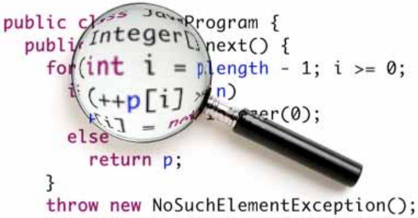
15-819M: Data, Code, Decisions

05: Proving Theorems in First-Order Logic with KeY

André Platzer

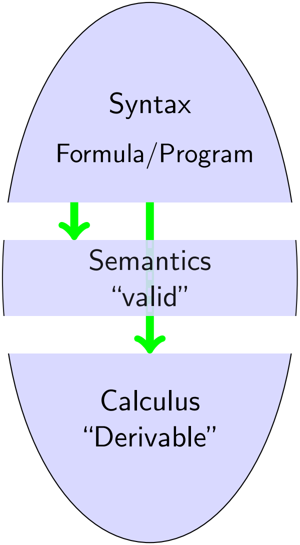
aplatzer@cs.cmu.edu
Carnegie Mellon University, Pittsburgh, PA



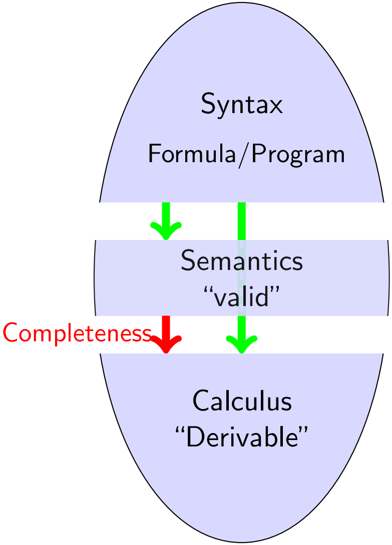
Outline

Outline

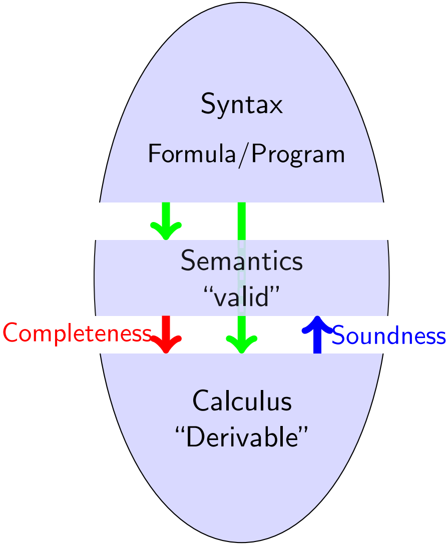
Syntax, Semantics, Calculus



Syntax, Semantics, Calculus



Syntax, Semantics, Calculus



Notation for Sequents

$$\psi_1, \ldots, \psi_m \quad \Longrightarrow \quad \phi_1, \ldots, \phi_n$$

Consider antecedent/succedent as sets of formulas, may be empty

Notation for Sequents

$$\psi_1, \ldots, \psi_m \quad \Longrightarrow \quad \phi_1, \ldots, \phi_n$$

Consider antecedent/succedent as sets of formulas, may be empty

Schema Variables

$\phi, \psi, \ldots$ match formulas, $\Gamma, \Delta, \ldots$ match sets of formulas
Characterize infinitely many sequents with a single schematic sequent

$$\Gamma \quad \Longrightarrow \quad \Delta, \phi \ \& \ \psi$$

Matches any sequent with occurrence of conjunction in succedent

Call $\phi \ \& \ \psi$ main formula and Γ, Δ side formulas of sequent

Any sequent of the form $\Gamma, \phi \implies \Delta, \phi$ is logically valid: axiom

Sequent Calculus Rules of Propositional Logic

Write syntactic transformation schema for sequents that reflects semantics of connectives as closely as possible

$$\text{RuleName}\ \frac{\overbrace{\Gamma_1 \Longrightarrow \Delta_1 \quad \cdots \quad \Gamma_r \Longrightarrow \Delta_r}^{\text{Premisses}}}{\underbrace{\Gamma \Longrightarrow \Delta}_{\text{Conclusion}}}$$

