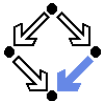


# The Language of Logic

Wolfgang Schreiner  
Wolfgang.Schreiner@risc.uni-linz.ac.at

Research Institute for Symbolic Computation (RISC)  
Johannes Kepler University, Linz, Austria  
<http://www.risc.uni-linz.ac.at>





# The Language of Logic

Two kinds of syntactic phrases.

- **Term**  $T$  denoting an object.
  - Variable  $x$
  - Object constant  $c$
  - Function application  $f(T_1, \dots, T_n)$   
 $n$ -ary function constant  $f$  (may be infix)
- **Formula**  $F$  denoting a truth value.
  - Atomic formula  $p(T_1, \dots, T_n)$  (may be infix)  
 $n$ -ary predicate constant  $p$ .
  - Negation  $\neg F$  ("not  $F$ ")
  - Conjunction  $F_1 \wedge F_2$  (" $F_1$  and  $F_2$ ")
  - Disjunction  $F_1 \vee F_2$  (" $F_1$  or  $F_2$ ")
  - Implication  $F_1 \Rightarrow F_2$  ("if  $F_1$ , then  $F_2$ ")
  - Equivalence  $F_1 \Leftrightarrow F_2$  ("if  $F_1$ , then  $F_2$ , and vice versa")
  - Universal quantification  $\forall x : F$  ("for all  $x$ ,  $F$ ")
  - Existential quantification  $\exists x : F$  ("for some  $x$ ,  $F$ ")



# Syntactic Shortcuts

- $\forall x_1, \dots, x_n : F$ 
  - $\forall x_1 : \dots : \forall x_n : F$
- $\exists x_1, \dots, x_n : F$ 
  - $\exists x_1 : \dots : \exists x_n : F$
- $\forall x \in S : F$ 
  - $\forall x : x \in S \Rightarrow F$
- $\exists x \in S : F$ 
  - $\exists x : x \in S \wedge F$
- **let**  $x = T$  **in**  $F$ 
  - $F[T/x]$

Help to make formulas more readable.



# Example

---

Terms and formulas may appear in various syntactic forms.

- **Terms:**

$$\exp(x)$$

$$a \cdot b + 1$$

$$a[i] \cdot b$$

$$\sqrt{\frac{x^2+2x+1}{(y+1)^2}}$$

- **Formulas:**

$$a^2 + b^2 = c^2$$

$$n \mid 2n$$

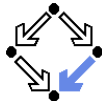
$$\forall x \in \mathbb{N} : x \geq 0$$

$$\forall x \in \mathbb{N} : 2 \mid x \vee 2 \mid (x + 1)$$

$$\forall x \in \mathbb{N}, y \in \mathbb{N} : x < y \Rightarrow$$

$$\exists z \in \mathbb{N} : x + z = y$$

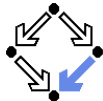
Terms and formulas may be nested arbitrarily deeply.



# The Meaning of Formulas

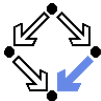
- Atomic formula  $p(T_1, \dots, T_n)$ 
  - True if the predicate denoted by  $p$  holds for the values of  $T_1, \dots, T_n$ .
- Negation  $\neg F$ 
  - True if and only if  $F$  is false.
- Conjunction  $F_1 \wedge F_2$  (“ $F_1$  and  $F_2$ ”)
  - True if and only if  $F_1$  and  $F_2$  are both true.
- Disjunction  $F_1 \vee F_2$  (“ $F_1$  or  $F_2$ ”)
  - True if and only if at least one of  $F_1$  or  $F_2$  is true.
- Implication  $F_1 \Rightarrow F_2$  (“if  $F_1$ , then  $F_2$ ”)
  - False if and only if  $F_1$  is true and  $F_2$  is false.
- Equivalence  $F_1 \Leftrightarrow F_2$  (“if  $F_1$ , then  $F_2$ , and vice versa”)
  - True if and only if  $F_1$  and  $F_2$  are both true or both false.
- Universal quantification  $\forall x : F$  (“for all  $x$ ,  $F$ ”)
  - True if and only if  $F$  is true for every possible value assignment of  $x$ .
- Existential quantification  $\exists x : F$  (“for some  $x$ ,  $F$ ”)
  - True if and only if  $F$  is true for at least one value assignment of  $x$ .

# Example



We assume the domain of natural numbers and the “classical” interpretation of constants  $1, 2, +, =, <$ .

