propositional logic

logical verification

week 1

2004 09 08

who

Femke

Freek

Paulien

what

- 13 lectures
- practical work
  - Coq proofs & paper proofs
  - 9 out of 12
- final test

stuff

- hand out
  - course notes
  - exercises
- web page
  - hand out
  - files for the exercises
  - solutions for the exercises
  - slides
  - old tests

where

- lectures: S201
  Coq lab: S345

  `http://www.cs.vu.nl/~tcs/al/`
  `tcs@cs.vu.nl`

- Freek:

  tuesdays & wednesdays: U333
  `freek@cs.ru.nl`

topic

computer science

formal methods

proof assistants

type theory

examples of applications of formal methods

- Intel bug
- driverless train
- spacecraft
- credit cards

proof assistants

- PVS
- Coq/NuPRL
- ACL2
- HOL/Isabelle
- Mizar

what we will do here

Curry-Howard-de Bruijn isomorphism

	logic	$\longleftrightarrow$	**lambda calculus**
	natural deduction		'type theory'
on paper	•		•
in Coq	•		•

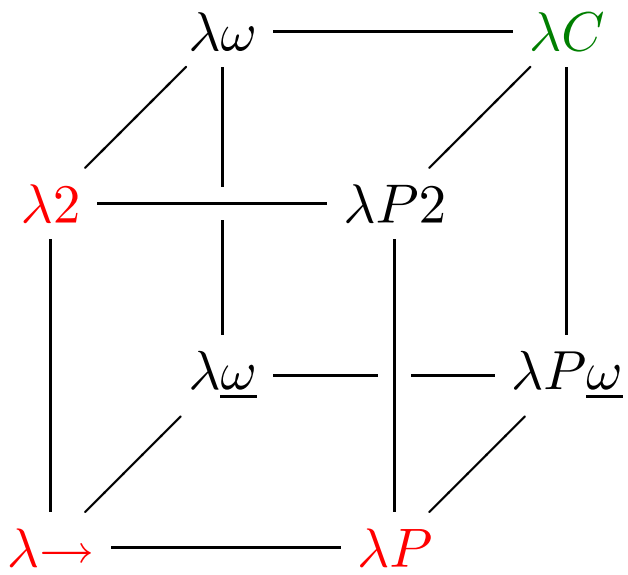
logics

- the systems in this course

propositional logic $\longleftrightarrow$ calculus called $\lambda\!\rightarrow$

predicate logic $\longleftrightarrow$ calculus called λP

2nd order propositional logic $\longleftrightarrow$ calculus called $\lambda 2$



and also

- inductive types
  - built-in
  - higher order encoding
- program extraction

today

first order propositional logic

first order predicate logic

second order propositional logic

formulas

$$A \to B$$

$$\bot$$

$$\top$$

$$\neg A \qquad := \quad A \to \bot$$

$$A \wedge B$$

$$A \vee B$$

logical rules

two kinds of rules

- introduction rules
- elimination rules

rules for $\rightarrow$

introduction rule

$$\dfrac{\begin{array}{c}[A^x]\\ \vdots \\ B\end{array}}{A \rightarrow B}\, I[x]\rightarrow$$

assumption

elimination rule

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ A \rightarrow B & A\end{array}}{B}\, E\rightarrow$$

example

$$A \to A$$

bigger example

$$((A \to B) \to (C \to D)) \to C \to B \to D$$

rules for the other connectives

$\bot$ elimination

$\top$ introduction

$\neg$ introduction

$\neg$ elimination

excluded middle $A \vee \neg A$

$\wedge$ introduction

$\wedge$ elimination, left rule

$\wedge$ elimination, right rule

$\vee$ introduction, left rule

$\vee$ introduction, right rule

$\vee$ elimination

the rules for $\vee$

$\vee$ introduction

$$\frac{\begin{array}{c}\vdots\\ A\end{array}}{A \vee B}\ Il\vee \qquad \frac{\begin{array}{c}\vdots\\ B\end{array}}{A \vee B}\ Ir\vee$$

$\vee$ elimination

$$\frac{\begin{array}{ccc}\vdots & \vdots & \vdots\\ A \vee B & A \to C & B \to C\end{array}}{C}\ E\vee$$

example with $\vee$

$$(A \vee B) \rightarrow (B \vee A)$$

Coq

- goals

  Coq tells you what is left to be proved

- tactics

  you tell Coq how to prove it

Coq term syntax

```
A -> B

False

True

~A

A /\ B

A \/ B
```

the Coq language

commands

- `Parameter`
- `Lemma`
- `Qed`

tactics

• `intro`	$I[x]{\rightarrow}$
• `apply`	$E{\rightarrow}$
• `elim`	$E\bot\ \ El{\wedge}\ \ Er{\wedge}\ \ E{\vee}$
• `exact`	*assumption*
• `split`	$I{\wedge}$
• `left right`	$Il{\vee}\ \ Ir{\vee}$

interfaces

- `coqtop` + `coqc`

  'command line'

- `xemacs` + Proof General
- `coqide`
- `pcoq`

example

```
A -> A
```

- with `coqtop`
- with Proof General

the second example

```
((A -> B) -> (C -> D)) -> C -> B -> D
```

and the third example

```
(A \/ B) -> (B \/ A)
```

summary

- formal methods
- type theory

  the Curry-Howard-de Bruijn isomorphism
- propositional logic
- Coq