Ontology Engineering

Lecture 2: First Order Logic

Maria Keet

email: mkeet@cs.uct.ac.za

home: http://www.meteck.org

Department of Computer Science
University of Cape Town, South Africa

Semester 2, Block I, 2019

Outline

Note on 'Block I' of OE (logics)

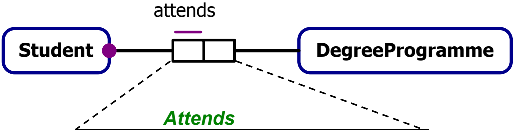
- There are only a few core concepts to get the general idea
- There are very many details
- Here we focus on the core concepts and some details and how that works out in computing
- More logic and details in the 'Logics for AI' course

Outline

Example data, model, and NL–how to formalise it?

Student **is an entity type**.
DegreeProgramme **is an entity type**.
Student attends DegreeProgramme.

Each Student attends **exactly one** DegreeProgramme.
It is possible that more than one Student attends **the same** DegreeProgramme.
OR, in the negative:
For each Student, **it is impossible that that** Student attends **more than one** DegreeProgramme.
It is impossible that any Student attends **no** DegreeProgramme.



Attends

Student	DegreeProgramme
John	Computer Science
Mary	Design
Fabio	Design
Claudio	Computer Science
Markus	Biology
Inge	Computer Science

Beginnings

- Truth values 1 or 0 (or something else with many-valued logics)
- True or false?
  - A = "Aristotle is alive"
  - B = "Cape Town is located in South Africa"
  - C = "Praise Allah"

Beginnings

- Truth values 1 or 0 (or something else with many-valued logics)
- True or false?
  - A = "Aristotle is alive"
  - B = "Cape Town is located in South Africa"
  - C = "Praise Allah"
- Realise that logic is not the study of truth, but of the relationship between the truth of one statement and that of another

Some definitions

- A formula is valid if it holds under *every* assignment. $\models F$ to denote this. A valid formula is called a tautology.
- A formula is satisfiable if it holds under *some* assignment.
- A formula is unsatisfiable if it holds under *no* assignment. An unsatisafiable formula is called a contradiction.

Tibbles

- Is the following argument valid?
    - If Tibbles roves the Upper Campus, then he lives in Rondebosch.
    - Tibbles lives in Rondebosch.
    - Therefore Tibbles roves the Upper Campus.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - **If** Tibbles roves the Upper Campus, **then** he lives in Rondebosch.
  - Tibbles lives in Rondebosch.
  - **Therefore** Tibbles roves the Upper Campus.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - **If** Tibbles roves the Upper Campus, **then** he lives in Rondebosch.
  - Tibbles lives in Rondebosch.
  - **Therefore** Tibbles roves the Upper Campus.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - **If** Tibbles roves the Upper Campus, **then** he [Tibbles] lives in Rondebosch.
  - Tibbles lives in Rondebosch.
  - **Therefore** Tibbles roves the Upper Campus.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - **If** A, **then** B.
  - B.
  - **Therefore** A.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - $A \to B$
  - B.
  - **Therefore** A.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - $A \rightarrow B$
  - $\wedge$ B
  - **Therefore** A.
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - $A \to B$
  - $\wedge$ B
  - $\to$ A
- Represent the argument formally and use truth tables to prove it.

Tibbles

- Is the following argument valid?
  - $A \rightarrow B$
  - $\wedge$ B
  - $\rightarrow$ A
- Represent the argument formally and use truth tables to prove it.
- So, $((A \rightarrow B) \wedge B) \rightarrow A$

Implication and talking about it in English

A	**B**	**A** $\rightarrow$ **B**	**can read it as**	
		$\neg$**A** $\vee$ **B**	If A then B	B follows from A
0	0	1	A implies B	A is sufficient for B
0	1	1	A only if B	B is necessary for A
1	0	0	B if A	B is a necessary condition for A
1	1	1	Whenever A, B	B whenever A
			Not A unless B	A is a sufficient condition for B

How to formalise it?

- Syntax
  - Alphabet
  - Languages constructs
  - Sentences to assert knowledge
- Semantics
  - Formal meaning

Outline

First order logic

The lexicon of a first order language contains:

- Connectives & Parentheses: $\neg$, $\rightarrow$, $\leftrightarrow$, $\wedge$, $\vee$, (and);
- Quantifiers: $\forall$ (universal) and $\exists$ (existential);
- Variables: $x, y, z, \ldots$ ranging over particulars;
- Constants: $a, b, c, \ldots$ representing a specific element;
- Functions: $f, g, h, \ldots$, with arguments listed as $f(x_1, \ldots x_n)$;
- Relations: $R, S, \ldots$ with an associated arity.

