D6.3
Library of examples in the heterogeneous KR&R system and evaluation of the mathematics case study

Authors	Ewen Maclean, Alan Smaill
Reviewers	Marco Schorlemmer

Grant agreement no.	611553
Project acronym	COINVENT - Concept Invention Theory
Date	October 1, 2016
Distribution	PU

Disclaimer

The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553.

Abstract

This deliverable indicates how to access the mathematical examples developed in the COINVENT project, and points to the Keyword list: **mathematical concept blending**, **generic representation**

Executive Summary

- There is a library of examples of mathematical blends developed in the COINVENT project. This document briefly describes how to access this resource.
- A case study was carried out as part of the Coinvent project on the domain of formal reasoning, to evalate the usefulness of the ideas in a situation that occurs in mathematical practice. The outcome is reported.

Contents

1 Introduction

This deliverable has two purposes:

- To provide a publicly accessible library of examples of COINVENT conceptual blending in mathematics and related areas;
- To report on evaluation of the mathematical case study that was carried out during the project. This involved structured interaction with university students with a background in machine-assisted proof, asked to come up with a "eureka" lemma in the course of proving properties of some data-type. Here analogical inference from a developed theory of a different data-type is possible, and the usefulness of COINVENT blending can be assessed.

1.1 Structure of this Document

In Section 2 we describe how to access our mathematical examples. Section 3 describes the case study.

2 Library of mathematical examples

We outline here how to access various formalisations of mathematical domains that have been developed in the course of the COINVENT project.

These are accessible either via the ontohub repository[1] or via github, as indicated.

Complex numbers The development of a theory of the complex numbers, by blending the theory of a field with that of a 2-dimensional vector space.

`https://ontohub.org/complex-blend`

Blends for analogous lemmas Blends between related theories, where one has a more developed set of proved theorems that the other.

`https://ontohub.org/lemma-examples`

From ICCC 2015 Various examples worked out for the paper on comceptual blending in mathematics, ICCC2015, including

- Integers as blend of natural numbers and a single invertible function.
- Blending finite and infinite processes (2 BMI blends)
- Prime ideals over CDRs

`https://github.com/ewenmaclean/ICCC2015_hetsfiles`

Galois Theory Some of the blends in the development of Galois theory by Danny de Jesus:

`https://ontohub.org/algebrablending`

The full set of base theories involved is at:

`https://github.com/dgomezramire/FieldsGaloisBlendingGeneration`

Further examples similar work are are collected, alongside other blends, in

`https://ontohub.org/conceptportal`

in particular, examples of transitive divisibility relations, and rings based on Gaoldbach's conjecture.

[1] `https://ontohub.org`

Transfer of spatial descriptions to other domains This work from Daniel Worthing show analogical transfer treated via conceotual blending.

`https://ontohub.org/repositories/daniel-worthing-examples`
`https://ontohub.org/dan-examples`

2.1 Future work

We will consolidate the examples above and other still in development in the ontohub repository, to allow uniform access to this aspect of the work in the COINVENT project.

3 Mathematical case study

3.1 Introduction

The case study was carried out looking at a problem which arises in proof of properties in theories involving inductively defined data-types, such as natural numbers (with successor), lists, trees end so on).

In such theories, proof by induction is typically performed by picking a variable and using induction on that variable. Schematically, for the natural numbers, induction allows inferences of the shape:

$$\frac{P(0) \quad \forall x\, (P(x) \to P(s(x)))}{\forall y\, P(y)}$$

Sometimes, just using induction in the form is not sufficient; for example

$$\forall x\, \forall y\, x + (x + x) = (x + x) + x$$

turns out to be problematic; instead the apparently stronger statement $x + (y + z) = (x + y) + z$ should replace the original goal — we take this to be the "creative" step in the proof (in the literature, the term "eureka lemma" is used.

The task was to see if proofs making use of a creative lemma in one domain could help in finding proofs in a different mathematical domain.

3.2 Experimental set-up

The 19 subjects had experience in mathematical reasoning at at least later stages of undergraduate study (they were mostly undergraduate students of Informatics). They also had experience of machine-assisted proof. They were aware of the issue of creative lemmas, through a lecture on the topic in the Automated Reasoning course at Edinburgh University if they had not seen this before,

The information given to the subjects was as follows.

> **Instructions**
>
> This experiment is designed to show the user some lemmas which are used to prove certain tricky theorems. It is best viewed on a chrome browser if possible. The theorems in question are referred to as source theorems. These theorems are provable, but only with the assistance of the creative lemma given by the source lemma.
>
> The system will show you in one area of the screen a simple theory expressed in simply typed first order logic (CASL to be precise) labelled "Source Theory", and underneath the source theorem and lemma. Both the source theorem and source lemma are known to be provable, and source lemma is known to be a valid generalisation which facilitates the proof of the source theorem.
>
> In the area of the screen labelled "Target Theory", another theory is shown. A new theorem is stated called the target theorem. This theorem is also known to be provable, but there is initially no suggestion given as to what a creative lemma may be

which would facilitate the proof of the theorem. Once you press the button labelled "Generate Analogous Lemma", a suggestion will be given in the box labelled "Target Lemma". There may be a target lemma given (you can resize the box if necessary), or the box may inform you that no target lemma has been invented.