Sequent Calculus Rules of Propositional Logic

Write syntactic transformation schema for sequents that reflects semantics of connectives as closely as possible

$$\text{RuleName}\ \frac{\overbrace{\Gamma_1 \Longrightarrow \Delta_1 \quad \cdots \quad \Gamma_r \Longrightarrow \Delta_r}^{\text{Premisses}}}{\underbrace{\Gamma \Longrightarrow \Delta}_{\text{Conclusion}}}$$

Example

$$\text{andRight}\ \frac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \ \&\ \psi, \Delta}$$

Sequent Calculus Rules of Propositional Logic

Write syntactic transformation schema for sequents that reflects semantics of connectives as closely as possible

$$\text{RuleName}\ \frac{\overbrace{\Gamma_1 \Longrightarrow \Delta_1 \quad \cdots \quad \Gamma_r \Longrightarrow \Delta_r}^{\text{Premisses}}}{\underbrace{\Gamma \Longrightarrow \Delta}_{\text{Conclusion}}}$$

Example

$$\text{andRight}\ \frac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi\ \&\ \psi, \Delta}$$

Sound rule (essential): $\models (\Gamma_1 \Longrightarrow \Delta_1\ \&\ \cdots\ \&\ \Gamma_r \Longrightarrow \Delta_r) \text{ -> } (\Gamma \Longrightarrow \Delta)$

Sequent Calculus Rules of Propositional Logic

Write syntactic transformation schema for sequents that reflects semantics of connectives as closely as possible

$$\text{RuleName}\ \frac{\overbrace{\Gamma_1 \Longrightarrow \Delta_1 \quad \cdots \quad \Gamma_r \Longrightarrow \Delta_r}^{\text{Premisses}}}{\underbrace{\Gamma \Longrightarrow \Delta}_{\text{Conclusion}}}$$

Example

$$\text{andRight}\ \frac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi\ \&\ \psi, \Delta}$$

Sound rule (essential): $\models (\Gamma_1 \Longrightarrow \Delta_1 \ \& \cdots \& \ \Gamma_r \Longrightarrow \Delta_r) \mathrel{-\!\!>} (\Gamma \Longrightarrow \Delta)$

Complete rule (desirable): $\models (\Gamma \Longrightarrow \Delta) \mathrel{-\!\!>} (\Gamma_1 \Longrightarrow \Delta_1 \ \& \cdots \& \ \Gamma_r \Longrightarrow \Delta_r)$

Admissible to have no premisses (iff conclusion is valid, eg axiom)

Rules of Propositional Sequent Calculus

main	left side (antecedent)	right side (succedent)
not	$\dfrac{\Gamma \Longrightarrow \phi, \Delta}{\Gamma, !\,\phi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \Delta}{\Gamma \Longrightarrow !\,\phi, \Delta}$

Rules of Propositional Sequent Calculus

main	left side (antecedent)	right side (succedent)
not	$\dfrac{\Gamma \Longrightarrow \phi, \Delta}{\Gamma, !\,\phi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \Delta}{\Gamma \Longrightarrow !\,\phi, \Delta}$
and	$\dfrac{\Gamma, \phi, \psi \Longrightarrow \Delta}{\Gamma, \phi \mathbin{\&} \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \mathbin{\&} \psi, \Delta}$

Rules of Propositional Sequent Calculus

main	left side (antecedent)	right side (succedent)
not	$\dfrac{\Gamma \Longrightarrow \phi, \Delta}{\Gamma, !\,\phi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \Delta}{\Gamma \Longrightarrow !\,\phi, \Delta}$
and	$\dfrac{\Gamma, \phi, \psi \Longrightarrow \Delta}{\Gamma, \phi \;\&\; \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \;\&\; \psi, \Delta}$
or	$\dfrac{\Gamma, \phi \Longrightarrow \Delta \qquad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \phi \mid \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \psi, \Delta}{\Gamma \Longrightarrow \phi \mid \psi, \Delta}$