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$
- Each student attends at least one degree programme

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$
- Each student attends at least one degree programme
  $\forall x, y(attends(x, y) \rightarrow Student(x) \wedge DegreeProg(y))$
  $\forall x(Student(x) \rightarrow \exists y\ attends(x, y))$

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$
- Each student attends at least one degree programme
  $\forall x, y(attends(x, y) \rightarrow Student(x) \wedge DegreeProg(y))$
  $\forall x(Student(x) \rightarrow \exists y \; attends(x, y))$
- Aliens exist

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$
- Each student attends at least one degree programme
  $\forall x, y(attends(x, y) \rightarrow Student(x) \wedge DegreeProg(y))$
  $\forall x(Student(x) \rightarrow \exists y\ attends(x, y))$
- Aliens exist
  $\exists x\ Alien(x)$

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$
- Each student attends at least one degree programme
  $\forall x, y(attends(x, y) \rightarrow Student(x) \wedge DegreeProg(y))$
  $\forall x(Student(x) \rightarrow \exists y \; attends(x, y))$
- Aliens exist
  $\exists x \; Alien(x)$
- There are books that are heavy

Example: Natural Language and First Order Logic

- Each animal is an organism
  All animals are organisms
  If it is an animal then it is an organism
  $\forall x(Animal(x) \rightarrow Organism(x))$
- Each student attends at least one degree programme
  $\forall x, y(attends(x, y) \rightarrow Student(x) \wedge DegreeProg(y))$
  $\forall x(Student(x) \rightarrow \exists y \; attends(x, y))$
- Aliens exist
  $\exists x \; Alien(x)$
- There are books that are heavy
  $\exists x(Book(x) \wedge heavy(x))$

First order logic

- (in logics) "A theory is a consistent set of sentences"
  - what does that mean?

First order logic

- (in logics) "A theory is a consistent set of sentences"
  - what does that mean?
- (countably infinite) Supply of **symbols** (signature): Variables, Functions , Constants, and Relations
- **Terms**: A term is inductively defined by two rules, being:
  1. Every variable and constant is a term.
  2. if f is a m-ary function and $t_1, \ldots t_m$ are terms, then $f(t_1, \ldots, t_m)$ is also a term.

First order logic

- (in logics) "A theory is a consistent set of sentences"
  - what does that mean?
- (countably infinite) Supply of **symbols** (signature): Variables, Functions , Constants, and Relations
- **Terms**: A term is inductively defined by two rules, being:
  1. Every variable and constant is a term.
  2. if f is a m-ary function and $t_1, \ldots t_m$ are terms, then $f(t_1, \ldots, t_m)$ is also a term.

Definition (atomic formula)

An *atomic formula* is a formula that has the form $t_1 = t_2$ or $R(t_1, ..., t_n)$ where R is an n-ary relation and $t_1, ..., t_n$ are terms.

R1. If ϕ is a formula then so is $\neg\phi$.

R2. If ϕ and ψ are formulas then so is $\phi \wedge \psi$.

R3. If ϕ is a formula then so is $\exists x \phi$ for any variable x.

FOL Cont. (informally)

formula: constructed from atomic formulas by repeated applications of rules R1, R2, and R3

free variable that variable in a formula that is not quantified ('bound' with an $\exists$ or a $\forall$)

sentence a formula that has no free variables

Outline

FOL Cont.: toward semantics

- Whether a sentence is true or not depends on the underlying set and the interpretation of the function, constant, and relation symbols.
- A *structure* consists of an *underlying set* together with an *interpretation* of functions, constants, and relations.
- Given a sentence ϕ and a structure M, M **models** ϕ means that the sentence ϕ is true with respect to M. More precisely,

FOL Cont.: toward semantics

- Whether a sentence is true or not depends on the underlying set and the interpretation of the function, constant, and relation symbols.
- A *structure* consists of an *underlying set* together with an *interpretation* of functions, constants, and relations.
- Given a sentence ϕ and a structure M, M **models** ϕ means that the sentence ϕ is true with respect to M. More precisely,

Definition (vocabulary)

A *vocabulary* $\mathcal{V}$ is a set of function, relation, and constant symbols.

FOL Cont.