- $1 + 1 = 2$ 
  - True.
- $1 + 1 = 2 \vee 2 + 2 = 2$ 
  - True.
- $1 + 1 = 2 \wedge 2 + 2 = 2$ 
  - False.
- $1 + 1 = 2 \Rightarrow 2 = 1 + 1$ 
  - True.
- $1 + 1 = 1 \Rightarrow 2 + 2 = 2$ 
  - True.
- $1 + 1 = 2 \Rightarrow 2 + 2 = 2$ 
  - False.
- $1 + 1 = 1 \Leftrightarrow 2 + 2 = 2$ 
  - True.



# Example

- $x + 1 = 1 + x$ 
  - True, for every assignment of a number  $a$  to variable  $x$ .
- $\forall x : x + 1 = 1 + x$ 
  - True (because for every assignment  $a$  to  $x$ ,  $x + 1 = 1 + x$  is true).
- $x + 1 = 2$ 
  - If  $x$  is assigned “one”, the formula is true.
  - If  $x$  is assigned “two”, the formula is false.
- $\exists x : x + 1 = 2$ 
  - True (because  $x + 1 = 2$  is true for assignment “one” to  $x$ ).
- $\forall x : x + 1 = 2$ 
  - False (because  $x + 1 = 2$  is false for assignment “two” to  $x$ ).
- $\forall x : \exists y : x < y$ 
  - True (because for every assignment  $a$  to  $x$ , there exists the assignment  $a + 1$  to  $y$  which makes  $x < y$  true).
- $\exists y : \forall x : x < y$ 
  - False (because for every assignment  $a$  to  $y$ , there is the assignment  $a + 1$  to  $x$  which makes  $x < y$  false).



# The Usage of Formulas

Precise formulation of statements describing object relationships.

- **Statement:**

If  $x$  and  $y$  are natural numbers and  $y$  is not zero, then  $q$  is the truncated quotient of  $x$  divided by  $y$ .

- **Formula:**

$$x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge y \neq 0 \Rightarrow \\ q \in \mathbb{N} \wedge \exists r \in \mathbb{N} : r < y \wedge x = y \cdot q + r$$

- **Problem specification:**

Given natural numbers  $x$  and  $y$  such that  $y$  is not zero, compute the truncated quotient  $q$  of  $x$  divided by  $y$ .

- Inputs:  $x, y$

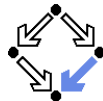
- Input condition:  $x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge y \neq 0$

- Output:  $q$

- Output condition:  $q \in \mathbb{N} \wedge \exists r \in \mathbb{N} : r < y \wedge x = y \cdot q + r$



# Problem Specifications



- **Specification** of a computation problem:
  - Input: variables  $X_1 \in S_1, \dots, X_n \in S_n$
  - Input condition: formula  $I$  in which only  $X_1, \dots, X_n$  may be free.
  - Output: variables  $Y_1 \in T_1, \dots, Y_m \in T_n$
  - Output condition: formula  $O$  in which only  $X_1, \dots, X_n, Y_1, \dots, Y_m$  may be free.

A variable is **free** in a formula, if it occurs in the formula outside the scope of a quantifier (such as  $\forall$  or  $\exists$ ).

- **Implementation** of the specification:
  - A function (program)  $f : S_1 \times \dots \times S_n \rightarrow T_1 \times \dots \times T_m$  such that
$$\forall X_1 \in S_1, \dots, X_n \in S_n : I(X_1, \dots, X_n) \Rightarrow$$
$$\text{let } (Y_1, \dots, Y_m) = f(X_1, \dots, X_n) \text{ in}$$
$$O(X_1, \dots, X_n, Y_1, \dots, Y_m)$$
  - For all arguments that satisfy the input condition,  $f$  must compute results that satisfy the output condition.

**Basis of all specification formalisms.**



## Example

---

Given an array  $a$ , a position  $p$ , and a length  $l$ , compute the array  $b$  derived from  $a$  by removing  $a[p], \dots, a[p + l]$ .

■ Input:  $a \in \text{Arr}$ ,  $p \in \mathbb{N}$ ,  $l \in \mathbb{N}$

■ Input condition:

$$p + l \leq \text{length}(a)$$

■ Output:  $b \in \text{Arr}$

■ Output condition:

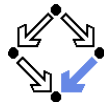
**let**  $n = \text{length}(a)$  **in**

$$\text{length}(b) = n - l \wedge$$

$$(\forall i \in \mathbb{N} : i < p \Rightarrow b[i] = a[i]) \wedge$$

$$(\forall i \in \mathbb{N} : p \leq i < n - l \Rightarrow b[i] = a[i + l])$$

# Proving Formulas



- **Proof:** a structured argument that a formula is true.
  - Can be depicted as a tree whose nodes represent proof situations.
  - Each proof situation consists of knowledge (formulas assumed to be true) and a goal (a formula to be proved relative to that knowledge).
  - The root goal is the overall formula to be proved.
- **Proof rules:** describe how a proof situation can be reduced to zero or more “subsituations”.
  - Zero subsituations: the current goal has been proved.
  - One or more subsituations: the current goal has been proved, if all subgoals have been proved.
  - **Top-down rules:** focus on goal formula.
    - Goal formula is decomposed into simpler formulas.
  - **Bottom-up rules:** focus on some formula in knowledge.
    - Additional formulas are added to the knowledge.

