Automated Reasoning 2017/2018
Coursework
Theorem Proving in Isabelle

Imogen Morris Jacques Fleuriot

16th October 2017

Introduction

The coursework for Automated Reasoning is designed to test and help develop your understanding and practical skills using the interactive theorem prover Isabelle/HOL.

Part 1 consists of a set of exercises involving the use of basic inference rules in Isabelle/HOL. You will have to prove theorems in propositional and first order logic.

In Part 2 you will have to prove more complex theorems in some formalisations of geometry in Isabelle/HOL. You will be asked to formalise axiomatic systems for geometry and prove theorems resulting from these axioms. Moreover, you will explore some models for these axiomatic systems.
To get started, download the file practical.thy from:

`http://www.inf.ed.ac.uk/teaching/courses/ar/`

Essential Reading

As you will be using Isabelle interactively, you will need to be familiar with the system before you start. Formal mathematics is not trivial! You will find this assignment much easier if you attend the lectures, attempt the various Isabelle exercises given on the course webpages and ask questions about using Isabelle before you start. You can find a lot of useful information on using

Isabelle in the Isabelle/HOL tutorial, located in the 'Documentation' tab in jEdit. In particular, Chapter 5 has a lot of information on all the basic reasoning tools and rules.

You may also find it useful to take a look at Chapter 5 of Concrete Semantics, where a clear overview of declarative (structured) proofs is presented:

`http://www.concrete-semantics.org/concrete-semantics.pdf`.

Part 1: Some propositional and first-order proofs [40%]

In this section you will be asked to prove statements in propositional and first-order logic. For this part of the assignment use only the following proof methods: `rule`, `rule_tac`, `drule`, `drule_tac`, `erule`, `erule_tac`, `frule`, `frule_tac` and `assumption`. You are also restricted to using only the following introduction and elimination rules: `exI`, `exE`, `allI`, `allE`, `spec`, `conjI`, `conjE`, `ccontr`, `notI`, `notE`, `notnotD`, `impI`, `impE`, `mp`, `iffI`, `iffE`, `iffD1`, `iffD2`, `disjI1`, `disjI2` and `disjE`. You must use the usual procedural style (a sequence of rule applications).

You should also be aware of the tactic `cut_tac`, which inserts a known rule or fact as an assumption in your proof. For example, the known fact:

```
excluded_middle: ¬P ∨ P
```

can be inserted as an assumption in your proof by using the command

```
apply (cut_tac excluded_middle)
```

If you wish to rename the variable P, to A say, then you can simply give the command

```
apply (cut_tac P=A in excluded_middle)
```

If you are struggling to mechanize a lemma or theorem in Isabelle, then the command `sorry` can be used. This allows the lemma or theorem to be asserted as true without completing the proof. It will enable you to make progress in the practical, however no marks will be allocated for the missing part of the proof. You should **not** use other people's proofs or formalisations.

Problem 1. (3 marks)

Prove the following statements:

1. `contrapos`:
   $P \longrightarrow Q \Longrightarrow \neg Q \longrightarrow \neg P$ **(1 mark)**

2. `flowers_knights`:
   $((\exists x.Fx) \longrightarrow (\forall x.Gx)) \longrightarrow (\forall xy.Fx \longrightarrow Gy)$ **(2 marks)**

For the next problems, you are also allowed to use any rules you proved in the previous problem. You are also allowed to prove more rules, if you think that they would be helpful.

Note: In what follows, to use a rule, you first have to give it a name. For example, if you have proved a statement `myrule`: $P \vdash Q$, and later you find $A \vdash Q$ as a goal in the proof of another theorem, writing `apply (rule myrule)` will yield $A \vdash P$ as a new goal.

Problem 2. (7 marks)

There are three chests in a room: one is gold, one is silver and one is lead. At least one of the boxes contains a portrait of Portia. Each box has an inscription.

Box	Inscription
Gold	$G \longrightarrow False$
Silver	$S \longrightarrow \neg(S \vee G)$
Lead	$L \longrightarrow (L \longrightarrow L)$

Supposing that G means 'The gold box contains the portrait', S means 'The silver box contains the portrait', L means 'The lead box contains the portrait' and that the inscriptions are true, find which boxes contain the portrait and which do not. Writing your claim as a conjunction and the inscriptions as assumptions, prove your claim.

