early again.
- If John left work early again, he will be fired.
- If John left work, then he did this early again.
- If John left work early again, then his wife was happy.
- If John left work early again, then he will work overtime next week.

? If John was at work today, he left work early again.

- If John left work early again, he doesn't need to come back.
- Either John left work early again or he would again have missed his bus.

? Either John left work early again or he didn't come at all.

- Either John left work early again or my clock is broken.

John left work early again (cont.)

(c) give an example of context in which one of the possible presuppositions is cancelled, i.e. defeat the presupposition

- 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.

? Either John got fired or he left work early again.

- 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.

Implicatures

Basic idea: Words are not ambiguous. Rather, they have a core meaning (semantics) which can be augmented by (defeasible) implicatures (pragmatics).

- What is said vs. what is implicated
- Types of implicatures

Conversational Implicatures (CI)

- *natural meaning* (also: literal meaning, sentence meaning, what is said) vs. *non-natural meaning* (also: meaning-nn, speaker meaning)
- Not all inferences that can be drawn from what is said and all the knowledge of the world that a participant has, are part of its communicative content. Only those intended by the speaker are.

Example

A: Can you tell me the time?
B: *Well, the milkman has come.*

Example

A: *I am out of petrol.*
B: *There's a garage just around the corner.*
Standard Conversational Implicature (SCI): A may obtain petrol at the garage just around the corner.

CIs are defeasible, non-detachable (attached to meaning-nn, e.g. ironic) and calculable

Conventional Implicatures (CnvIs)

CnvI are implicatures attached by convention to some particular items

- *But*:
  Same truth-conditional content as *and*.
  CnvI: there is a contrast between the conjuncts.
- *However, Although, Yet*:
  Same truth-conditional content as *and*.
  CnvI: violation of an expected (general) rule
- *Even*:
  Same truth-conditional content as without.
  CnvI: The least likely 'alternative'.
- politeness markers (e.g., forms of address:
  Ge. *du, Sie*, Fr. *tu, vous* Cz. *ty, vy*
- etc.

CnvIs are non-defeasible, detachable and non-calculable

Standard General Implicatures (SGIs)

- scalar implicatures
- clausal implicatures

Example

Some of the boys went to the party.
SQGI: Not all of the boys went to the party.

⟨ all, most, many, some, few ⟩, ⟨ none, not all ⟩, ⟨ n, . . . , 5, 4, 3, 2, 1 ⟩, ⟨ excellent, good ⟩, ⟨ hot, warm ⟩, ⟨ necessarily p, p, possibly p ⟩, ⟨ certain that p, probable that p, possible that p ⟩,⟨ always, often, sometimes ⟩, ⟨ must, should, may ⟩, ⟨ succeed in Ving, try to V, want to V ⟩, ⟨ adore, love, like ⟩

Standard General Implicatures (SGIs), cont.

If S uses some linguistic expression which does not commit her to some embedded proposition p and there is another expression that would commit her so then S implicates that she does not know whether p.

Definition: If S asserts some complex expression r, such that

(i) r contains an embedded sentence p and
(ii) r neither entails nor presupposes that p is true and
(iii) there is an alternative expression r' of roughly equal brevity which does entail or presuppose that p is true

then, by asserting r rather than r', S implicates that she doesn't know whether p is true or false, i.e. S implicates ($\Diamond q$ and $\Diamond \neg q$).

Standard General Implicatures (SGIs), cont.

Clausal Quantity GIs

I believe John is away.

CQGI: I do not know whether John is away. Since there is an alternative expression *I know John is away.* which contains *John is away* and entails it.

Clausal Quantity GIs

The Russians or the Americans have just landed on Mars.

CQGCI: S does not know whether it was the R or the A who has just landed on Mars, possibly even both.

Standard General Implicatures (SGIs), cont.

Because of the several types of implicatures, the implicatures of an expression may not be the simple sum of its implicatures (some implicatures might cancel others).

Clausal cancels Scalar

Some, if not all, of the workers went on strike.
(i) Scalar Implicature of "some": Not all of the workers went on strike
(ii) Clausal Implicature of "if": Possibly all of the workers went on strike

Propositional Logic can't say

- If X is married to Y, then Y is married to X.

Propositional Logic can't say

- If X is married to Y, then Y is married to X.
- If X is west of Y, and Y is west of Z, then X is west of Z.