In each proof situation, we aim at showing that the goal is “apparently” true with respect to the given knowledge.

# Conjunction



Formula  $F_1 \wedge F_2$ .

- Formula as goal.

- Create two subsituations with goals  $F_1$  and  $F_2$ .

*We have to show  $F_1 \wedge F_2$ .*

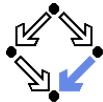
- *We show  $F_1$ : ... (proof continues with goal  $F_1$ )*
- *We show  $F_2$ : ... (proof continues with goal  $F_2$ )*

- Formula in knowledge.

- Create one subsituation with  $F_1$  and  $F_2$  in knowledge.

*We know  $F_1 \wedge F_2$ . We thus know  $F_1$  and also  $F_2$ .  
(proof continues with current goal and additional  
knowledge  $F_1$  and  $F_2$ )*

# Disjunction



Formula  $F_1 \vee F_2$ .

- Formula as goal.
  - Create one subsituation where  $F_2$  is proved under the assumption that  $F_1$  does not hold (or vice versa):

*We have to show  $F_1 \vee F_2$ . We assume  $\neg F_1$  and show  $F_2$ .  
(proof continues with goal  $F_2$  and additional knowledge  $\neg F_1$ )*

- Formula in knowledge.
  - Create two subsituations, one with  $F_1$  and one with  $F_2$  in knowledge.

*We know  $F_1 \vee F_2$ . We thus proceed by case distinction:*

    - *Case  $F_1$ : ... (proof continues with current goal and additional knowledge  $F_1$ ).*
    - *Case  $F_2$ : ... (proof continues with current goal and additional knowledge  $F_2$ ).*

# Implication



Formula  $F_1 \Rightarrow F_2$ .

- Formula as goal.

- Create one subsituation where  $F_2$  is proved under the assumption that  $F_1$  holds:

*We have to show  $F_1 \Rightarrow F_2$ . We assume  $F_1$  and show  $F_2$ .  
(proof continues with goal  $F_2$  and additional knowledge  $F_1$ )*

- Formula in knowledge.

- Create two subsituations, one with goal  $F_1$  and one with knowledge  $F_2$ .

*We know  $F_1 \Rightarrow F_2$ .*

- *We show  $F_1$ : ... (proof continues with goal  $F_1$ )*
- *We know  $F_2$ : ... (proof continues with current goal and additional knowledge  $F_2$ ).*

# Equivalence



Formula  $F_1 \Leftrightarrow F_2$ .

- Formula as goal.
  - Create two subsituations with implications in both directions as goals:  
*We have to show  $F_1 \Leftrightarrow F_2$ .*
    - *We show  $F_1 \Rightarrow F_2$ : ... (proof continues with goal  $F_1 \Rightarrow F_2$ )*
    - *We show  $F_2 \Rightarrow F_1$ : ... (proof continues with goal  $F_2 \Rightarrow F_1$ )*
- Formula in knowledge.
  - Create two subsituations, one with goal  $F_1$  and one with knowledge  $F_2$ .  
*We know  $F_1 \Leftrightarrow F_2$ .*
    - *We show  $F_1 (\neg F_1)$ : ... (proof continues with goal  $F_1$ )*
    - *We know  $F_2 (\neg F_2)$ : ... (proof continues with current goal and additional knowledge  $F_2$ )*

# Universal Quantification



Formula  $\forall x : F$

- Formula as goal.

- Introduce new (arbitrarily named) constant  $x_0$  and create one subsituation with goal  $F[x_0/x]$ .

*We have to show  $\forall x : F$ . Take arbitrary  $x_0$ .*

*We show  $F[x_0/x]$ . (proof continues with goal  $F[x_0/x]$ )*

- Formula in knowledge.

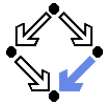
- Choose term  $T$  to create one subsituation with formula  $F[T/x]$  added to the knowledge.

*We know  $\forall x : F$  and thus also  $F[T/x]$ .*

*(proof continues with current goal and additional knowledge  $F[T/x]$ )*



# Existential Quantification



Formula  $\exists x : F$

- Formula as goal.

