Formal Software Engineering (génie logiciel avec l'approche formelle)

J. Christian Attiogbé

Master Alma, Septembre 2015

Two parts course, shared with Pr. Claude Jard and M. Benoît Delahaye

First Order Logic

A proposition is a sentence named P, Q, E... with a value TRUE or FALSE; the construction of a proposition is made with the following grammar:

$$\begin{array}{lcl} prop & ::= & P, Q, E, \ldots \\ & \mid & prop \wedge prop \\ & \mid & \neg\ prop \\ & \mid & prop \Rightarrow prop \end{array}$$

Parentheses can be used if necessary.

Other operators (logical connectors) : $\vee$, $\equiv$

The semantics of a proposition (with the connectors) is given by a truth table (Exercice).

Examples of Proposition

A cat with a hat is Lion

Peter rides bycicle

$0 > 3$

Inference rules of propositional calculus

$\wedge\ intr$	$\dfrac{HYP \vdash P \qquad HYP \vdash Q}{HYP \vdash P \wedge Q}$	use backward to decompose into simple subgoals with the same hypotheses
$\wedge\ elim$	$\dfrac{HYP \vdash P \wedge Q}{HYP \vdash P \qquad HYP \vdash Q}$	
$\Rightarrow intr$	$\dfrac{HYP, P \vdash Q}{HYP \vdash P \Rightarrow Q}$	deduction rule
$\Rightarrow elim$	$\dfrac{HYP \vdash P \Rightarrow Q}{HYP, P \vdash Q}$	anti-deduction

Modus Ponens	$\dfrac{HYP \vdash P \qquad HYP \vdash P \Rightarrow Q}{HYP \vdash Q}$	
Contradiction	$\dfrac{HYP, \neg Q \vdash P \qquad HYP, \neg Q \vdash \neg P}{HYP \vdash Q}$	first rule for ¬
	$\dfrac{HYP, Q \vdash P \qquad HYP, Q \vdash \neg P}{HYP \vdash \neg Q}$	second rule for ¬

Predicates

Propositional calculus deals with : absolute truth.
Predicate calculus deals with : relative truth,
it is an extension of propositional calculus.

$$x > 2$$

$$x \in I\!N \Rightarrow x \geq 0$$

Two kinds of variables are used in predicates: free variables and bound variables which are introduced with quantifiers.

How to use predicates

- Substitution

$$[x := 5](x \in I\!N \Rightarrow x \geq 0)$$

$$(5 \in I\!N \Rightarrow 5 \geq 0)$$

$$[x := elephant](BigEars(x) \Rightarrow African(x))$$

- Quantification

$$\forall x.BigEars(x) \Longrightarrow African(x),$$

$$\forall x.(Animal(x) \wedge BigEars(x)) \Longrightarrow African(x)$$

How to use predicates

Construction of predicates

Predicat	::=	$Predicat \Rightarrow Predicat$
	|	$Predicat \wedge Predicat$
	|	$\neg\ Predicat$
	|	$\forall Variable.Predicat$
	|	$[Variable := Expression]Predicat$
	|	$Expression = Expression$
Expression	::=	$Variable$
	|	$[Variable := Expression]Expression$
Variable	::=	$Identifier$

Usage of Logics

- for modelling : *predicates*

  predicate = formula to be proved

$$P \wedge Q$$
$$P \Rightarrow Q$$
$$0 < 3$$
$$\{0,3\} \subset \{0,4,8,3\}$$

- for reasoning : *sequents*

$$H \vdash P$$

$$\left.\begin{array}{l} H : Hypotheses \\ P : conjecture \end{array}\right\} predicates$$

Reasoning

- Inference rules

  An inference rule links sequents and its defines a valid step of a proof.
  An inference rule has the following shape:

$$\frac{\sum_1, \sum_2, \ldots, \sum_n}{\sum}$$

The sequents $\sum_1, \sum_2, \ldots, \sum_n$ are called *antecedents*, and the sequent $\sum$ is called *consequent*.

Reasoning (continued)

- Proof principle

  To prove a sequent, one uses the inference rules
  - as derivation rules : forward rule application,
  - as reduction rules : backward rule application.

Implementation

- Theorem to prove / Inference

  To prove a theorem

$$P \vdash Q$$

  one transforms it into inference rule

$$\frac{H \vdash P}{H \vdash Q}$$

- Proof - forward or backward - tactics