Definition ($\mathcal{V}$-structure)

A $\mathcal{V}$-*structure* consists of a non-empty underlying set Δ along with an interpretation of $\mathcal{V}$. An interpretation of $\mathcal{V}$ assigns an element of Δ to each constant in $\mathcal{V}$, a function from Δ^n to Δ to each n-ary function in $\mathcal{V}$, and a subset of Δ^n to each n-ary relation in $\mathcal{V}$. We say M is a *structure* if it is a $\mathcal{V}$-structure of some vocabulary $\mathcal{V}$.

FOL Cont.

Definition ($\mathcal{V}$-structure)

A $\mathcal{V}$-*structure* consists of a non-empty underlying set Δ along with an interpretation of $\mathcal{V}$. An interpretation of $\mathcal{V}$ assigns an element of Δ to each constant in $\mathcal{V}$, a function from Δ^n to Δ to each n-ary function in $\mathcal{V}$, and a subset of Δ^n to each n-ary relation in $\mathcal{V}$. We say M is a *structure* if it is a $\mathcal{V}$-structure of some vocabulary $\mathcal{V}$.

Definition ($\mathcal{V}$-formula)

Let $\mathcal{V}$ be a vocabulary. A $\mathcal{V}$-*formula* is a formula in which every function, relation, and constant is in $\mathcal{V}$. A $\mathcal{V}$-*sentence* is a $\mathcal{V}$-formula that is a sentence.

FOL Cont.

- When we say that M *models* ϕ, denoted with $M \models \phi$, this is with respect to M being a $\mathcal{V}$-structure and $\mathcal{V}$-sentence ϕ is true in M.
- Model theory: the interplay between M and a set of first-order sentences $\mathcal{T}(M)$, which is called the *theory of M*, and its 'inverse' from a set of sentences Γ to a class of structures.

FOL Cont.

- When we say that M *models* ϕ, denoted with $M \models \phi$, this is with respect to M being a $\mathcal{V}$-structure and $\mathcal{V}$-sentence ϕ is true in M.
- Model theory: the interplay between M and a set of first-order sentences $\mathcal{T}(M)$, which is called the *theory of M*, and its 'inverse' from a set of sentences Γ to a class of structures.

Definition (theory of M)

For any $\mathcal{V}$-structure M, the *theory of M*, denoted with $\mathcal{T}(M)$, is the set of all $\mathcal{V}$-sentences ϕ such that $M \models \phi$.

FOL Cont.

- When we say that M *models* ϕ, denoted with $M \models \phi$, this is with respect to M being a $\mathcal{V}$-structure and $\mathcal{V}$-sentence ϕ is true in M.
- Model theory: the interplay between M and a set of first-order sentences $\mathcal{T}(M)$, which is called the *theory of M*, and its 'inverse' from a set of sentences Γ to a class of structures.

Definition (theory of M)

For any $\mathcal{V}$-structure M, the *theory of M*, denoted with $\mathcal{T}(M)$, is the set of all $\mathcal{V}$-sentences ϕ such that $M \models \phi$.

Definition (model)

For any set of $\mathcal{V}$-sentences, a *model* of Γ is a $\mathcal{V}$-structure that models each sentence in Γ. The class of all models of Γ is denoted by $\mathcal{M}(\Gamma)$.

Theory in the context of logic

Definition (complete $\mathcal{V}$-theory)

Let Γ be a set of $\mathcal{V}$-sentences. Then Γ is a *complete $\mathcal{V}$-theory* if, for any $\mathcal{V}$-sentence ϕ either ϕ or $\neg\phi$ is in Γ and it is not the case that both ϕ and $\neg\phi$ are in Γ.

- It can then be shown that for any $\mathcal{V}$-structure M, $\mathcal{T}(M)$ is a complete $\mathcal{V}$-theory (for proof, see *e.g.* [Hedman04, p90])

Theory in the context of logic

Definition (complete $\mathcal{V}$-theory)

Let Γ be a set of $\mathcal{V}$-sentences. Then Γ is a *complete $\mathcal{V}$-theory* if, for any $\mathcal{V}$-sentence ϕ either ϕ or $\neg\phi$ is in Γ and it is not the case that both ϕ and $\neg\phi$ are in Γ.

- It can then be shown that for any $\mathcal{V}$-structure M, $\mathcal{T}(M)$ is a complete $\mathcal{V}$-theory (for proof, see *e.g.* [Hedman04, p90])

Definition

A set of sentences Γ is said to be *consistent* if no contradiction can be derived from Γ.