Rules of Propositional Sequent Calculus

main	left side (antecedent)	right side (succedent)
not	$\dfrac{\Gamma \Longrightarrow \phi, \Delta}{\Gamma, !\,\phi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \Delta}{\Gamma \Longrightarrow !\,\phi, \Delta}$
and	$\dfrac{\Gamma, \phi, \psi \Longrightarrow \Delta}{\Gamma, \phi \;\&\; \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \;\&\; \psi, \Delta}$
or	$\dfrac{\Gamma, \phi \Longrightarrow \Delta \qquad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \phi \mid \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \psi, \Delta}{\Gamma \Longrightarrow \phi \mid \psi, \Delta}$
imp	$\dfrac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \phi \mathbin{{-}{>}} \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \mathbin{{-}{>}} \psi, \Delta}$

Rules of Propositional Sequent Calculus

main	left side (antecedent)	right side (succedent)
not	$\dfrac{\Gamma \Longrightarrow \phi, \Delta}{\Gamma, !\,\phi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \Delta}{\Gamma \Longrightarrow !\,\phi, \Delta}$
and	$\dfrac{\Gamma, \phi, \psi \Longrightarrow \Delta}{\Gamma, \phi \;\&\; \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \;\&\; \psi, \Delta}$
or	$\dfrac{\Gamma, \phi \Longrightarrow \Delta \qquad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \phi \mid \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow \phi, \psi, \Delta}{\Gamma \Longrightarrow \phi \mid \psi, \Delta}$
imp	$\dfrac{\Gamma \Longrightarrow \phi, \Delta \qquad \Gamma, \psi \Longrightarrow \Delta}{\Gamma, \phi \;\text{->}\; \psi \Longrightarrow \Delta}$	$\dfrac{\Gamma, \phi \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \phi \;\text{->}\; \psi, \Delta}$

close $\dfrac{}{\Gamma, \phi \Longrightarrow \phi, \Delta}$ true $\dfrac{}{\Gamma \Longrightarrow \mathrm{true}, \Delta}$ false $\dfrac{}{\Gamma, \mathrm{false} \Longrightarrow \Delta}$

Sequent Calculus in KeY

Reduce a given sequent by applying rules and producing simpler subgoals until all leaves of proof tree are "axioms"

Example (KeY input syntax for propositional validity problem)

```
\predicates {
  p;
  q;
}
\problem {
  (p & (p -> q)) -> q
}
```

Demo

`Examples/lect09/prop.key`

Proving Validity of First-Order Formulas

Proving a universally quantified formula

$\forall\, T\, x;\, \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

Proving Validity of First-Order Formulas

Proving a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

All even numbers are divisible by 2 $\quad \forall\, \mathbf{int}\, x;\ (\mathsf{even}(x)\ \texttt{->}\ \mathsf{divByTwo}(x))$

Proving Validity of First-Order Formulas

Proving a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

All even numbers are divisible by 2 $\quad \forall\, \mathbf{int}\, x;\ (\mathsf{even}(x) \mathrel{-\!\!>} \mathsf{divByTwo}(x))$

Let c be an arbitrary number $\quad$ Declare “unused” constant $\mathbf{int}\ \mathtt{c}$

Proving Validity of First-Order Formulas

Proving a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

All even numbers are divisible by 2	$\forall\, \mathbf{int}\, x;\ (\mathsf{even}(x) \mathrel{-\!\!>} \mathsf{divByTwo}(x))$
Let c be an arbitrary number	Declare "unused" constant $\mathbf{int}$ c
The even number c is divisible by 2	$\mathsf{even}(c) \mathrel{-\!\!>} \mathsf{divByTwo}(c)$

Proving Validity of First-Order Formulas

Proving a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

All even numbers are divisible by 2	$\forall\, \mathbf{int}\, x;\ (\mathrm{even}(x) \text{ -> } \mathrm{divByTwo}(x))$
Let c be an arbitrary number	Declare "unused" constant $\mathbf{int}\ \mathtt{c}$
The even number c is divisible by 2	$\mathrm{even}(c) \text{ -> } \mathrm{divByTwo}(c)$

Sequent rule $\forall$-right

$$\text{forallRight}\ \frac{\Gamma \Longrightarrow [x/c]\,\phi, \Delta}{\Gamma \Longrightarrow \forall\, T\, x;\ \phi, \Delta}$$

- $[x/c]\,\phi$ is result of replacing each occurrence of x in ϕ with c
- c **new** constant of type T

