Mathematics for Computer Science

José Proença & Alexandre Madeira
(slides from Luis Soares Barbosa)

HASLab
HIGH-ASSURANCE SOFTWARE LABORATORY
INESCTEC

Universidade do Minho

Expressions

September-October, 2017

Expressions

- Syntax refers to the structure of expressions, or the rules for putting symbols together to form an expression.
- Semantics refers to the meaning of expressions, or how they are evaluated.

Expressions

Recall the syntax of conventional mathematical expressions:

$$E \quad := \quad c \mid v \mid (E) \mid \circ E \mid E * E$$

where c belongs to a set of constants, v to a set of variables, $\circ$ is a unary operator and $*$ a binary operator.

Textual substitution

$$E[x := E']$$

introduced as an inference rule:

$$\frac{E}{E[x := E']}\,(\text{substitution})$$

Equality

- Semantic characterization — in terms of expression evaluation: Evaluation of the expression $X = Y$ in a state yields the value true if expressions X and Y have the same value and yields false if they have different values.
- Syntactic characterization — in terms of a set of laws that can be used to show that two expressions are equal, without calculating their values.

Equality

reflexivity: $X = X$

symmetry: $(X = Y) = (Y = X)$

transitivity: $$\frac{X = Y,\, Y = Z}{X = Z}$$

Leibniz: $$\frac{X = Y}{E[z := X] = E[z := Y]}$$

Equality

Functions

- A function is a rule for computing a value from another value
- application (can be defined in terms of textual substitution)

$$f(v) = E$$

  then

$$f\ X = E[v := X]$$

- The Leibniz rule links equality and function application:

$$\frac{X = Y}{f\,X = f\,Y}\ \text{(Leibniz)}$$

- composition

Boolean expressions

i.e. expressions taking values in $\{\textsc{True}, \textsc{False}\}$

Boole's great contribution was an algebraic basis for logic, something Leibniz had dreamed about 200 years earlier and that De Morgan was unable to devise.
In *The Laws of Thought* Boole's aim was to *"investigate the fundamental laws ... by which reasoning is performed, ... give expression to them in the language of a Calculus, and upon this foundation ... establish the Science of Logic ... "*.

Boolean expressions

- Syntax: Boolean connectives

$$E \;:=\; \textsc{True} \mid \textsc{False} \mid v \mid (E) \mid \neg E \mid E \wedge E \mid E \vee E \mid E \Rightarrow E$$

- Semantics: interpretation; truth tables
- Equality versus equivalence
  - $=$ has a conjunctional flavour:
    $b = c = d$ is an abbreviation for $b \equiv c \wedge c \equiv d$
    Therefore $b = c = d$ and $(b = c) = d$ are different.
  - $\equiv$ is associative (thus not conjunctional)

Boolean expressions

$$
\begin{array}{ll}
 & \textsc{False} = \textsc{False} = \textsc{True} \\
= & \quad \{ \text{ $=$ is abbreviates a conjunction } \} \\
 & (\textsc{False} = \textsc{False}) \wedge (\textsc{False} = \textsc{True}) \\
= & \quad \{ \text{ evaluate both $=$ } \} \\
 & \textsc{True} \wedge \textsc{False} \\
= & \quad \{ \text{ evaluation $\wedge$ } \} \\
 & \textsc{False}
\end{array}
$$

Boolean expressions

$$
\begin{array}{ll}
 & \textsc{False} \equiv \textsc{False} \equiv \textsc{True} \\
= & \quad \{ \text{ evaluates first } \equiv \ \} \\
 & \textsc{True} \equiv \textsc{True} \\
= & \quad \{ \text{ evaluate } \equiv \ \} \\
 & \textsc{True}
\end{array}
$$

Treat the conjunctional use of $=$ and other operators as syntactic sugar, i.e. as an extension to the basic definition of expressions to make writing some expressions easier. Whenever an expression that contains this syntactic sugar is to be evaluated or manipulated, first remove the syntactic sugar.

Boolean expressions: Satisfiability and validity

- A Boolean expression is satisfied in a state if its value is true in that state;
- it is satisfiable if there is a state in which it is satisfied;
- it is valid if it is satisfied in every state;
- A valid boolean expression is called a tautology

Boolean expressions: Duality

The dual E_D of a boolean expression E is built from E by interchanging occurrences of $\textsc{True}$ and $\textsc{False}$, $\wedge$ and $\vee$, $\equiv$ and $\not\equiv$, $\Rightarrow$ and $\not\Leftarrow$, and $\Leftarrow$ and $\not\Rightarrow$,

Duality meta-theorem

- E is valid iff $\neg E_D$ is valid
- $E \equiv F$ is valid iff $E_D \equiv F_D$ is valid

Modelling natural language propositions

Proposition

is a statement that can be interpreted as being either true or false

Modelling natural language requirements as Boolean expressions

- Introduce boolean variables to denote subpropositions.
- Replace these subpropositions by their corresponding boolean variables.
- Translate the results of step 2 into a Boolean expression,

Modelling natural language propositions

Exercises

If Superman were able and willing to prevent evil, he would do so. If Superman were unable to prevent evil, he would be impotent; if he were unwilling to prevent evil, he would be malevolent. Superman does not prevent evil. If Superman exists, he is neither impotent nor malevolent. Therefore, Superman does not exist.