- Choose term  $T$  to create one subsituation with goal  $F[T/x]$ .

*We have to show  $\exists x : F$ . It suffices to show  $F[T/x]$ .  
(proof continues with goal  $F[T/x]$ )*

- Formula in knowledge

- Introduce new (arbitrarily named constant)  $x_0$  and create one subsituation with additional knowledge  $F[x_0/x]$ .

*We know  $\exists x : F$ . Let  $x_0$  be such that  $F[x_0/x]$ .  
(proof continues with current goal and additional  
knowledge  $F[x_0/x]$ )*



# Formula Equivalences

Formulas in knowledge and goals may be replaced by equivalent formulas.

- $\neg\neg F_1 \iff F_1$
- $\neg(F_1 \wedge F_2) \iff \neg F_1 \vee \neg F_2$
- $\neg(F_1 \vee F_2) \iff \neg F_1 \wedge \neg F_2$
- $\neg(F_1 \Rightarrow F_2) \iff F_1 \wedge \neg F_2$
- $\neg\forall x : F \iff \exists x : \neg F$
- $\neg\exists x : F \iff \forall x : \neg F$
- $F_1 \Rightarrow F_2 \iff \neg F_2 \Rightarrow \neg F_1$
- $F_1 \Rightarrow F_2 \iff \neg F_1 \vee F_2$
- $F_1 \Leftrightarrow F_2 \iff \neg F_1 \Leftrightarrow \neg F_2$
- ...

Transformation sometimes offers new view on a proof situation.

# Indirect Proofs



Proof situation with goal formula  $G$ .

- Add  $\neg G$  to the knowledge and show a contradiction.
  - Prove that “false” is true.
  - Prove that a formula  $F$  is true and also prove that it is false.
  - Prove that a formula  $F$  in the knowledge is false, i.e. that  $\neg F$  is true.
    - Switches goal  $G$  and some knowledge  $F$  (negating both).

Sometimes simpler than a direct proof.



# Example

---

We show

$$(a) (\exists x : p(x)) \wedge (\forall x : p(x) \Rightarrow \exists y : q(x, y)) \Rightarrow (\exists x, y : q(x, y))$$

We assume

$$(1) (\exists x : p(x)) \wedge (\forall x : p(x) \Rightarrow \exists y : q(x, y))$$

and show

$$(b) \exists x, y : q(x, y)$$

From (1), we know

$$(2) \exists x : p(x)$$

$$(3) \forall x : p(x) \Rightarrow \exists y : q(x, y)$$

From (2) we know for some  $x_0$

$$(4) p(x_0)$$

...



## Example (Contd)

---

...

From (3), we know

$$(5) p(x_0) \Rightarrow \exists y : q(x_0, y)$$

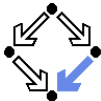
From (4) and (5), we know

$$(6) \exists y : q(x_0, y)$$

From (6), we know for some  $y_0$

$$(7) q(x_0, y_0)$$

From (7), we know (b). QED.



# Example

---

We show

$$(a) (\exists x : \forall y : P(x, y)) \Rightarrow (\forall y : \exists x : P(x, y))$$

We assume

$$(1) \exists x : \forall y : P(x, y)$$

and show

$$(b) \forall y : \exists x : P(x, y)$$

Take arbitrary  $y_0$ . We show

$$(c) \exists x : P(x, y_0)$$

From (1) we know for some  $x_0$

$$(2) \forall y : P(x_0, y)$$

From (2) we know

$$(3) P(x_0, y_0)$$

From (3), we know (c). QED.



## Example (Indirect Proof)

---

We show

$$(a) (\exists x : \forall y : P(x, y)) \Rightarrow (\forall y : \exists x : P(x, y))$$

We assume

$$(1) \exists x : \forall y : P(x, y)$$

and show

$$(b) \forall y : \exists x : P(x, y)$$

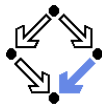
We assume

$$(2) \neg \forall y : \exists x : P(x, y)$$

and show a contradiction.

...

# Example (Indirect Proof Contd)



...

From (2), we know

$$(3) \exists y : \forall x : \neg P(x, y)$$

Let  $y_0$  be such that

$$(4) \forall x : \neg P(x, y_0)$$

From (1) we know for some  $x_0$

$$(5) \forall y : P(x_0, y)$$

From (5) we know

$$(6) P(x_0, y_0)$$

From (4), we know

$$(7) \neg P(x_0, y_0)$$

From (6) and (7), we have a contradiction. QED.