UNIVERSITY OF CAMBRIDGE

Heterogeneous Proofs

Spider Diagrams meet Higher-Order Provers

Matej Urbas and Mateja Jamnik

{mu232,mj201}@cl.cam.ac.uk

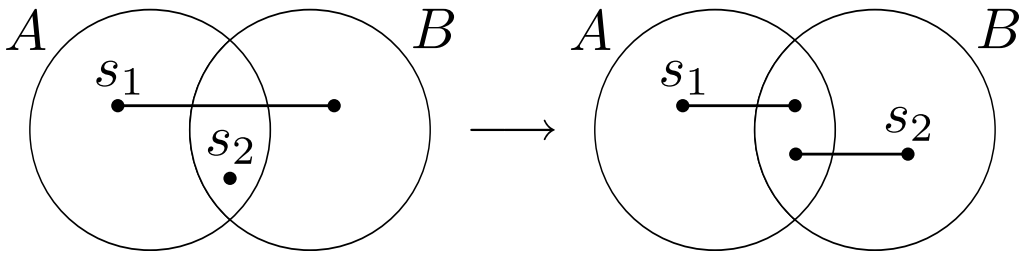
Heterogeneous reasoning?

Heterogeneous reasoning is...

*...reasoning about **diagrammatic** and **sentential** propositions with **mixed** diagrammatic and sentential **inference steps**.*

In other words:

- two languages (one of them is diagrammatic),
- two sets of inference rules, and
- a bi-directional translation between the two.



$$(\exists\, s\ s'.\ s \neq s' \wedge s \in A \cap B \wedge s' \in (A - B) \cup (B - A)) \longrightarrow$$
$$(\exists\, s\ s'.\ s \neq s' \wedge s \in A \wedge s' \in B)$$

Hypothesis

Main hypothesis:

Interactive heterogeneous reasoning is feasible, and it can be done **formally** with **automated verification** of proofs (i.e.: with proof reconstruction).

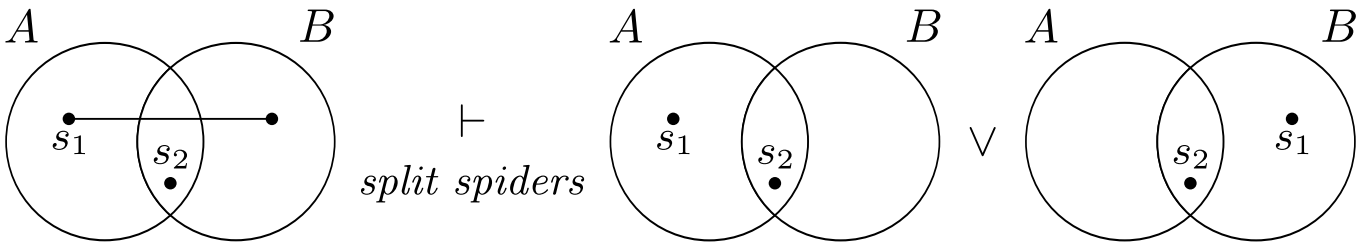
We also hope to show that:

- HR can, for specific domains, produce more concise or intuitive formulae and proofs (we currently target MFOL).
- HR enables easier extensions to the diagrammatic logic (by adding new diagrammatic language elements and automatically importing or formalising new inference rules).

Heterogeneous framework

The diagrammatic language: **Spider Diagrams**

- A well-defined language on MFOL with a set of sound and complete inference rules.



- We are developing a diagrammatic reasoner for spider diagrams, called **Speedith** (sources at `http://gitorious.net/speedith`).

Heterogeneous framework

The sentential reasoner: **Isabelle**

- Formalisation of spider diagrams in Isabelle/HOL.

```
lemma sd_rule_split_spiders:
  "⟦ habs = (h#hs); habA ∪ habB = h ⟧ ⟹ sd_sem (PrimarySD habs shzs) =
   (sd_sem (PrimarySD (habA#hs) shzs) ∨ sd_sem (PrimarySD (habB#hs) shzs))"
```

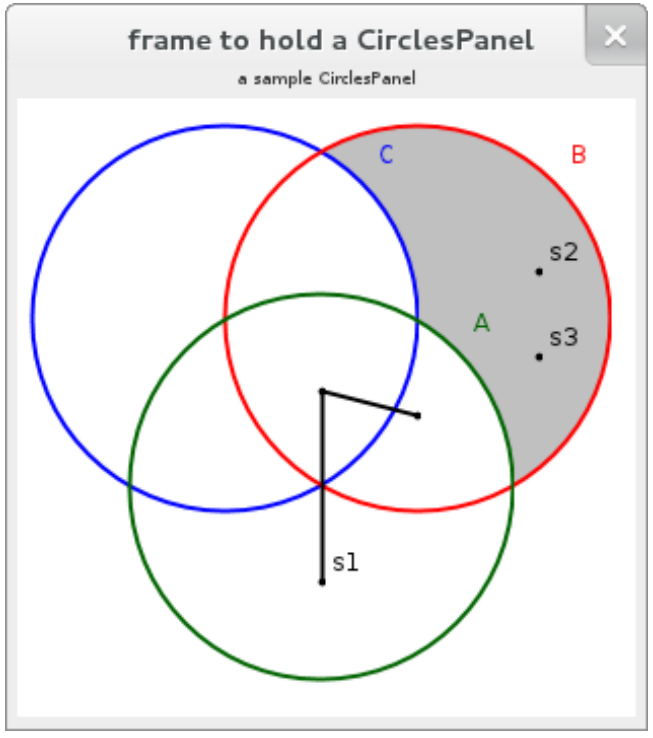
- ML-level translation procedures.
- Invocation of Speedith through custom tactics.

```
(* This lemma should land in the unit tests. *)
lemma testA: "(∃s1 s2. distinct[s1, s2] ∧ s1 ∈ A ∩ B ∧ s2 ∈ (A - B) ∪ (B - A))
              ⟶ (∃s1 s2. distinct[s1, s2] ∧ s1 ∈ A ∧ s2 ∈ B)"
  apply (sd_tac split_spiders sdi: 1 sp: "s2" r: "[([\"A\"],[\"B\"])]")
  apply (sd_tac add_feet sdi: 3 sp: "s2" r: "[([\"A\", \"B\"],[])]")
  apply (sd_tac add_feet sdi: 3 sp: "s1" r: "[([\"A\"],[\"B\"])]")
  apply (sd_tac add_feet sdi: 2 sp: "s2" r: "[([\"A\", \"B\"],[])]")
  apply (sd_tac add_feet sdi: 2 sp: "s1" r: "[([\"B\"],[\"A\"])]")
  apply (sd_tac idempotency sdi: 1)
  by auto
```