Hint: You are also allowed the assumption $G \vee S \vee L$.

Knights and knaves problems [30%]

In the following problems you may use the method `case_tac` *if you find this helpful.*

In 'Knights and Knaves' puzzles, knights always tell the truth, and knaves always lie. We also assume that we are on an island inhabited by only knights and knaves (i.e. a person is a knave iff he is not a knight). The rest of the problems in this section are formalised in a locale which contains these basic facts about knights and knaves.

```
locale knights_knaves =
   fixes V :: "'a ⇒ bool"
   fixes G :: "'a ⇒ bool"
   fixes S :: "nat ⇒ 'a ⇒ bool"
   assumes
     V_iff_not_G: "∀ x. V x ⟷ G x"
   and
     V_imp_not_S:"∀x. ∀y. V x ⟶ S y x"
   and
     G_imp_S:"∀x. ∀y. G x ⟶ S y x"
```

We formalise 'a is a knave' as $V\ a$ and 'a is a knight' as $G\ a$. 'Person a says statement P' is formalised as $S\ n\ a = P$, where n is some natural number. We index the statement by n because one person may make more than one statement. If person a makes two separate statements, P and Q, it is *not* correct to write $S\ 1\ a = P \wedge Q$ since it is possible that $P \wedge Q$ could be false, but that P on its own could be true. This could change whether someone is a knight or a knave. We are also assuming that the domain of the quantifiers is all the inhabitants of the island (so you, as a visitor to the island, are not included).

Problem 3. (4 marks)

Prove the following lemmas:

```
lemma S_imp_G:"∀x. ∀y. S y x ⟶ G x"
lemma not_S_imp_V:"∀x. ∀y. ¬ S y x ⟶ V x"
```

These say that if someone is telling the truth they are a knight, and that if someone is telling a lie they are a knave. These lemmas will be useful to you in the next couple of problems.

Problem 4. (6 marks)

Consider the following knights and knaves puzzle:

You meet two inhabitants: Zoey and Mel. Zoey tells you that Mel is a knave. Mel says, 'Neither Zoey nor I are knaves.' [1].

Who is a knight and who is a knave? The answer has been formalised for you in the lemma `Mel_and_Zoey`: prove this lemma.

Problem 5. (20 marks)

You have a conversation with two inhabitants: Abel and Beatrice. You remark that there are lots of beautiful flowers on the island. Abel says 'If there is someone who likes flowers, then everyone is a knight'. Beatrice replies 'It is not the case that for all x and for all y, if x likes flowers then y is a knight'. She then tells you 'Abel and I are knights'. Finally Abel says confidentially to you 'Beatrice is a knave'. Who is a knight and who is a knave? Which of Abel and Beatrice, if either, likes flowers? Write your claim as a single conjunction (plus some assumptions) and then prove it. Use syntax which matches the locale: this way you will be able to use the facts that the locale provides you with.

Part 2: Geometry with order and signed areas [60%]

In this part of the assignment you are asked to prove some more complex theorems of geometry. For this purpose, you are given access to some of Isabelle's more powerful reasoning tools. Specifically, you can use the tactics: `subst`, `auto`, `simp`, `blast`, `fast`, `force`, `fastforce`, `presburger`, `algebra`, `arith`, `linarith`, `unfold`, as well as all the methods from Part 1. Moreover, now you **must** write proofs in Isabelle's declarative (structured proof) style:

```
lemma x: ...
proof ...
  assume ...
  ...
  show ...
qed
```

For this part you are **not** allowed to use the tactics `metis`, `meson` or `smt`,

unless explicitly stated. You can use the tools `sledgehammer`, `try` and `try0` to suggest methods, but take into account that if they suggest the use of `metis`, `meson` or `smt` you should then find an alternative proof.