Propositional Logic can't say

- If X is married to Y, then Y is married to X.
- If X is west of Y, and Y is west of Z, then X is west of Z.
- And a million other simple things.

Propositional Logic can't say (cont.)

- In propositional logic, the best we can do is to say $\phi \wedge \psi \implies \sigma$. We loose the internal structure.

Propositional Logic can't say (cont.)

- In propositional logic, the best we can do is to say $\phi \wedge \psi \implies \sigma$. We loose the internal structure.

Example

Every person likes ice cream. Billy is a person. Therefore, Billy likes ice cream.

Propositional Logic can't say (cont.)

- In propositional logic, the best we can do is to say $\phi \wedge \psi \implies \sigma$. We loose the internal structure.

Example

Every person likes ice cream. Billy is a person. Therefore, Billy likes ice cream.

- We need to be able to refer to objects

Propositional Logic can't say (cont.)

- In propositional logic, the best we can do is to say $\phi \wedge \psi \implies \sigma$. We loose the internal structure.

Example

Every person likes ice cream. Billy is a person. Therefore, Billy likes ice cream.

- We need to be able to refer to objects
- We also need to refer to relations between objects.

Propositional Logic can't say (cont.)

- In propositional logic, the best we can do is to say $\phi \wedge \psi \implies \sigma$. We loose the internal structure.

Example

Every person likes ice cream. Billy is a person. Therefore, Billy likes ice cream.

- We need to be able to refer to objects
- We also need to refer to relations between objects.
- If we can refer to objects, we also want to be able to capture the meaning of every and some of.

Propositional Logic can't say (cont.)

- In propositional logic, the best we can do is to say $\phi \wedge \psi \implies \sigma$. We loose the internal structure.

Example

Every person likes ice cream. Billy is a person. Therefore, Billy likes ice cream.

- We need to be able to refer to objects
- We also need to refer to relations between objects.
- If we can refer to objects, we also want to be able to capture the meaning of every and some of.
- The predicates and quantifiers of predicate logic allow us to capture these concepts.

NL Semantics: Two Basic Issues

- How can we automate the process of associating semantic representations with expressions of natural language?

- How can we use semantic representations of NL expressions to automate the process of drawing inferences?

Associating Semantic Representations Automatically

- Design a semantic representation language

Associating Semantic Representations Automatically

- Design a semantic representation language

- Figure out how to compute the semantic representation of sentences

Associating Semantic Representations Automatically

- Design a semantic representation language
- Figure out how to compute the semantic representation of sentences
- Link this computation to the grammar and lexicon

Semantic Representation Language

- Logical form (LF) is the name used by logicians to talk about the representation of context-independent meaning

Semantic Representation Language

- Logical form (LF) is the name used by logicians to talk about the representation of context-independent meaning

- Semantic representation language has to encode the LF

Semantic Representation Language

- Logical form (LF) is the name used by logicians to talk about the representation of context-independent meaning

- Semantic representation language has to encode the LF

- One concrete representation for LF is First-Order Logic (FOL)

Why is FOL a good thing?

- Flexible, well-understood, and computational tractable
- Produced directly from the syntactic structure of a sentence
- Specify the sentence meaning without having to refer back natural language itself
- Context-independent: does not contain the results of any analysis that requires interpretation of the sentences in context

Facilitate concise representations and semantics for sound reasoning procedures

Anatomy of FOL: variables

Terms: devices to represent objects

Anatomy of FOL: variables

Terms: devices to represent objects

Variables

- make assertions and draw references about objects without having to make reference to any particular named object (anonymous objects)
- depicted as single lower-case letters, e.g. $\mathtt{x} \mid \mathtt{y} \mid \mathtt{z} \mid \ldots$

Anatomy of FOL: variables

Terms: devices to represent objects

Variables

- make assertions and draw references about objects without having to make reference to any particular named object (anonymous objects)
- depicted as single lower-case letters, e.g. $\mathtt{x} \mid \mathtt{y} \mid \mathtt{z} \mid \ldots$

$$g_1[y \mapsto Benny] \begin{bmatrix} x \to Anna \\ y \to Benny \\ z \to Bjorn \\ \ldots \end{bmatrix} \qquad g_2[z \mapsto Benny] \begin{bmatrix} x \to Anna \\ y \to Bjorn \\ z \to Benny \\ \ldots \end{bmatrix}$$