Proving Validity of First-Order Formulas

Proving an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

Proving Validity of First-Order Formulas

Proving an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

There is at least one prime number $\quad \exists\, \mathbf{int}\, x;\ \mathrm{prime}(x)$

Proving Validity of First-Order Formulas

Proving an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

There is at least one prime number $\exists\, \mathbf{int}\, x;\ \mathrm{prime}(x)$

Provide any "witness", say, 7 Use variable-free term $\mathbf{int}\ 7$

Proving Validity of First-Order Formulas

Proving an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

There is at least one prime number	$\exists\, \mathbf{int}\, x;\ \mathsf{prime}(x)$
Provide any "witness", say, 7	Use variable-free term $\mathbf{int}\ 7$
7 is a prime number	$\mathsf{prime}(7)$

Proving Validity of First-Order Formulas

Proving an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a claim proven in mathematics?

There is at least one prime number	$\exists\, \mathbf{int}\, x;\ \mathrm{prime}(x)$
Provide any "witness", say, 7	Use variable-free term $\mathbf{int}\ 7$
7 is a prime number	$\mathrm{prime}(7)$

Sequent rule $\exists$-right

$$\text{existsRight}\ \frac{\Gamma \Longrightarrow [x/t']\,\phi,\ \exists\, T\, x;\ \phi, \Delta}{\Gamma \Longrightarrow \exists\, T\, x;\ \phi, \Delta}$$

- t' any variable-free term with declared type $T' \sqsubseteq T$
- Proof might not work with t'! Need to keep premise to try again

Proving Validity of First-Order Formulas

Using a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Proving Validity of First-Order Formulas

Using a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

We know that "all" primes are odd $\quad \forall\, \mathbf{int}\, x;\ (\mathrm{prime}(x) \mathrel{-\!\!>} \mathrm{odd}(x))$

Proving Validity of First-Order Formulas

Using a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

We know that "all" primes are odd	$\forall\, \mathbf{int}\, x;\ (\mathsf{prime}(x) \;\text{->}\; \mathsf{odd}(x))$
In particular, this holds for 17	Use variable-free term $\mathbf{int}\ 17$

Proving Validity of First-Order Formulas

Using a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

We know that "all" primes are odd	$\forall\, \mathbf{int}\, x;\ (\mathrm{prime}(x) \mathrel{->} \mathrm{odd}(x))$
In particular, this holds for 17	Use variable-free term $\mathbf{int}$ 17
We know: if 17 is prime it is odd	$\mathrm{prime}(17) \mathrel{->} \mathrm{odd}(17)$

Proving Validity of First-Order Formulas

Using a universally quantified formula

$\forall\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

We know that "all" primes are odd	$\forall\, \mathbf{int}\, x;\ (\mathrm{prime}(x) \mathrel{->} \mathrm{odd}(x))$
In particular, this holds for 17	Use variable-free term $\mathbf{int}\ 17$
We know: if 17 is prime it is odd	$\mathrm{prime}(17) \mathrel{->} \mathrm{odd}(17)$

Sequent rule ∀-left

$$\text{forallLeft}\ \frac{\Gamma, \forall\, T\, x;\ \phi,\ [x/t']\,\phi \Longrightarrow \Delta}{\Gamma, \forall\, T\, x;\ \phi \Longrightarrow \Delta}$$

- t' any variable-free term with declared type $T' \sqsubseteq T$
- We might need other instances besides t'! Keep premise

Proving Validity of First-Order Formulas

Using an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Proving Validity of First-Order Formulas

Using an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Every set s can be well-ordered $\exists\, \texttt{OSet}\, x;\ (\mathrm{sameElem}(s, x)\ \&\ \mathrm{wellOrder}(x))$

Proving Validity of First-Order Formulas

Using an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Every set s can be well-ordered $\exists\, \texttt{OSet}\, x;\ (\mathrm{sameElem}(s, x)\ \&\ \mathrm{wellOrder}(x))$

Let s' be a well-order of s $\quad s'$ **new** constant of type OSet

Proving Validity of First-Order Formulas

Using an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Every set s can be well-ordered	$\exists\, \texttt{OSet}\, x;\ (\text{sameElem}(s, x)\ \&\ \text{wellOrder}(x))$
Let s' be a well-order of s	s' **new** constant of type OSet
We know: s' is well-order of s	$\text{sameElem}(s, s')\ \&\ \text{wellOrder}(s')$

