Outline

Introduction

Course Organisation

1. All course material and news will be available on my home page `http://www.voronkov.com`
2. The tool (Vampire) is available at `http://www.vprover.org`

First-Order Logic: Exercises

Which of the following statements are true?

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.
4. First-order logic is decidable.

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.
4. First-order logic is decidable.
5. First-order logic is an extension of arithmetic;

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.
4. First-order logic is decidable.
5. First-order logic is an extension of arithmetic;
6. One can axiomatise integers in first-order logic;

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.
4. First-order logic is decidable.
5. First-order logic is an extension of arithmetic;
6. One can axiomatise integers in first-order logic;
7. Compactness is the following property: a set of formulas having arbitrarily large finite models has an infinite model;

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.
4. First-order logic is decidable.
5. First-order logic is an extension of arithmetic;
6. One can axiomatise integers in first-order logic;
7. Compactness is the following property: a set of formulas having arbitrarily large finite models has an infinite model;
8. Having proofs is good.

First-Order Logic: Exercises

Which of the following statements are true?

1. First-order logic is an extension of propositional logic;
2. First-order logic is NP-complete.
3. In first-order logic you can use quantifiers over sets.
4. First-order logic is decidable.
5. First-order logic is an extension of arithmetic;
6. One can axiomatise integers in first-order logic;
7. Compactness is the following property: a set of formulas having arbitrarily large finite models has an infinite model;
8. Having proofs is good.
9. Vampire is a first-order theorem prover.

Future and Our Motivation

1. Theorem proving will remain central in software verification and program analysis. The role of theorem proving in these areas will be growing.
2. Theorem provers will be used by a large number of users who do not understand theorem proving and by users with very elementary knowledge of logic.
3. Reasoning with both quantifiers and theories will remain the main challenge in practical applications of theorem proving (at least) for the next decade.
4. Theorem provers will be used in reasoning with very large theories. These theories will appear in knowledge mining and natural language processing.

Future and Our Motivation

1. Theorem proving will remain central in software verification and program analysis. The role of theorem proving in these areas will be growing.
2. Theorem provers will be used by a large number of users who do not understand theorem proving and by users with very elementary knowledge of logic.
3. Reasoning with both quantifiers and theories will remain the main challenge in practical applications of theorem proving (at least) for the next decade.
4. Theorem provers will be used in reasoning with very large theories. These theories will appear in knowledge mining and natural language processing.

Future and Our Motivation

1. Theorem proving will remain central in software verification and program analysis. The role of theorem proving in these areas will be growing.
2. Theorem provers will be used by a large number of users who do not understand theorem proving and by users with very elementary knowledge of logic.
3. Reasoning with both quantifiers and theories will remain the main challenge in practical applications of theorem proving (at least) for the next decade.
4. Theorem provers will be used in reasoning with very large theories. These theories will appear in knowledge mining and natural language processing.

Future and Our Motivation

1. Theorem proving will remain central in software verification and program analysis. The role of theorem proving in these areas will be growing.
2. Theorem provers will be used by a large number of users who do not understand theorem proving and by users with very elementary knowledge of logic.
3. Reasoning with both quantifiers and theories will remain the main challenge in practical applications of theorem proving (at least) for the next decade.
4. Theorem provers will be used in reasoning with very large theories. These theories will appear in knowledge mining and natural language processing.

First-Order Theorem Proving. Example

Group theory theorem: if a group satisfies the identity $x^2 = 1$, then it is commutative.

First-Order Theorem Proving. Example

Group theory theorem: if a group satisfies the identity $x^2 = 1$, then it is commutative.
More formally: in a group "assuming that $x^2 = 1$ for all x prove that $x \cdot y = y \cdot x$ holds for all x, y."

First-Order Theorem Proving. Example

Group theory theorem: if a group satisfies the identity $x^2 = 1$, then it is commutative.
More formally: in a group "assuming that $x^2 = 1$ for all x prove that $x \cdot y = y \cdot x$ holds for all x, y."
What is implicit: axioms of the group theory.

$$
\begin{array}{l}
\forall x(1 \cdot x = x) \\
\forall x(x^{-1} \cdot x = 1) \\
\forall x \forall y \forall z((x \cdot y) \cdot z = x \cdot (y \cdot z))
\end{array}
$$

Formulation in First-Order Logic

Axioms (of group theory):	$\forall x(1 \cdot x = x)$
	$\forall x(x^{-1} \cdot x = 1)$
	$\forall x \forall y \forall z((x \cdot y) \cdot z = x \cdot (y \cdot z))$
Assumptions:	$\forall x(x \cdot x = 1)$
Conjecture:	$\forall x \forall y(x \cdot y = y \cdot x)$

In the TPTP Syntax

The TPTP library (Thousands of Problems for Theorem Provers), http://www.tptp.org contains a large collection of first-order problems. For representing these problems it uses the TPTP syntax, which is understood by all modern theorem provers, including Vampire.

In the TPTP Syntax

The TPTP library (Thousands of Problems for Theorem Provers), http://www.tptp.org contains a large collection of first-order problems.
For representing these problems it uses the TPTP syntax, which is understood by all modern theorem provers, including Vampire.
In the TPTP syntax this group theory problem can be written down as follows:

```
%---- 1 * x = 1
fof(left_identity,axiom,
    ! [X] : mult(e,X) = X).
%---- i(x) * x = 1
fof(left_inverse,axiom,
    ! [X] : mult(inverse(X),X) = e).
%---- (x * y) * z = x * (y * z)
fof(associativity,axiom,
    ! [X,Y,Z] : mult(mult(X,Y),Z) = mult(X,mult(Y,Z))).
%---- x * x = 1
fof(group_of_order_2,hypothesis,
    ! [X] : mult(X,X) = e).
%---- prove x * y = y * x
fof(commutativity,conjecture,
    ! [X] : mult(X,Y) = mult(Y,X)).
```

Running Vampire of a TPTP file

is easy: simply use

```
vampire <filename>
```

Running Vampire of a TPTP file

is easy: simply use

```
vampire <filename>
```

One can also run Vampire with various options, some of them will be explained later. For example, save the group theory problem in a file `group.tptp` and try

```
vampire --thanks Andrei group.tptp
```