Anatomy of FOL: variables

Terms: devices to represent objects

Variables

- make assertions and draw references about objects without having to make reference to any particular named object (anonymous objects)
- depicted as single lower-case letters, e.g. x | y | z |...

$$g_1[y \mapsto Benny] \begin{bmatrix} x \rightarrow Anna \\ y \rightarrow Benny \\ z \rightarrow Bjorn \\ \dots \end{bmatrix} \qquad g_2[z \mapsto Benny] \begin{bmatrix} x \rightarrow Anna \\ y \rightarrow Bjorn \\ z \rightarrow Benny \\ \dots \end{bmatrix}$$

Semantic Rule: variables

the denotation of the variable x with respect to model M and assignment function g: $[\![x]\!]^{M,g}$

Anatomy of FOL: constants

Constants

- refer to specific objects in the world being described
- depicted as single single letters or single single words, e.g. $a \mid g \mid maharani \mid \ldots$

Anatomy of FOL: individual constants (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

Anatomy of FOL: individual constants (semantics)

Denotation of an expression α relative to model M is $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ determines a domain of individuals D and an interpretation function I

if α is an individual constant then $[\![\alpha]\!]^M = I(\alpha)$

Anatomy of FOL: individual constants (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

if α is an individual constant then $[\![\alpha]\!]^M = I(\alpha)$

Example

Marilyn sings.

$Marilyn \rightsquigarrow m$ (English to logic)

$[\![m]\!]^{M_0} = Marilyn$ (logic to denotation)

$[\![Marilyn]\!] = Marilyn$ (English to denotation *direct interpretation*)

Anatomy of FOL: syntax (cont.)

Terms: Predicates

Anatomy of FOL: syntax (cont.)

Terms: Predicates

- symbols refer to the **relations** holding among some fixed number of objects in a given domain
- or symbols refer to the **properties** of a single object, e.g. encode the category membership
- arguments of a predicate are terms, not other predicates, e.g. `Visit(x,maharani)`

Anatomy of FOL: syntax (cont.)

Terms: Predicates

- symbols refer to the **relations** holding among some fixed number of objects in a given domain
- or symbols refer to the **properties** of a single object, e.g. encode the category membership
- arguments of a predicate are terms, not other predicates, e.g. `Visit(x,maharani)`

Predicate *Visit* APPLIES TO, RELATES *x* and *maharani*

Anatomy of FOL: syntax (cont.)

Terms: Predicates

- symbols refer to the **relations** holding among some fixed number of objects in a given domain
- or symbols refer to the **properties** of a single object, e.g. encode the category membership
- arguments of a predicate are terms, not other predicates, e.g. `Visit(x,maharani)`

Predicate *Visit* APPLIES TO, RELATES x and *maharani*
The number of argument that a predicate takes is its ARITY, VALENCE, ADICITY

Anatomy of FOL: syntax (cont.)

Terms: Predicates

- symbols refer to the **relations** holding among some fixed number of objects in a given domain
- or symbols refer to the **properties** of a single object, e.g. encode the category membership
- arguments of a predicate are terms, not other predicates, e.g. `Visit(x,maharani)`

Predicate *Visit* APPLIES TO, RELATES x and *maharani*
The number of argument that a predicate takes is its ARITY, VALENCE, ADICITY

Example

Marilyn sings.
$Sings(m)$ one-place predicate
Bjorn loves Marilyn.
$Loves(b, m)$ two-place predicate
Bjorn gives Marilyn roses.
$Gives(b, m, r)$ three-place predicate

Anatomy of FOL: predicates (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

Anatomy of FOL: predicates (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

if α is a predicate then $[\![\alpha]\!]^M = I(\alpha)$

Anatomy of FOL: predicates (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

if α is a predicate then $[\![\alpha]\!]^M = I(\alpha)$

Example

Marilyn sings.

Marilyn sings $\rightsquigarrow$ $Sings(m)$ (English to logic)

$[\![Sings(m)]\!]^M = 1$ if $[\![m]\!]^M \in [\![Sings]\!]^M$ (logic to denotation)

Anatomy of FOL: predicates (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

if α is a predicate then $[\![\alpha]\!]^M = I(\alpha)$

Example

Marilyn sings.