Theory in the context of logic

Definition (complete $\mathcal{V}$-theory)

Let Γ be a set of $\mathcal{V}$-sentences. Then Γ is a *complete $\mathcal{V}$-theory* if, for any $\mathcal{V}$-sentence ϕ either ϕ or $\neg\phi$ is in Γ and it is not the case that both ϕ and $\neg\phi$ are in Γ.

- It can then be shown that for any $\mathcal{V}$-structure M, $\mathcal{T}(M)$ is a complete $\mathcal{V}$-theory (for proof, see *e.g.* [Hedman04, p90])

Definition

A set of sentences Γ is said to be *consistent* if no contradiction can be derived from Γ.

Definition (theory)

A *theory* is a consistent set of sentences.

Some definitions

- A formula is valid if it holds under *every* assignment. $\models \phi$ to denote this. A valid formula is called a tautology.
- A formula is satisfiable if it holds under *some* assignment.
- A formula is unsatisfiable if it holds under *no* assignment. An unsatisafiable formula is called a contradiction.

Example

- Is this a theory?

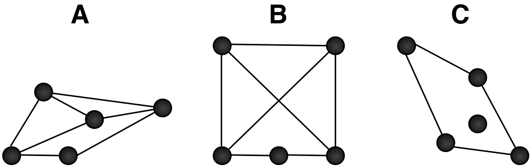
$\forall x(Woman(x) \rightarrow Female(x))$
$\forall x(Mother(x) \rightarrow Woman(x))$
$\forall x(Man(x) \leftrightarrow \neg Woman(x))$
$\forall x(Mother(x) \rightarrow \exists y(partnerOf(x,y) \wedge Spouse(y))$
$\forall x(Spouse(x) \rightarrow Man(x) \vee Woman(x))$
$\forall x,y(Mother(x) \wedge partnerOf(x,y) \rightarrow Father(y))$

Example

- Is this a theory?
  $\forall x(Woman(x) \rightarrow Female(x))$
  $\forall x(Mother(x) \rightarrow Woman(x))$
  $\forall x(Man(x) \leftrightarrow \neg Woman(x))$
  $\forall x(Mother(x) \rightarrow \exists y(partnerOf(x, y) \wedge Spouse(y))$
  $\forall x(Spouse(x) \rightarrow Man(x) \vee Woman(x))$
  $\forall x, y(Mother(x) \wedge partnerOf(x, y) \rightarrow Father(y))$
- Is it still a theory if we add:
  $\forall x(Hermaphrodite(x) \rightarrow Man(x) \wedge Woman(x))$

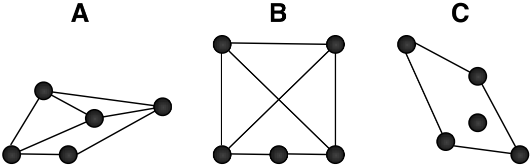
Examples of first-order structures (exercise)

- Graphs are mathematical structures.
- A graph is a set of points, called **vertices**, and lines, called **edges** between them. For instance:



Examples of first-order structures (exercise)

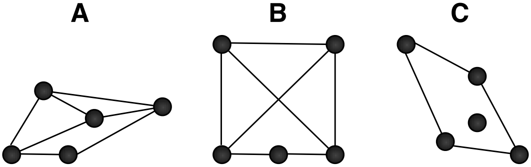
- Graphs are mathematical structures.
- A graph is a set of points, called **vertices**, and lines, called **edges** between them. For instance:



- Figures A and B are different depictions, but have the same descriptions w.r.t. the vertices and edges. Check this.
- Graph C has a property that A and B do not have. Represent this in a first-order sentence.

Examples of first-order structures (exercise)

- Graphs are mathematical structures.
- A graph is a set of points, called **vertices**, and lines, called **edges** between them. For instance:



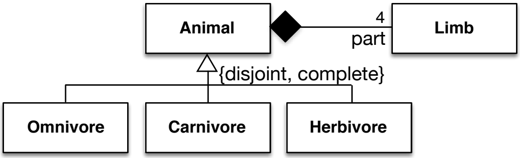
- Figures A and B are different depictions, but have the same descriptions w.r.t. the vertices and edges. Check this.
- Graph C has a property that A and B do not have. Represent this in a first-order sentence.
- Find a suitable first-order language for A (/B), and formulate at least two properties of the graph using quantifiers.