Proving Validity of First-Order Formulas

Using an existentially quantified formula

$\exists\, T\, x;\ \phi$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Every set s can be well-ordered	$\exists\, \texttt{OSet}\, x;\ (\mathrm{sameElem}(s, x)\ \&\ \mathrm{wellOrder}(x))$
Let s' be a well-order of s	s' **new** constant of type OSet
We know: s' is well-order of s	$\mathrm{sameElem}(s, s')\ \&\ \mathrm{wellOrder}(s')$

Sequent rule ∃-left

$$\text{existsLeft}\ \frac{\Gamma, [x/c]\,\phi \Longrightarrow \Delta}{\Gamma, \exists\, T\, x;\ \phi \Longrightarrow \Delta}$$

- c **new** constant of type T

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\exists\, x;\ \forall\, y;\ p(x, y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x, y)$$

Untyped logic: let static type of x and y be $\top$

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\frac{\forall\, y;\ p(c, y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x, y)}{\exists\, x;\ \forall\, y;\ p(x, y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x, y)}$$

$\exists$-left: substitute new constant c of type $\top$ for x

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\cfrac{\cfrac{\forall y;\ p(c,y) \Longrightarrow \exists x;\ p(x,d)}{\forall y;\ p(c,y) \Longrightarrow \forall y;\ \exists x;\ p(x,y)}}{\exists x;\ \forall y;\ p(x,y) \Longrightarrow \forall y;\ \exists x;\ p(x,y)}$$

$\forall$-right: substitute new constant d of type $\top$ for y

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\dfrac{\dfrac{\dfrac{p(c, d), \forall y;\ p(c, y) \Longrightarrow \exists x;\ p(x, d)}{\forall y;\ p(c, y) \Longrightarrow \exists x;\ p(x, d)}}{\forall y;\ p(c, y) \Longrightarrow \forall y;\ \exists x;\ p(x, y)}}{\exists x; \forall y;\ p(x, y) \Longrightarrow \forall y;\ \exists x;\ p(x, y)}$$

$\forall$-left: free to substitute any term of type $\top$ for y, choose d

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$
\cfrac{\cfrac{\cfrac{p(c, d) \Longrightarrow \exists\, x;\ p(x, d)}{\forall\, y;\ p(c, y) \Longrightarrow \exists\, x;\ p(x, d)}}{\forall\, y;\ p(c, y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x, y)}}{\exists\, x;\ \forall\, y;\ p(x, y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x, y)}
$$

$\forall$-left not needed anymore (hide)

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\cfrac{\cfrac{\cfrac{\cfrac{p(c,d) \Longrightarrow p(c,d),\ \exists\, x;\ p(x,y)}{p(c,d) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}}{\exists\, x;\ \forall\, y;\ p(x,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}$$

$\exists$-right: free to substitute any term of type $\top$ for x, choose c

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\dfrac{\dfrac{\dfrac{\dfrac{p(c,d) \Longrightarrow p(c,d)}{p(c,d) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}}{\exists\, x;\ \forall\, y;\ p(x,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}$$

$\exists$-right not needed anymore (hide)

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{p(c,d) \Longrightarrow p(c,d)}}{p(c,d) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}}{\exists\, x;\ \forall\, y;\ p(x,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}$$

Close

Proving Validity of First-Order Formulas

Example (A simple theorem about binary relations)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{p(c,d) \Longrightarrow p(c,d)}}{p(c,d) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \exists\, x;\ p(x,d)}}{\forall\, y;\ p(c,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}}{\exists\, x;\ \forall\, y;\ p(x,y) \Longrightarrow \forall\, y;\ \exists\, x;\ p(x,y)}$$

Demo

`Examples/lect09/relSimple.key`

Proving Validity of First-Order Formulas

Using an equation between terms

$t \doteq t'$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Proving Validity of First-Order Formulas

Using an equation between terms

$t \doteq t'$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Use $x \doteq y-1$ to simplify $x+1/y$ $\qquad$ $x \doteq y-1 \implies 1 \doteq x+1/y$