Marilyn sings $\rightsquigarrow$ $Sings(m)$ (English to logic)

$[\![Sings(m)]\!]^M = 1$ if $[\![m]\!]^M \in [\![Sings]\!]^M$ (logic to denotation)

$[\![\pi(\alpha)]\!]^M = 1$ if $[\![\alpha]\!]^M \in [\![\pi]\!]^M$, and 0 otherwise (general denotation of unary predicate)

Anatomy of FOL: predicates (semantics)

Denotation of an expression α *relative to model* M *is* $[\![\alpha]\!]^M$
A model $M = \langle D, I \rangle$ *determines a domain of individuals* D *and an interpretation function* I

if α is a predicate then $[\![\alpha]\!]^M = I(\alpha)$

Example

Marilyn sings.

Marilyn sings $\rightsquigarrow$ $Sings(m)$ (English to logic)

$[\![Sings(m)]\!]^M = 1$ if $[\![m]\!]^M \in [\![Sings]\!]^M$ (logic to denotation)

$[\![\pi(\alpha)]\!]^M = 1$ if $[\![\alpha]\!]^M \in [\![\pi]\!]^M$, and 0 otherwise (general denotation of unary predicate)

Semantic Rule: Predication

$[\![\pi(\alpha_1, \dots, \alpha_n)]\!]^M = 1$ if $\langle [\![\alpha_1]\!]^M, \dots, [\![\alpha_n]\!]^M \in [\![\pi]\!]^M \rangle$, and 0 otherwise

Anatomy of FOL: syntax (cont.)

Terms: Functions

- refer to unique objects without having to associate a name constant with them
- syntactically the same as single predicates, e.g. `lecturerOf` | `ownerOf` | ...

Anatomy of FOL: syntax (cont.)

Terms: Functions

- refer to unique objects without having to associate a name constant with them
- syntactically the same as single predicates, e.g. `lecturerOf` | `ownerOf` | ...

Example

Marilyn's husband $\rightsquigarrow$ $\mathit{spouseOf}(m)$ — unary function

Marilyn loves her husband. $\rightsquigarrow$ $\mathit{Loves}(m, \mathit{spouseOf}(m))$

Bjorn is Marilyn's husband. $\rightsquigarrow$ $\mathit{spouseOf}(b, m) \land \mathit{spouseOf}(m, b)$ — binary functions

Anatomy of FOL: syntax (cont.)

Terms: Functions

- refer to unique objects without having to associate a name constant with them
- syntactically the same as single predicates, e.g. `lecturerOf` | `ownerOf` | ...

Example

Marilyn's husband $\rightsquigarrow$ $spouseOf(m)$ unary function

Marilyn loves her husband. $\rightsquigarrow$ $Loves(m, spouseOf(m))$

Bjorn is Marilyn's husband. $\rightsquigarrow$ $spouseOf(b, m) \wedge spouseOf(m, b)$ binary functions

Syntactic Rule: Complex Terms

given any function γ with arity n:

$\gamma(\alpha_1, ..., \alpha_n)$

is a term, where $(\alpha_1, ..., \alpha_n)$ is a sequence of expressions that are themselves terms.

Anatomy of FOL: functions (semantics)

Semantic Rule: Binary Function

$$[\![\gamma(\alpha, \beta)]\!]^M = [\![\gamma]\!]^M(\langle [\![\alpha]\!]^M, [\![\beta]\!]^M \rangle)$$

Anatomy of FOL: functions (semantics)

Semantic Rule: Binary Function

$$[\![\gamma(\alpha, \beta)]\!]^M = [\![\gamma]\!]^M(\langle [\![\alpha]\!]^M, [\![\beta]\!]^M \rangle)$$

Semantic Rule: Complex Terms

$$[\![\gamma(\alpha_1, ..., \alpha_n)]\!]^M = [\![\gamma]\!]^M(\langle [\![\alpha_1]\!]^M, ..., [\![\alpha_n]\!]^M \rangle)$$

Anatomy of FOL

Logical Connectives:

- $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\implies$ (imply) operators
- 16 possible truth functional binary values

P	Q	$\neg$ P	P $\wedge$ Q	P $\vee$ Q	P $\implies$ Q
False	False	True	False	False	True
False	True	True	False	True	True
True	False	False	False	True	False
True	True	False	True	True	True

