COMP4161: Advanced Topics in Software Verification

λ^{\frown} and HOL

Gerwin Klein, June Andronick, Ramana Kumar, Miki Tanaka S2/2017



data61.csiro.au

DATA

Last time...

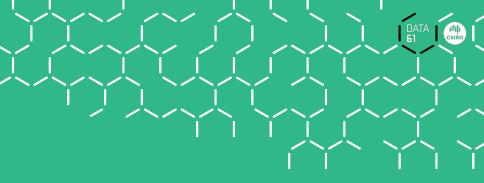


- → Simply typed lambda calculus: λ^{\rightarrow}
- → Typing rules for λ^{\rightarrow} , type variables, type contexts
- \clubsuit $\beta\text{-reduction}$ in λ^{\rightarrow} satisfies subject reduction
- \clubsuit $\beta\text{-reduction}$ in λ^{\rightarrow} always terminates
- \rightarrow Types and terms in Isabelle

Content

	ontent	DATA I
 Lambda Calculus, natural deduction [1,2] Higher Order Logic [3^a] Term rewriting [4] → Proof & Specification Techniques Inductively defined sets, rule induction [5] Datatypes, recursion, induction [6, 7] Hoare logic, proofs about programs, C verification [8^b,9] (mid-semester break) Writing Automated Proof Methods [10] 	→ Intro & motivation, getting started	[1]
 Inductively defined sets, rule induction [5] Datatypes, recursion, induction [6, 7] Hoare logic, proofs about programs, C verification [8^b,9] (mid-semester break) Writing Automated Proof Methods [10] 	 Lambda Calculus, natural deduction Higher Order Logic 	[3ª]
	 Inductively defined sets, rule induction Datatypes, recursion, induction Hoare logic, proofs about programs, C verification (mid-semester break) Writing Automated Proof Methods 	[6, 7] [8 ^b ,9] [10]

^aa1 due; ^ba2 due; ^ca3 due



Preview: Proofs in Isabelle

Proofs in Isabelle

DATA 61

General schema:

```
lemma name: "<goal>"
apply <method>
apply <method>
```

done

→ Sequential application of methods until all subgoals are solved.

The Proof State



1.
$$\bigwedge x_1 \dots x_p \cdot \llbracket A_1; \dots; A_n \rrbracket \Longrightarrow B$$

2. $\bigwedge y_1 \dots y_q \cdot \llbracket C_1; \dots; C_m \rrbracket \Longrightarrow D$

$x_1 \dots x_p$	Parameters
$A_1 \dots A_n$	Local assumptions
В	Actual (sub)goal

Isabelle Theories



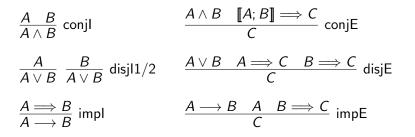
```
Syntax:
theory MyTh
imports ImpTh<sub>1</sub> ... ImpTh<sub>n</sub>
begin
(declarations, definitions, theorems, proofs, ...)*
end
```

- → MyTh: name of theory. Must live in file MyTh.thy
- → $ImpTh_i$: name of *imported* theories. Import transitive.

Unless you need something special: theory *MyTh* imports Main begin ... end

Natural Deduction Rules





For each connective $(\land, \lor, \text{ etc})$: introduction and elimination rules

Proof by assumption



apply assumption

proves

1. $\llbracket B_1; \ldots; B_m \rrbracket \Longrightarrow C$

by unifying C with one of the B_i

There may be more than one matching B_i and multiple unifiers.

Backtracking!

Explicit backtracking command: back

Intro rules



Intro rules decompose formulae to the right of \Longrightarrow .

```
apply (rule <intro-rule>)
```

Intro rule $\llbracket A_1; \ldots; A_n \rrbracket \Longrightarrow A$ means \rightarrow To prove A it suffices to show $A_1 \ldots A_n$

Applying rule $\llbracket A_1; \ldots; A_n \rrbracket \Longrightarrow A$ to subgoal *C*:

- \rightarrow unify A and C
- → replace C with n new subgoals $A_1 \ldots A_n$

Elim rules



Elim rules decompose formulae on the left of \Longrightarrow .