Proving Validity of First-Order Formulas

Using an equation between terms

$t \doteq t'$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Use $x \doteq y-1$ to simplify $x+1/y$ $\qquad$ $x \doteq y-1 \Longrightarrow 1 \doteq x+1/y$

Replace x in conclusion with right-hand side of equation

Proving Validity of First-Order Formulas

Using an equation between terms

$t \doteq t'$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Use $x \doteq y-1$ to simplify $x+1/y$ $\qquad x \doteq y-1 \Longrightarrow 1 \doteq x+1/y$

Replace x in conclusion with right-hand side of equation

We know: $x+1/y$ equal to $y-1+1/y$ $\qquad x \doteq y-1 \Longrightarrow 1 \doteq y-1+1/y$

Proving Validity of First-Order Formulas

Using an equation between terms

$t \doteq t'$ is true in any model $\mathcal{M}$

How is such a fact used in a mathematical proof?

Use $x \doteq y-1$ to simplify $x+1/y$ $\qquad x \doteq y-1 \Longrightarrow 1 \doteq x+1/y$

Replace x in conclusion with right-hand side of equation

We know: $x+1/y$ equal to $y-1+1/y$ $\qquad x \doteq y-1 \Longrightarrow 1 \doteq y-1+1/y$

Sequent rule $\doteq$-left

$$\text{applyEq}\ \frac{\Gamma, t \doteq t', [t/t']\,\psi \Longrightarrow [t/t']\,\phi, \Delta}{\Gamma, t \doteq t', \psi \Longrightarrow \phi, \Delta}$$

- Always replace left- with right-hand side (use eqSymm if necessary)
- Replacing term must be type-compatible wity replaced term
- t any variable-free term with declared type T, t' with type $T' \sqsubseteq T$

Proving Validity of First-Order Formulas

Closing a subgoal in a proof

- We derived a sequent that is obviously valid

$$\text{close}\ \frac{}{\Gamma, \phi \Longrightarrow \phi, \Delta} \qquad \text{true}\ \frac{}{\Gamma \Longrightarrow \mathrm{true}, \Delta} \qquad \text{false}\ \frac{}{\Gamma, \mathrm{false} \Longrightarrow \Delta}$$

- We derived an equation that is obviously valid

$$\text{eqClose}\ \frac{}{\Gamma \Longrightarrow t \doteq t, \Delta}$$

Features of the KeY Theorem Prover

Demo

`Examples/lect09/rel.key`

Feature List

- Can work on multiple proofs simultaneously (task list)
- Proof trees visualized as JAVA Swing tree
- Point-and-click navigation within proof
- Undo proof steps, prune proof trees
- Pop-up menu with proof rules applicable in pointer focus
- Preview of rule effect as tool tip
- Quantifier instantiation and equality rules by drag-and-drop
- Possible to hide (and unhide) parts of a sequent
- Saving and loading of proofs

Sequent Calculus for FOL at One Glance

	left side, antecedent	right side, succedent
$\forall$	$\dfrac{\Gamma, \forall\, T\, x;\ \phi,\ [x/t']\,\phi \Longrightarrow \Delta}{\Gamma, \forall\, T\, x;\ \phi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow [x/c]\,\phi, \Delta}{\Gamma \Longrightarrow \forall\, T\, x;\ \phi, \Delta}$
$\exists$	$\dfrac{\Gamma, [x/c]\,\phi \Longrightarrow \Delta}{\Gamma, \exists\, T\, x;\ \phi \Longrightarrow \Delta}$	$\dfrac{\Gamma \Longrightarrow [x/t']\,\phi,\ \exists\, T\, x;\ \phi, \Delta}{\Gamma \Longrightarrow \exists\, T\, x;\ \phi, \Delta}$
$\doteq$	$\dfrac{\Gamma, t \doteq t', [t/t']\,\psi \Longrightarrow [t/t']\,\phi, \Delta}{\Gamma, t \doteq t', \psi \Longrightarrow \phi, \Delta}$	$\dfrac{}{\Gamma \Longrightarrow t \doteq t, \Delta}$

- $[t/t']\,\phi$ is result of replacing each occurrence of t in ϕ with t'
- t any variable-free term with declared type T
t' any variable-free term with declared type $T' \sqsubseteq T$
- c **new** constant of type T (occurs not on current proof branch)
- Equations can be reversed by commutativity

