AR Coursework QA

Filip Smola

4th November 2024

Information

- Demonstrator/TA: Filip Smola `f.smola@ed.ac.uk`
- Lab sessions: Mondays from 9am to 11am in 4.12, Appleton Tower
- Submission deadline: 12 noon on 18th November

Structured Higher-Order Statements

Lemmas

Higher-order logic (meta-level) statement:

```
lemma finite_UnI:
  "⋀F G. ⟦finite F; finite G⟧ ⟹ finite (F ∪ G)"
proof -
  fix F G :: "'a set"
  assume "finite F"
  assume "finite G"
  show "?thesis F G" sorry
qed
```

Structured Higher-Order Statements

Lemmas

Letting Isabelle take care of universally quantifying free variables:

```
lemma finite_UnI:
  "⟦finite F; finite G⟧ ⟹ finite (F ∪ G)"
proof -
  assume "finite F"
  assume "finite G"
  show ?thesis sorry
qed
```

Structured Higher-Order Statements

Lemmas

Structured statement using outer syntax:

```
lemma finite_UnI:
  assumes "finite F"
      and finite_G: "finite G"
    shows "finite (F ∪ G)"
proof -
  show ?thesis
    using ‘finite F‘ finite_G sorry
qed
```

Structured Higher-Order Statements

Lemmas

Explicitly stating type of free variables:

```
lemma finite_UnI:
      fixes F G :: "'a set"
  assumes "finite F"
      and finite_G: "finite G"
    shows "finite (F ∪ G)"
proof -
  show ?thesis
    using ‘finite F‘ finite_G sorry
qed
```

Structured Higher-Order Statements

Lemmas

No need to use structured statements for simple cases:

```
lemma finite_lt:
  "finite {x :: nat. x < n}"
  by simp
```

Structured Higher-Order Statements

Intermediate Facts

Single higher-order statement:

```
have "⋀F G. ⟦finite F; finite G⟧ ⟹ finite (F ∪ G)"
proof -
  fix F G :: "'a set"
  assume "finite F"
  assume "finite G"
  show "?thesis F G" sorry
qed
then have ...
```

Structured Higher-Order Statements

Intermediate Facts

Raw proof block:

```
{
  fix F G :: "'a set"
  assume "finite F"
  assume "finite G"
  show "finite (F ∪ G)" sorry
}
then have ...
```

Structured Higher-Order Statements

Intermediate Facts

Structured statement:

```
have "finite (F ∪ G)"
  if "finite F" and "finite G" for F G :: "'a set"
proof -
  show ?thesis
    using that sorry
qed
then have ...
```

Gathering Facts in Isar

Manually:

```
have a: "x < 0 ⟹ P"
  sorry
have b: "x = 0 ⟹ P"
  sorry
have c: "x > 0 ⟹ P"
  sorry
have P
  using a b c by blast
```

Gathering Facts in Isar

Automatically:

```
have "x < 0 ⟹ P"
  sorry
moreover have "x = 0 ⟹ P"
  sorry
moreover have "x > 0 ⟹ P"
  sorry
ultimately have P
  by blast
```

Overview

- Submission deadline: 12 noon on 18th November
- Automated tactics (`auto`, `simp`, ...) can be used from Part 2 onwards
- Use "Query" panel or **find_theorems** to search for useful theorems
- Break things into sublemmas, especially in Part 2
- Ask questions (labs, Piazza, email)