Heterogeneous framework

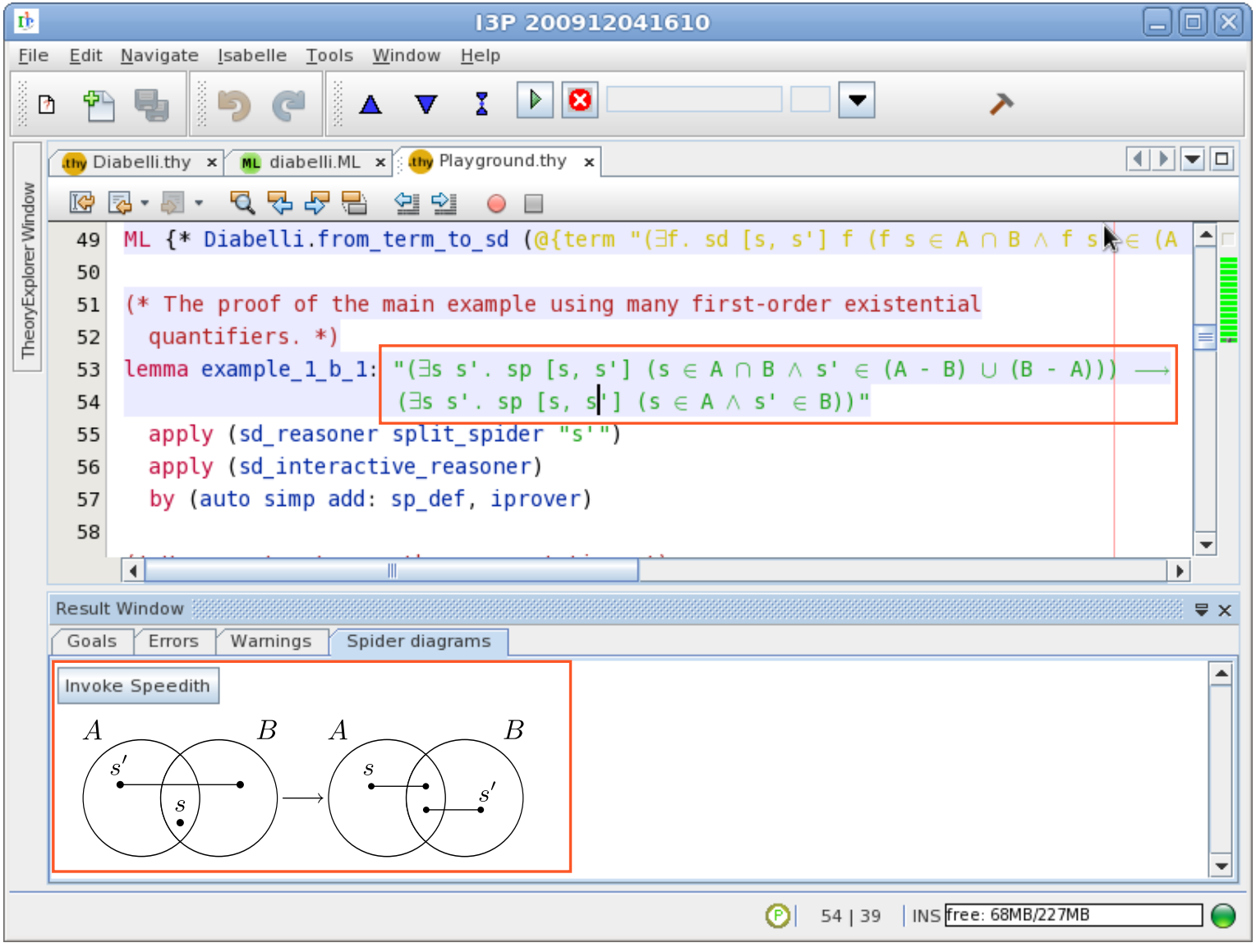
TODO: GUI integration

- Graphical interactive input and inference rule selection.
- Visualisation of spider diagrams with *iCircles* by Stapleton and Flower.



How should it work?

Something like this:



Thank you.

UNIVERSITY OF CAMBRIDGE

Heterogeneous Proofs: Spider Diagrams meet Higher-Order Provers

Matej Urbas
Matej.Urbas@cam.ac.uk

Mateja Jamnik
Mateja.Jamnik@cam.ac.uk

Resources:

- Speedith:

  `http://gitorious.net/speedith`

- iCircles:

  `https://gitorious.org/speedith/inductive_circles`