First-Order Validity Problems in KeY Syntax

```
\sorts { // types are called ``sorts''
   Person; // one declaration per line
}
\functions {
   int age(Person); // ``int'' predefined type
}
\predicates {
   parent(Person,Person);
}
\problem { // Formula to be proven valid
   \forall Person son; \forall Person father;
      (parent(father,son) -> age(father) > age(son))
}
```

Types and Symbols with Fixed Meaning

When doing JAVA verification, we want many function and predicate symbols to have the semantics prescribed by the JLS in all models

Reserved symbols with fixed meaning so far: $\doteq$, $\sqsubseteq T$, (T)

Types and Symbols with Fixed Meaning

When doing Java verification, we want many function and predicate symbols to have the semantics prescribed by the JLS in all models

Reserved symbols with fixed meaning so far: $\doteq$, $\sqsubseteq T$, (T)

Types & symbols with fixed meaning in context of modeling Java

- $\mathcal{D}^{\mathbf{int}} = \{d \in \mathcal{D} \mid \delta(d) = \mathbf{int}\} = \mathbb{Z}$
- KeY can switch to $\{\texttt{Integer.MIN\_VALUE}, \ldots, \texttt{Integer.MAX\_VALUE}\}$
- Default interpretation (and always used in first-order) is $\mathbb{Z}$
- Similar for **short**, **byte**
- Value types incomparable to reference types
- $\mathcal{D}^{\mathbf{boolean}} = \{d \in \mathcal{D} \mid \delta(d) = \mathbf{boolean}\} = \{F, T\}$
- Usual operators in expressions as pre-defined signature symbols:
  Fixed meaning: $\mathcal{I}(+) = +_{\mathbb{Z}}$, $\mathcal{I}(*) = *_{\mathbb{Z}}$, ...
  `+,-,*,/,%,mod,...,-1,0,1,...,<,<=,>,>=,TRUE,FALSE`

Rules for Type Casts and Type Predicates

- **Type predicate** formulas $t \sqsubseteq T$

  true iff dynamic type $\delta(val_{\mathcal{M}}(t))$ is subtype of T
- **Type cast** terms $(T)t$

  yields $val_{\mathcal{M}}(t)$ (identity) if cast succeeds, arb. element otherwise

Typical typing rule

The run-time type of a term is always compatible to its declared type

$$\text{typeStatic}\ \frac{\Gamma, t \sqsubseteq T \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta} \quad T \text{ declared type of } t$$

Ensures type-safety of typed first-order logic

- KeY first-order strategy applies suitable typing rules automatically
- All rules in KeY-Book Chapter 2, p59

Outline

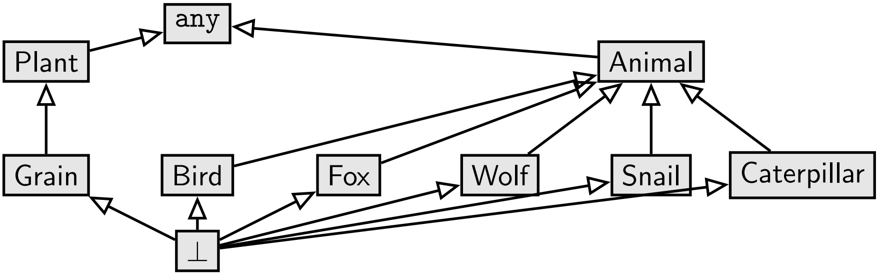
An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

An Example with Types




```
\sorts { Animal;
         Wolf \extends Animal;
         Bird \extends Animal;
         Fox \extends Animal;
         Caterpillar \extends Animal;
         Snail \extends Animal;
         Plant;
         Grain \extends Plant; }
```

An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

```
\predicates {
  eats(Animal,any);
  smaller(any,any);
}
```

An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

```
(\forall Caterpillar c; \forall Bird b; smaller(c,b)) &
(\forall Snail s; \forall Bird b; smaller(s,b)) &
(\forall Bird b; \forall Fox f; smaller(b,f)) &
(\forall Fox f; \forall Wolf w; smaller(f,w))
```