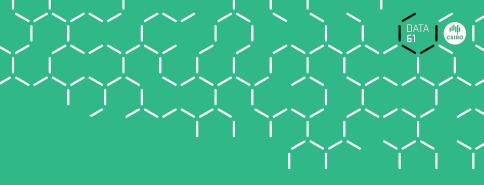
```
apply (erule <elim-rule>)
```

Elim rule $\llbracket A_1; \ldots; A_n \rrbracket \Longrightarrow A$ means

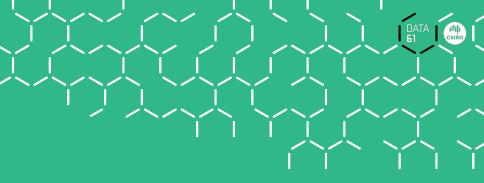
→ If I know A_1 and want to prove A it suffices to show $A_2 \dots A_n$

Applying rule $\llbracket A_1; \ldots; A_n \rrbracket \Longrightarrow A$ to subgoal C: Like **rule** but also

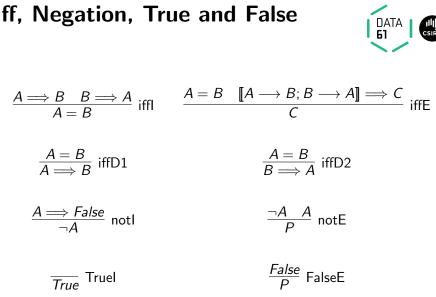
- \rightarrow unifies first premise of rule with an assumption
- \rightarrow eliminates that assumption







More Proof Rules



Iff, Negation, True and False

14 | COMP4161 | © Data61, CSIRO: provided under Creative Commons Attribution License

Equality



$$\frac{s=t}{t=t}$$
 refl $\frac{s=t}{t=s}$ sym $\frac{r=s}{r=t} \frac{s=t}{t}$ trans

$$\frac{s=t P s}{P t}$$
 subst

Rarely needed explicitly — used implicitly by term rewriting

Classical



$$\overline{P = True \lor P = False}$$
 True-or-False

 $\overline{P \lor \neg P}$ excluded-middle

$$\frac{\neg A \Longrightarrow \textit{False}}{A} \text{ ccontr } \frac{\neg A \Longrightarrow A}{A} \text{ classical}$$

- → excluded-middle, ccontr and classical not derivable from the other rules.
- \rightarrow if we include True-or-False, they are derivable

They make the logic "classical", "non-constructive"

Cases



$\overline{P \lor \neg P}$ excluded-middle

is a case distinction on type bool

Isabelle can do case distinctions on arbitrary terms:

apply (case_tac term)

Safe and not so safe



Safe rules preserve provability

conjl, impl, notl, iffi, refl, ccontr, classical, conjE, disjE

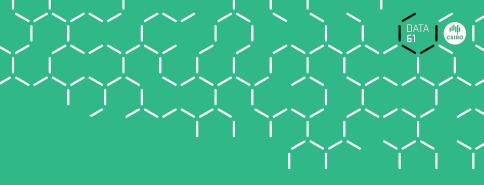
$$\frac{A \quad B}{A \land B} \text{ conjl}$$

Unsafe rules can turn a provable goal into an unprovable one

disjl1, disjl2, impE, iffD1, iffD2, notE

$$\frac{A}{A \lor B}$$
 disjl1

Apply safe rules before unsafe ones



Demo

What we have learned so far...



- → natural deduction rules for $\land,\,\lor,\,\longrightarrow,\,\neg,$ iff...
- \rightarrow proof by assumption, by intro rule, elim rule
- → safe and unsafe rules
- → indent your proofs! (one space per subgoal)
- → prefer implicit backtracking (chaining) or $rule_tac$, instead of back
- → prefer and defer
- → oops and sorry





Assignment 1 is out today!

Reminder: DO NOT COPY

- → Assignments and exams are take-home. This does NOT mean you can work in groups. Each submission is personal.
- → For more info, see Plagiarism Policy