- used to form larger composite representations

Example

I have five dollars and I don't have a lot of time

Have(speaker, fiveDollars) $\wedge$ $\neg$ Have(speaker, lotOfTime)

Anatomy of FOL: LF of Phrases

Word	**POS**	**Logic**	**Representation**
Maharani	proper name	individual constant	maharani
restaurant	common noun	1-place predicate	Restaurant(x)
delicious	adjective	1-place predicate	Delicious(x)
delicious vegetarian food	adj/noun	1-place predicate	Delicious(x) $\wedge$ Vegetarian(x) $\wedge$ Food(x)
snore	intransitive verb	1-place predicate	Snores(x)
study	transitive verb	2-place predicate	Study(x,y)
give	ditransitive verb	3-place predicate	Give(x,y,z)

Anatomy of FOL (cont.)

Quantifiers:

Anatomy of FOL (cont.)

Quantifiers:

- The existential quantifier $\exists$: pronounced as "there exists"

Anatomy of FOL (cont.)

Quantifiers:

- The existential quantifier $\exists$: pronounced as "there exists"

Example

A restaurant that serves Mexican food is near UdS.
$\exists$x [Restaurant(x) $\wedge$ Serve(x,mexicanFood) $\wedge$ Near(locationOf(x),locationOf(uds)]

Anatomy of FOL (cont.)

Quantifiers:

- The existential quantifier $\exists$: pronounced as "there exists"

Example

A restaurant that serves Mexican food is near UdS.
$\exists x$ [Restaurant(x) $\wedge$ Serve(x,mexicanFood) $\wedge$ Near(locationOf(x),locationOf(uds)]

- The universal quantifier $\forall$: pronounced as "for all"

Anatomy of FOL (cont.)

Quantifiers:

- The existential quantifier $\exists$: pronounced as “there exists”

Example

A restaurant that serves Mexican food is near UdS.

$\exists$x [Restaurant(x) $\wedge$ Serve(x,mexicanFood) $\wedge$ Near(locationOf(x),locationOf(uds)]

- The universal quantifier $\forall$: pronounced as “for all”

Example

All vegetarian restaurant serve vegetarian food.

$\forall$x [VegetarianRestaurant(x) $\implies$ Serve(x,vegetarianFood)]

Anatomy of FOL: quantification (syntax)

Syntactic Rule: Complex Terms

given any variable u, if ϕ is a formula then:

Anatomy of FOL: quantification (syntax)

Syntactic Rule: Complex Terms

given any variable u, if ϕ is a formula then:

$$[\forall u.\phi]$$

Anatomy of FOL: quantification (syntax)

Syntactic Rule: Complex Terms

given any variable u, if ϕ is a formula then:

$$[\forall u.\phi]$$

is a formula,

Anatomy of FOL: quantification (syntax)

Syntactic Rule: Complex Terms

given any variable u, if ϕ is a formula then:

$$[\forall u.\phi]$$

is a formula,and so is

$$[\exists u.\phi]$$

Anatomy of FOL: quantification (semantics)

RECALL: Semantic Rule: variables

the denotation of the variable x with respect to model M and assignment function g:
$[\![x]\!]^{M,g}$

Anatomy of FOL: quantification (semantics)

RECALL: Semantic Rule: variables

the denotation of the variable x with respect to model M and assignment function g:
$[\![x]\!]^{M,g}$

Semantic Rule: Existential quantification

$[\![\exists x.\phi]\!]^{M,g} = 1$ iff there an individual $k \in D$ such that: $[\![\phi]\!]^{M,g[x \mapsto k]} = 1$

Anatomy of FOL: quantification (semantics)

RECALL: Semantic Rule: variables

the denotation of the variable x with respect to model M and assignment function g:
$[\![x]\!]^{M,g}$

Semantic Rule: Existential quantification

$[\![\exists x.\phi]\!]^{M,g} = 1$ iff there an individual $k \in D$ such that: $[\![\phi]\!]^{M,g[x \mapsto k]} = 1$

Semantic Rule: Universal quantification

$[\![\forall v.\phi]\!]^{M,g} = 1$ iff for all individuals $k \in D$ such that: $[\![\phi]\!]^{M,g[v \mapsto k]} = 1$

Anatomy of FOL: LF of Sentences

Example

John kicks Fido.

Kick(john,fido)

Example

Every student read a book.