An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

```
(\forall Wolf w; \forall Fox f; !eats(w,f)) &
(\forall Wolf w; \forall Grain g; !eats(w,g)) &
(\forall Bird b; \forall Caterpillar c; eats(b,c)) &
(\forall Bird b; \forall Snail s; !eats(b,s))
(\forall Caterpillar c; \exists Plant p; eats(c,p)) &
(\forall Snail s; \exists Plant p; eats(s,p)) &
```

An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

```
(\forall Animal a;
  ((\forall Plant p; eats(a,p)) |
   (\forall Animal as;
     ((smaller(as,a) &
      \exists Plant p; eats(as,p)) -> eats(a,as)))))
```

An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

```
( ... ) ->
  (\exists Animal a;
   \exists Animal ga; ((\exists Grain g; eats(ga,g)) &
                       eats(a,ga)))
```

An Example with Types

Schubert's Steamroller

Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also, there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants.

Therefore, there is an animal that likes to eat a grain-eating animal.

```
( ... ) ->
  (\exists Animal a;
   \exists Animal ga; ((\exists Grain g; eats(ga,g)) &
                       eats(a,ga)))
```

Demo

Examples/lect09/sr.key

Outline

Automated Proof Search

KeY has built-in heuristics to apply FO rules automatically

- Select Proof Search Strategy "FOL"
- Specify Max. Rule Applications or Time limit
- Run/Stop button
- See Goals tab

Automated Proof Search

KeY has built-in heuristics to apply FO rules automatically

- Select Proof Search Strategy "FOL"
- Specify Max. Rule Applications or Time limit
- Run/Stop button
- See Goals tab

Look out for common problems

- Long branches with same rule applied to quantified formulas
- Too low bound on proof search
- If search doesn't terminate:
  - Check Java DL Proof Search Strategy
  - Instantiate quantifiers "by-hand"
    (might need to declare suitable constant in problem)

Outline

Failed Proofs

Sometimes (often) interactive or automatic proof attempts fail

Reasons for failed proofs

- Automatically:
  - The automatic proof strategy of KeY is too weak
  - Did you check Proof Strategy FOL?
- Manually:
  - Did you use the right instantiations?
  - Perhaps you need to apply an equality?
- Your goal is not a valid formula!
  An unsuccessful proof can give important clues why!

Learning from Failed Proofs

Theorem

Let the formula G be the goal of a sequent proof.
Assume there is an open leaf $L = \Gamma \Longrightarrow \Delta$ in a sequent proof such that:

1. *L is not closed*
2. *There is a first-order model $\mathcal{M}$ that:*
    - $\mathcal{M} \models \gamma$ *for all* $\gamma \in \Gamma$
    - $\mathcal{M} \models\, !\,\delta$ *for all* $\delta \in \Delta$

Then $\mathcal{M} \models\, !\,G$, i.e., $\mathcal{M}$ is a counter example for G.

Learning from Failed Proofs

How to proceed

1. Java DL Proof Search Strategy with
   Quantifier Treatment unrestricted
2. Run prover, inspect open Goals L
3. If necessary, instantiate $\forall$-left, $\exists$-right by hand
4. Find model that makes L's antecedent true and succedent false
5. Go back to G and find out was was wrong
   Often, the patch is to add a $\gamma \in L$ or a $!\,\delta \in L$ to the premise of G

Learning from Failed Proofs

How to proceed

1. Java DL Proof Search Strategy with
   Quantifier Treatment unrestricted
2. Run prover, inspect open Goals L
3. If necessary, instantiate $\forall$-left, $\exists$-right by hand
4. Find model that makes L's antecedent true and succedent false
5. Go back to G and find out was was wrong
   Often, the patch is to add a $\gamma \in L$ or a $!\,\delta \in L$ to the premise of G

Demo

`Examples/lect09/model.key`

Outline

Literature for this Lecture

Essential

KeY Book Verification of Object-Oriented Software (see course web page), Chapter 10: Using KeY (up to and incl. 10.2.2)

KeY Book Verification of Object-Oriented Software (see course web page), Chapter 2: First-Order Logic

Recommended/Background

Huth & Ryan Logic in Computer Science, 2nd edn., Cambridge University Press, 2004

Fitting First-Order Logic and Automated Theorem Proving, 2nd edn., Springer 1996