You are then presented with a set of questions depending on whether a suggestion was given:

Is the target lemma true?	Please answer yes, no or don't know
Do you think this target lemma will help prove the target theorem?	Please answer yes, no or don't know.
Do you have a better suggestion for a target lemma?	If you have a good idea for a creative lemma to help prove the target theorem please enter it.
Do you think this target lemma will help prove the target theorem?	Please answer yes, no or don't know.
Do you have a better suggestion for a target lemma?	If you have a good idea for a creative lemma to help prove the target theorem please enter it.
Was the system helpful?	Do you think the target lemma given would help you to formulate a useful lemma for the theory? i.e. do you think it was better than having to invent a target lemma for yourself?
Was the target lemma creative?	Do you think inventing the target lemma displayed creativity?
What is your suggestion?	In the event no target lemma is given you are invited to provide a suggestion for a target lemma if you can think of one.

For each subject, 5 pairs of examples were chosen in random order, and suggestions were generated or not also at random. One of the suggestions generated by the system is in fact wrong.

The experimental interface is at

`http://server.coinvent-project.eu/static/experiment_unsupervised`

and the raw data of results is at

`http://server.coinvent-project.eu/static/experiment_results`

3.3 Simple example

A simple example that is in the test suite involves comparison between natural number arithmetic (with plus, etc) and list operations (with append). This example is reported in Eppe et al. (2015).

A theory of natural numbers, with exponentiation defined has a quick version of exponentiation `qexp` defined, together with a proof that it does define exponentiation; this proof requires a eureka lemma.

The related theory of lists is given in less developed form, along with a quick reverse function `qrev`. The task is to show that `qrev` does indeed compute the reverse; here the eureka lemma is absent.

The two theories are shown below:

spec NAT =
  sort *Nat*
  ops $zero : Nat$;
    $s : Nat \to Nat$;
    $plus : Nat \times Nat \to Nat$;
    $sum : Nat \to Nat$;
    $qsum : Nat \times Nat \to Nat$
  $\forall x, y : Nat$
  • $sum(zero) = zero$
  • $sum(s(x)) = plus(s(x), sum(x))$
  • $qsum(s(x), y) = qsum(x, plus(s(x), y))$
  • $qsum(zero, y) = sum(y)$
  • $plus(zero, y) = y$
  • $plus(s(x), y) = s(plus(x, y))$
  • $sum(x) = qsum(x, zero)$
  %%**proved_theorem**%%
  • $plus(sum(x), y) = qsum(x, y)$
  %%**eureka_lemma**%%
end

spec LIST =
  sorts *El*, *L*
  ops $nil : L$;
    $cons : El \times L \to L$;
    $app : L \times L \to L$;
    $rev : L \to L$;
    $qrev : L \times L \to L$
  $\forall x, y : L; h : El$
  • $rev(nil) = nil$
  • $rev(cons(h, x)) = app(rev(x), cons(h, nil))$
  • $qrev(nil, y) = y$
  • $qrev(cons(h, x), y) = qrev(x, cons(h, y))$
  • $app(nil, y) = y$
  • $app(cons(h, x), y) = cons(h, app(x, y))$
  • $rev(x) = qrev(x, nil)$
  %%**To_prove**%%
end

Here in the theory NAT, it is assumed that there is a proof of the **proved theorem** using the **eureka lemma**. In the theory LIST, a proof is sought for **to prove**, and the eureka lemma (correctly) suggested by the system is:

$$\forall x, y : L; h : El \bullet app(rev(x), y,) = qrev(x, y)$$

3.4 Results summary

- In the easier examples, system suggestions were rated as helpful, and usually creative.
- In harder examples, system suggestions were rated as helpful only if accepted, but rated as creative sometimes even if not accepted.
- Over all examples, in the absence of system suggestions, only a few subjects explicitly generated their own suggestions (4/19)
- Many subjects failed to answer the full set of questions, including who provided no input at all.

While the responses are relatively encouraging for the minority who were able to engage seriously with the experiment, the significant number of subjects who found the task too hard means that it is not possible to draw any firm conclusion from the results.

Two possible explanations are suggested:
one is that the syntax of CASL in which the theories were provided was an obstacle to making sense of these theories; another is that the problem area is too remote from the experience of the subjects for then to make judgements they felt comfortable with,

We believe that the syntax, while not part of the curriculum, is close enough to the sort of logical syntax that is current in the area that in itself this is unlikely to be the main source of the problem.

3.5 Future work

Clearly, we need more solid evidence in order to claim that conceptual blending in mathematics can serve as an aid to creativity in mathematics.

We envisage experiments along similar lines as the one carried out, but where the match between the subjects' expertise and the problem are is better. In addition, we would like to see how these ideas play out at different levels of mathematical experience, for example in early schooling, where there are many opportunities provided the right sort of interface can be designed.

References

Eppe, Manfred et al. (2015). 'ASP, Amalgamation and the Conceptual Blending Workflow'. In: *Logic Programming and Nonmonotonic Reasoning: 13th International Conference, LPNMR 2015*. Ed. by Francesco Calimeri, Giovambattista Ianni, and Miroslaw Truszczynski. Vol. 9345. Lecture Notes in Artificial Intelligence. Springer.