As a general hint, consider that every theorem that you prove can be used as a lemma for constructing the proof of another theorem. Moreover, if you get stuck in one proof you can skip it by writing the command `sorry` (in place of proof methods). Naturally, you cannot get full marks for a proof that contains `sorry`. However, you can get partial marks for it, and it allows you to move on with the rest of the assignment. Moreover, `sorry` can be used so that you can `pretend` that you have completed the proof. In other words, `sorry` is like an (invalid) inference rule that allows you to prove **anything**. Therefore, if you use `sorry` in the proof of one lemma you can use that lemma afterwards in the proof of something else, as if it was a proven fact. Thus, you must be very careful not to use sorry in a **false** statement.

Geometry with ordered points [14%]

We can define a simple geometry in terms of an order on points. A locale called `Points` is declared, with the 'points' having type `'p` and fixing a constant 'order'[1]:

```
locale points =
  fixes order :: "'p ⇒ 'p ⇒ 'p ⇒ bool"
  assumes order_CBA: "order A B C ⟹ order C B A"
      and order_notBCA: "order A B C ⟹
                                    ¬ order B C A"
      and order_distinctAC:"order A B C ⟹ A ≠ C"
```

Problem 6. (3 marks)

Prove that if the points A, B and C are in the order ABC then A is distinct from B and B is distinct from C.

[1] As you may notice, these three axioms are not everything we will be using. In fact, we inherit everything from Isabelle's `Main` and `Real` theories. This background allows us to use various lemmas (e.g., those that `sledgehammer` may suggest), and makes tactics like `simp` and `auto` quite powerful. You can search for relevant theorems in the 'Query' box next to 'Output'.

We can now define lines in terms of `order`:

```
definition "A ≠ B ⟹ line A B = {X. X=A ∨ X=B ∨
  order A B X ∨ order A X B ∨ order X A B}"

definition "Lines = {l. ∃ C D. l = line C D}"
```

Note that `line` is defined using a conditional definition, i.e. it is undefined unless you can prove that its arguments are distinct. The second definition `Lines` is of the set of all lines.

Problem 7. (6 marks)

We have now defined a locale which imports the `points` locale; this allows us to use every theorem of the `points` locale in our new locale. We have three axioms already in the locale[2]:

```
locale lines = points order
  for order :: "'p ⇒ 'p ⇒ 'p ⇒ bool" +
  assumes A_V:"A ≠ B ⟹ ∃C. order A B C"
  and A_VI:"⟦C ∈ line A B; D ∈ line A B; C ≠ D⟧ ⟹ A
    ∈ line C D"
  and unique_line:"A≠B ⟹ ∃!l∈Lines. A ∈ l ∧ B ∈ l"
```

Formalise the two axioms below and add them to the locale[3]. You must use `line` and `Lines` (defined above) in your formalisation of A VIII:

A VII: *There exist three distinct points A, B and C not in any of the orders ABC, BCA or CAB.*

A VIII: *If three distinct points A, B, C do not lie on the same line and D and E are two points in the order BCD and CEA, then a point F exists in the order AFB and such that D, E, F lie on the same line.*

Problem 8. (5 marks)

Formalise and prove that, given a line, there is a point not on the line

[2]The axiom `unique_line` could be proven from the others, so this is not a minimal set of axioms.

[3]Be very, very careful with the way that you formalise these axioms. The wrong axiomatisation can make the locale too weak (it does not entail certain necessary properties), too strong (it entails more than we are asking), inefficient (it is difficult to reason with it) or outright inconsistent (it entails a contradiction!).

(see Theorem 5 on page 355 of Veblen's paper [2]).

Triangle Geometry [26%]

We will now consider what at first sight appears to be a rather different sort of geometry. The axioms of this geometry are formalised in the locale `triangles`.

```
locale triangles =
  fixes Δ :: "'a ⇒ 'a ⇒ 'a ⇒ real"
  assumes axiom0_a : "Δ x y z = Δ y z x"
    and   axiom0_b : " - Δ z y x = Δ x y z"
    and   axiom2 :
    "x ≠ y ⟹ ∃z. (R::real) = Δ x y z"
    and   axiom3_a :
    "Δ x y z + Δ h z y + Δ z h x + Δ y x h = 0"
    and   axiom5 : "Δ x y z = 0 ⟹
    (Δ h x y)*(Δ k x z) = (Δ k x y)*(Δ h x z)"
```