Examples of first-order structures (exercise)

- That example in the introduction of the slides (students attending a degree programme)
- Formalise the type-level information of that ORM diagram

Examples of first-order structures (exercise)

- That example in the introduction of the slides (students attending a degree programme)
- Formalise the type-level information of that ORM diagram
- Then try to formalise the following UML diagram



Outline

Reasoning

- Representing the knowledge in a suitable logic is one thing, reasoning over it another. e.g.:
  - *How* do we find out whether a formula is valid or not?
  - *How* do we find out whether our knowledge base is satisfiable?

Reasoning

- Representing the knowledge in a suitable logic is one thing, reasoning over it another. e.g.:
  - *How* do we find out whether a formula is valid or not?
  - *How* do we find out whether our knowledge base is satisfiable?
- We need some way to do this automatically

Essential to realising automated reasoning

- The choice of the class of problems the software program has to solve: *what is it supposed to solve?*
  - e.g., checking satisfiability of the theory
- The language in which to represent the problems;
  - e.g.: first order predicate logic
- How the program has to compute the solution;
  - e.g., deduction
- How to do this efficiently
  - e.g., constrain the language

Deduction, abduction, induction

- Deduction: ascertain if $T \models \alpha$, where α is not explicitly asserted in T, i.e., whether α can be *derived* from the premises through repeated application of *deduction rules*.

Deduction, abduction, induction

- Deduction: ascertain if $T \models \alpha$, where α is not explicitly asserted in T, i.e., whether α can be *derived* from the premises through repeated application of *deduction rules*.
- Abduction: try to infer *a* as an explanation of *b*. set of observations + a theory of the domain of the observations + a set of (possible, hypothesised) explanations that one would hope to find.

Deduction, abduction, induction

- Deduction: ascertain if $T \models \alpha$, where α is not explicitly asserted in T, i.e., whether α can be *derived* from the premises through repeated application of *deduction rules*.
- Abduction: try to infer *a* as an explanation of *b*. set of observations + a theory of the domain of the observations + a set of (possible, hypothesised) explanations that one would hope to find.
- Induction: generalise to a conclusion based on a set of individuals. The conclusion is *not* a logical consequence of the premise, but premises provide a *degree of support* so as to infer *a* as an explanation of *b*

Reasoning over ontologies

- Most popular for ontologies: deductive
- Some work on other approaches, e.g., e.g., belief revision, probabilistic abductive reasoning, and Bayesian networks for abductive reasoning, ML for inductive reasoning

Reasoning over ontologies - techniques

- NOT truth tables (doesn't scale, at all)
- Many options, e.g.:
  - Case-based reasoning
  - Automata-based techniques
  - Tableaux (current 'winner')
- Many variants with many optimisations, for many logics

Tableaux

- A sound and complete procedure deciding satisfiability is all we need, and the **tableaux method is a decision procedure which checks the existence of a model**
- It exhaustively looks at all the possibilities, so that it can eventually prove that no model could be found for unsatisfiable formulas.

Tableaux

- A sound and complete procedure deciding satisfiability is all we need, and the **tableaux method is a decision procedure which checks the existence of a model**
- It exhaustively looks at all the possibilities, so that it can eventually prove that no model could be found for unsatisfiable formulas.
- $\phi \models \psi$ iff $\phi \wedge \neg\psi$ is NOT satisfiable—if it is satisfiable, we have found a counterexample

Tableaux

- A sound and complete procedure deciding satisfiability is all we need, and the **tableaux method is a decision procedure which checks the existence of a model**
- It exhaustively looks at all the possibilities, so that it can eventually prove that no model could be found for unsatisfiable formulas.
- $\phi \models \psi$ iff $\phi \land \neg\psi$ is NOT satisfiable—if it is satisfiable, we have found a counterexample
- Decompose the formula in top-down fashion

Tableaux

- Tableaux calculus works only if the formula has been translated into **Negation Normal Form**, *i.e.*, all the negations have been pushed inside
- Recall the list of equivalences, apply those to arrive at NNF, if necessary. (pp32-33 of the book)
- If a model satisfies a **conjunction**, then it also satisfies each of the conjuncts:

$$\frac{\phi \wedge \psi}{\begin{array}{c}\phi \\ \psi\end{array}}$$

- If a model satisfies a **disjunction**, then it also satisfies one of the disjuncts. non-deterministic

$$\frac{\phi \vee \psi}{\phi \mid \psi}$$

Tableaux

- If a model satisfies a **universally quantified formula** ($\forall$), then it also satisfies the formula where the quantified variable has been substituted with a ground term (constant or function)

  $$\frac{\forall x.\phi}{\phi\{x/t\}}$$
  $\forall x.\phi$

- For an **existentially quantified formula**, if a model satisfies it, then it also satisfies the formula where the quantified variable has been substituted with a new Skolem constant,

  $$\frac{\exists x.\phi}{\phi\{x/a\}}$$
  Note: this is a *'brand new'* constant in the theory

