

Méthodes Formelles (enfin) appliquées

Pierre Bourreau

¹IUT Bordeaux 1 - Licence Pro

February 2, 2010

Introduction

Introduction

Logique du
premier ordre

Logique des Propositions

Logique des Prédicats

Logique du premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Un peu de concret!

JML

ESC/Java2

Conclusion

Conclusion

Introduction

Introduction

Logique du premier ordre

Logique des Propositions

Logique des Prédicats

Logique du premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Un peu de concret!

JML

ESC/Java2

Conclusion

Conclusion

- ▶ pas rapide...
- ▶ pas pratique...
- ▶ erreurs humaines au niveau démonstration.
- ▶ intérêt: utiliser la logique en programmation.

ESC/Java 2

- ▶ intégration de ces méthodes à Java (fini le papier)
- ▶ rapide (... plus qu'un programmeur avec un stylo...)
- ▶ **utilité?**

Qu'est-ce que c'est?

- ▶ insertion de "commentaires" spéciaux (**JML**)
- ▶ un interprète analyse le code+JML
- ▶ indique si des erreurs d'exécution sont susceptibles d'arriver.

Introduction

Logique du premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Conclusion

Introduction

Logique du
premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Conclusion

Logique des propositions

Définitions

- ▶ Les propositions $p, q, r, s \dots, \perp, \top$
- ▶ Les connecteurs binaires $\wedge, \vee, \rightarrow, \leftrightarrow$
- ▶ Les connecteurs unaires \neg

Quelques règles

- ▶ $\neg\neg p \iff p$
- ▶ $\neg(p \wedge q) \iff \neg p \vee \neg q$
- ▶ $(p \rightarrow q) \wedge (q \rightarrow p) \iff (p \leftrightarrow q)$
- ▶ $p \rightarrow q \iff \neg p \vee q$

- ▶ On dit qu'une formule \mathcal{F} de la logique des propositions est valide (noté $\models \mathcal{F}$) ssi $\mathcal{F} \iff 1$

Déduction, calcul

- ▶ $\neg 0 \iff 1$
- ▶ $p \vee \neg p \iff 1$
- ▶ $p \leftrightarrow p \iff 1$
- ▶ Modus Ponens: $(p \rightarrow q) \wedge p \iff q$

Exemple: exprimer que le solde d'un compte est supérieur à une demande de retrait.

Définitions

- ▶ Les variables x, y, \dots
- ▶ Les constantes n – aires (fonctions): a, b, \dots, f, g, \dots
- ▶ Les prédicats: P, Q, R, \dots
- ▶ Les connecteurs binaires $\wedge, \vee, \rightarrow, \leftrightarrow$
- ▶ Les connecteurs unaires \neg, \forall, \exists

Remarques

- ▶ soit V l'ensemble des variables, et C_1 celui des constantes 1-aire; une constante n – aire f est une fonction de V dans C_1
- ▶ un prédicat $P : V \rightarrow \{\perp, \top\}$
- ▶ beaucoup plus expressif
 - ▶ exemple: arithmétique($<, =, +, 1, 2, \dots?$)
 - ▶ exemple: exprimer que dans un tableau de longueur n , l'élément position i est supérieur à l'élément suivant.

- ▶ On dit qu'une formule \mathcal{F} de la logique des propositions est valide (noté $\models \mathcal{F}$) ssi $\mathcal{F} \iff 1$

Déduction, calcul

- ▶ on garde le calcul des propositions (sur des variables ou constantes)
- ▶ $\forall x P(x) \Rightarrow P(a)$
- ▶ La validité n'a pas grande importance.
- ▶ Il faut s'intéresser aux modèles... c'est ce que l'on va construire!

Sommaire

ESC/Java 2

Pierre Bourreau

Introduction

Introduction

Logique du premier ordre

Logique des Propositions

Logique des Prédicats

Logique du
premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Un peu de concret!

JML

ESC/Java2

Conclusion

Conclusion

C'est quoi?

- ▶ Langage de spécifications
- ▶ Proche programmation par contrat de Eiffel
- ▶ **Description** du comportement de méthodes ET de classes
- ▶ Principalement: **invariants, pré-conditions, post-conditions**
- ▶ Peut être utilisé par différents outils (JMLRAC, ESC/Java)

- ▶ Exprimer des conditions fortes sur nos programmes
- ▶ ... grâce à la logique du 1er ordre

Exemple

```
/*@ requires amount >= 0;
   ensures balance ==
\old(balance-amount) &&
   \result == balance;
@*/
public int debit(int amount) {
    ...
}
```

- ▶ `requires`
précondition.
- ▶ `ensures`
postcondition
- ▶ `invariant`
...
- ▶ `signals`
condition de levée d'exception
- ▶ `assignable`
attributs pouvant être modifiés
- ▶ `pure`
méthode ne modifiant rien
- ▶ `assert`
spécifie qu'une propriété doit être vérifiée

- ▶ `non_null`
permet d'éviter des conditions sur toute la classe
- ▶ `\old(<name>)`
valeur avant exécution
- ▶ `\result`
valeur résultante
- ▶ `\forall`, `\exists`
quantificateurs \forall, \exists
- ▶ `a ==> b`, `a <==> b` “a implique b” et “a équivaut à b”
- ▶ `&&`, `||`, `not`, \wedge , \vee , *not*

1. En entrée: programme Java annoté
2. **traducteur**: formules, conditions de vérification
3. **theorem prover (SIMPLIFY)**: contre-exemples
4. **post-processeur**: avertissements

Sommaire

Introduction

Logique du premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Conclusion

ESC/Java 2

Pierre Bourreau

Introduction

Logique du
premier ordre

Logique des Propositions

Logique des Prédicats

Un peu de concret!

JML

ESC/Java2

Conclusion

- ▶ A vous de jouer!
- ▶ J'attends vos conclusions, critiques, ...

- ▶ Installation:
 - ▶ <http://kind.ucd.ie/products/opensource/ESCJava2/>
- ▶ Documentation:
 - ▶ dans la version téléchargée
 - ▶ <http://wiki.student.info.ucl.ac.be/index.php/EscJava>
 - ▶ on the web...