$\forall x\ [Student(x) \implies \exists y\ [Book(y) \wedge Read(x,y)]]$

$\exists y\ [Book(y) \wedge \forall x\ [Student(x) \implies Read(x,y)]]$

Semantic ambiguity related to quantifier scope

Predicate Logic: Syntax

The syntax of predicate logic consists of:

- constants
- variables x, y, ...
- functions f(), g(), ...
- predicates P(), Q(), ...
- logical connectives $\wedge$, $\vee$, $\neg$, $\implies$
- quantifiers $\exists$ $\forall$
- punctuations: , . ( ) []

Predicate Logic: Syntax

Definition: Terms are defined inductively as follows:

Base cases

- Every constant is a term.
- Every variable is a term.

Inductive cases

- If t_1,t_2,t_3,..., t_n are terms then f(t_1,t_2,t_3,..., t_n) is a term, where f is an n-ary function.
- Nothing else is a term.

Predicate Logic: Syntax (well-formed formulas)

Definition: well-formed formulas (wffs) are defined inductively as follows: Base cases

- $P(t_1, t_2, t_3, \ldots, t_n)$ is a wff, where t_i is a term, and P is an n-place predicate. These are called atomic formulas.

Inductive cases

- If A and B are wffs, then so are $\neg A$, $A \wedge B$, $A \vee B$, $A \implies B$
- If A is a wff, so is $\exists x.\ A$
- If A is a wff, so is $\forall x.\ A$
- Nothing else is a wff.

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

$\forall$x. Happy(x)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

$\forall$x. Happy(x)

$\forall$x. Happy(y)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

$\forall$x. Happy(x)

$\forall$x. Happy(y)

$\exists$x. Loves(x,x)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

$\forall$x. Happy(x)

$\forall$x. Happy(y)

$\exists$x. Loves(x,x)

$\exists$x. Loves(x,z)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

$\forall$x. Happy(x)

$\forall$x. Happy(y)

$\exists$x. Loves(x,x)

$\exists$x. Loves(x,z)

$\exists$x. $\exists$ y. Loves(x,y)

Predicate Logic: Syntax (well-formed formulas)

Which of below are well-formed formulas of L_1

[Happy(m) $\wedge$ Happy(m)]

Happy(k)

Happy(m,m)

$\neg\neg$ Happy(n)

$\forall$x. Happy(x)

$\forall$x. Happy(y)

$\exists$x. Loves(x,x)

$\exists$x. Loves(x,z)

$\exists$x. $\exists$ y. Loves(x,y)

$\exists$x. Happy(m)

Predicate Logic: Scope and Binding Variables

Translation of sentences with more than one quantifier

Predicate Logic: Scope and Binding Variables

Translation of sentences with more than one quantifier

Consider

(a) Everyone loves someone.
(b) Someone is loved by everyone.

Predicate Logic: Scope and Binding Variables

Translation of sentences with more than one quantifier

Consider

(a) Everyone loves someone.
(b) Someone is loved by everyone.

What are the truth-conditions for (a) and (b)?

Predicate Logic: Scope and Binding Variables

Translation of sentences with more than one quantifier

Consider

(a) Everyone loves someone.
(b) Someone is loved by everyone.

What are the truth-conditions for (a) and (b)?

(a) is true in a situation in which everyone has fallen in love with a person whoever they are, e.g. Leonard and Penny loves each other, Sheldon and Amy loves each other, etc.
$\forall x \exists y$ Loves(x,y)

Predicate Logic: Scope and Binding Variables

Translation of sentences with more than one quantifier

Consider

(a) Everyone loves someone.
(b) Someone is loved by everyone.

What are the truth-conditions for (a) and (b)?

(a) is true in a situation in which everyone has fallen in love with a person whoever they are, e.g. Leonard and Penny loves each other, Sheldon and Amy loves each other, etc.
$\forall x \exists y$ Loves(x,y)
(b) is true in a situation in which there is one single person and everyone loves that one person, e.g. Ted loves Robyn, Barney loves Robin, etc.
$\exists y \forall x$ Loves(x,y)

Predicate Logic: Scope and Binding Variables

(a) $\forall x \exists y$ Loves(x,y)
(b) $\exists y \forall x$ Loves(x,y)

For every variable (x and y) in (a) and (b) there is a corresponding quantifier

Thus, x and y are bound variables.

Quizz for Today

TBA in class