

From second-order logic into planning

Storyboard proposed for ICAPS 2011 System Demonstration

Aldo Porco

Universidad Simón Bolívar
Caracas, Venezuela
aldo@gia.usb.ve

Alejandro Machado

Universidad Simón Bolívar
Caracas, Venezuela
alejandromachado@gia.usb.ve

Blai Bonet

Universidad Simón Bolívar
Caracas, Venezuela
bonet@ldc.usb.ve

Description

We have designed a minimalistic web application that prompts the user for a second-order existential formula, a signature and a finite structure; and automatically translates the input to a planning task, showing the user the equivalent STRIPS problem expressed using the Planning Domain Definition Language.

Introduction

We will start the demonstration providing an introduction to our modelling language based on second-order logic, and briefly guiding the user step-by-step through the modelling process using previously tested examples of well-known NP-complete problems, e.g. directed hamiltonian path or three-colorability of graphs.

Let us consider, for example, the modelling process of the boolean satisfiability problem. Let us define the relation $(?P \ ?x \ ?y)$, which means the variable $?x$ appears as positive in the clause $?y$, and the reciprocal $(?N \ ?x \ ?y)$, that means the variable $?x$ appears as negative in the clause $?y$.

Now, we need to devise a second-order formula that captures SAT. This problem consists in finding which variables should be assigned to **true** in order to satisfy a CNF formula. In **every** clause **at least one** variable is satisfied. A variable $?x$ is *satisfied* inside a clause $?y$ if its truth value $?T$ is coherent with its sign: if it appears as positive, $(?T \ ?x)$ should be true; if it appears as negative, $(?T \ ?x)$ should be false. Thus, we show the corresponding formula below.

```
(so-exists (?T 1)
  (forall (?y) (exists (?x)
    (or
      (and (?P ?x ?y)
        (?T ?x))
      (and (?N ?x ?y)
        (not (?T ?x)))
    )
  )
))
```

Let us consider this sample problem instance for SAT:

$$(p \vee \neg q \vee r) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee q)$$

Since our translation function always deals with finite domains with objects ranging from `zero` to `max`, an adequate representation for this problem in a first-order structure is described in the following paragraph.

p is the constant `zero`, q is the constant `obj1`, r is the constant `max`; and the first, second and third clauses are represented by `zero`, `obj1`, and `max`, respectively. Considering these facts, the sample problem instance is encoded as:

```
(?P zero zero) (?N obj1 zero) (?P max zero)
(?N zero obj1) (?N max obj1)
(?N zero max) (?P obj1 max)
```

In this fashion, we intend to explain attendees how to model SAT and a graph NP-complete problem in our logical language.

Live translation

Next, we will instruct the user to input the second-order formulas, signatures and first-order structures modelled in the previous step in our web application.

When we submit the information and get the translated response in PDDL, we will explain how the translation was carried out, outlining our simple parsing algorithm and describing the different elements present in our reduction. A demonstrational video that shows how the SAT formula is translated into STRIPS PDDL is available at this address¹.

Automated problem solving

After the basics of the translation have been made clear, our web application will invoke the M planner, by Jussi Rintanen, to solve the problem that has been translated.

The output of the planner will be shown to the attendees, along with a simple, graphical interpretation of the solution.

Conclusion

We will end the demonstration with a short summary of our contributions and experimental results, as well as a discussion of the challenges that lie ahead for translating general PSPACE problems formulated as second-order logic with transitive closure into STRIPS.

¹http://is.gd/icaps11_logic2planning