Tableaux

- Apply the completion rules until either
  (a) an explicit contradiction due to the presence of two opposite literals in a node (a clash) is generated in each branch, or
  (b) there is a completed branch where no more rule is applicable

Example

- Input to the tableau:
  1. $\forall x \neg P(x, a)$
  2. $P(a, b)$
  3. $\forall x, y(\neg P(x, y) \vee P(y, x))$
- Apply one of the rules. which one?

Example

- Input to the tableau:
  1 $\forall x \neg P(x, a)$
  2 $P(a, b)$
  3 $\forall x, y(\neg P(x, y) \vee P(y, x))$
- Apply one of the rules. which one?
- The first axiom is the only ‘reasonable’ option:
  4 $\neg P(b, a)$ (substitute x with b)

Example

- Input to the tableau:
  1 $\forall x \neg P(x, a)$
  2 $P(a, b)$
  3 $\forall x, y(\neg P(x, y) \vee P(y, x))$
- Apply one of the rules. which one?
- The first axiom is the only 'reasonable' option:
  4 $\neg P(b, a)$ (substitute x with b)
- Now let's 'get rid of' the other $\forall$'s from line 3:
  5 $\forall y(\neg P(a, y) \vee P(y, a))$ (substitute x with a)
  6 $(\neg P(a, b) \vee P(b, a))$ (then y with b)

Example

- Input to the tableau:
  1 $\forall x \neg P(x, a)$
  2 $P(a, b)$
  3 $\forall x, y(\neg P(x, y) \vee P(y, x))$
- Apply one of the rules. which one?
- The first axiom is the only 'reasonable' option:
  4 $\neg P(b, a)$ (substitute x with b)
- Now let's 'get rid of' the other $\forall$'s from line 3:
  5 $\forall y(\neg P(a, y) \vee P(y, a))$ (substitute x with a)
  6 $(\neg P(a, b) \vee P(b, a))$ (then y with b)
- Process the disjunction, generating two branches:
  7a $\neg P(a, b)$ clash! (with line 2)
  7b $P(b, a)$ clash! (with line 4)

Example

- Theory T consists of:
  - R is reflexive: $\forall x(R(x,x))$
  - R is asymmetric: $\forall x, y(R(x,y) \rightarrow \neg R(y,x))$
- Now what if we add $\neg\forall x, y(R(x,y))$ to T?
- Any equivalences and NNF?
  - $\forall x, y(R(x,y) \rightarrow \neg R(y,x))$ rewritten as $\forall x, y(\neg R(x,y) \vee \neg R(y,x))$
- add the negation of $\neg\forall x, y(R(x,y))$ to T, i.e., $\forall x, y(R(x,y))$

Tableau for the example

Number	Tableau	Explanation
1	$\forall x.R(x,x)$	Reflexivity axiom in the original theory T
2	$\forall x,y.\ \neg R(x,y) \vee \neg R(y,x)$	Asymmetry axiom in the original theory T
3	$\forall x,y.R(x,y)$	The negated axiom added to theory T
4		Substitute *x* for term *a* in 1,2,3
5	$R(a,a)$	
6	$\forall y.\ \neg R(a,y) \vee \neg R(y,a)$	
7	$\forall y.R(a,y)$	
8		Substitute *y* for term *a* in 2 and 3
9	$R(a,a)$	
10	$\neg R(a,a) \vee \neg R(a,a)$	
11	$R(a,a)$	
12		Split the disjunction of 10
13	$\neg R(a,a)$ $\neg R(a,a)$	Which each generate a clash with 9 and 11, hence, $\neg\forall x,y.R(x,y)$ is entailed by T.

Relevance?

- DLs are fragments of FOL (next lecture)
- Most reasoning algorithms for DL use this sort of tableau reasoning as well (optimised)
- OWL ontology languages based on DLs: this is roughy what happens when you press the start/synchronise reasoner in Protégé that uses HerMiT for reasoner
- We'll see more examples and exercises later

Summary

1. Introduction
2. Syntax
3. Semantics
   - Some definitions
   - First Order Structures
4. Reasoning
   - General idea
   - Tableaux