We can think of the constant Δ as the signed area of the triangle defined by the 'points' which are its arguments. The signed area of a triangle is just the area of that triangle, multiplied by -1 if the points of that triangle are traversed clockwise, and by 1 otherwise. Now we can understand what the geometric meanings of `axiom0_a` and `axiom0_b` are. As Figure 1, shows, listing the points in the same order, even if we begin with a different point, just describes the same triangle, and listing the points in reverse order describes traversing the points in the opposite direction, which means that we must change the sign of the area, by definition. The other axioms also have geometric meanings and figuring these out may help in some of the following problems.

Problem 9. (16 marks)

In the new locale, the following lemmas are formalised for you. Give structured proofs for `axiom1` and `axiom6` (see pages 1 and 5 respectively of Dijkstra's note [3]).

```
 lemma reverse_order: "- Δ y x z = Δ x y z"
"- Δ x z y = Δ x y z" " - Δ z y x = Δ x y z"
```

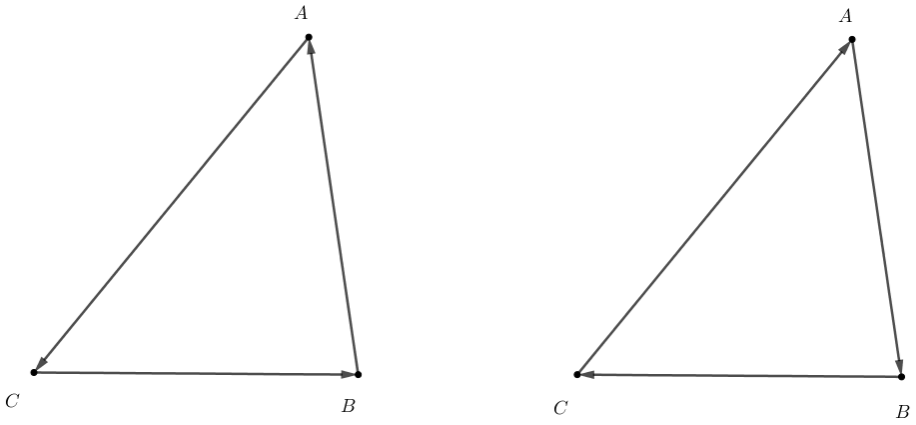

Figure 1: `axiom0_a`: $\Delta ABC = \Delta BCA$ and `axiom0_b`:$\Delta ABC = -\Delta CBA$

```
lemma same_order:
"Δ x y z = Δ z x y" "Δ x y z = Δ y z x"

lemma order_eq_zero: assumes "Δ x y z = 0"
 shows "Δ y x z = 0"  and "Δ x z y = 0"
   and "Δ y z x = 0"  and "Δ z y x = 0"
   and "Δ z x y = 0"  and "-Δ x y z = 0"
   and "-Δ y x z = 0" and "-Δ x z y = 0"
   and "-Δ y z x = 0" and "-Δ z y x = 0"
   and "-Δ z x y = 0"

lemma pos_order_neq_zero:
 assumes "Δ x y z ≠ 0"
   shows "Δ y x z ≠ 0"  and "Δ x z y ≠ 0"
     and "Δ y z x ≠ 0"  and "Δ z y x ≠ 0"
     and "Δ z x y ≠ 0"

lemma axiom1: assumes "x = y" shows "Δ x y z = 0"

lemma axiom3_b:
"Δ x y z = Δ h y z + Δ x h z + Δ x y h"
```

```
lemma axiom3_c:
"Δ x h y + Δ y k x = Δ h y k + Δ k x h"

lemma lemma4: assumes "Δ x y z = 0"
shows "Δ x h y + Δ y h z = Δ x h z"

lemma two_points: "Δ x x y = 0"
"Δ x y y = 0" "Δ x y x = 0"

lemma a_b_distinct: assumes "Δ a b c ≠ 0"
shows " a ≠ b"

lemma axiom6: assumes "Δ x y z = 0" and "x ≠ z"
shows " ∃L. ∀h. Δ h x y = L * Δ h x z"
```

Hint for proving axiom6: You wish to show $\exists L.\forall h.\Delta hxy = L * \Delta hxz$. One way to do this is to define such an L that works for a single value of h, call this q, and then show it works for all h. To work out a definition for L, try interpreting the theorem geometrically: it may be related to the theorem that the ratio of areas of two triangles of equal altitudes is equal to the ratio of their bases (this follows from the 'half base times height' formula for the area of a triangle).

For all of the following proofs you can use any method except `smt`. *You may also use any previously proven Isabelle lemmas in the theory or its imports.*

Problem 10. (10 marks)

So far we have used the idea of Δ corresponding to signed area to allow us to geometrically visualise the lemmas and proofs in our locale. We will now use this correspondence to instantiate the `triangles` locale to points considered as coordinates in $\mathbb{R}^2$:

```
type_synonym point = "(real*real)"

definition xCoord :: "point ⇒ real"
  where "xCoord P = fst P"

definition yCoord :: "point ⇒ real"
  where "yCoord P = snd P"
```

```
definition signedArea :: "[point, point, point] ⇒
   real"
where "signedArea a b c = (1/2) *
((xCoord b - xCoord a)*(yCoord c - yCoord a)
- (yCoord b - yCoord a)*(xCoord c - xCoord a) )"
```

You are given proofs that all the axioms of the `triangles` locale, except `axiom5`, are satisfied by considering Δ to be `signedArea`. Formulate `axiom5` with Δ as `signedArea`. Name it `signedArea_5` and prove it. Then use the command `interpretation` to instantiate the `triangles` locale so that Δ corresponds to `signedArea`.

Challenge - defining ordered points using the triangle geometry [20%]

Problem 11. (20 marks)

Finally we come to the connection between the two apparently different geometries we have been formalising. A new locale `triangles_continuum_pt` which extends the previous `triangles` locale by one axiom has been defined.

```
locale triangles_continuum_pt = triangles +
  assumes "∃ (a::'a) b. a ≠ b"
```

- Within this locale, give a definition of ordered triples of collinear points in terms of Δ and any other notions that you find helpful to define. E.g. notice that three points are collinear if and only if the area of the triangle they form is zero.
- Now use your definition to instantiate the `points` locale. The lines defined in the `triangles` locale should agree with those defined in your instantiation of the `points` locale[4], although you do not have to prove this.

Hint: Consider three collinear points a, b, c. Then take a point p not on the line: the four points will then define some triangles of nonzero area. Examine what changes in the areas of these triangles for different orderings of a, b,

[4] at least when the points defining the line are distinct

c. Remember to draw plenty of diagrams (but you do not have to submit them)!

Marking

You will receive partial credit for any incomplete attempt that is related to the correct formalisation.

Demonstrator Hours

The demonstrator, Imogen Morris (s1402592@sms.ed.ac.uk), will be available to give advice on Mondays from 9am-11am in 5.05 West Lab, Appleton Tower, or any time by email and on Piazza.

Submission

By 4pm on 20th November 2017 you must submit your solution in electronic form. This should consist of your theory file `Practical.thy` and can be submitted using the command:

```
submit ar cw1 Practical.thy
```

Late coursework will be penalised in accordance with the Informatics standard policy (see `http://edin.ac/1LRblYG`). Please consult your course guide for specific information about this. Also note that, while we encourage students to discuss the practical among themselves, we take plagarism **seriously** and any suspected case will be treated appropriately. Please consult your student guide for more information about this matter.

References

[1] Puzzle from Critical thinking web, developed by Philosophy Department, University of Hong Kong. The puzzle was generated by a computer program written by Zachary Ernst. You can find more of them at: `http://philosophy.hku.hk/think/logic/knights.php`.

[2] O. Veblen. 'A System of Axioms for Geometry'. In: *Trans. Amer. Math. Soc.* 5 (1904), pp. 343-384. Available at: `http://edin.ac/2ykyo1f`.

[3] E. W. Dijkstra. 'A Tough Experiment with the Triangle Calculus'. In:*E. W. Dijkstra Archive*. Available at: `http://edin.ac/2ggU7Ay`