Logical reasoning and programming, lab session XII

(December 16, 2019)

XII.1 Produce all the possible paramodulants, but do not perform paramodulations into variables, of

$$\{\{p(X), \neg q(X,Y), f(c,Y) = g(X)\}, \{p(Z), q(g(a), f(Z,b)), c = f(c,c)\}\}.$$

For simplicity, we use $=$ instead of $\approx$ here.

XII.2 Formulate the following problems in the TPTP language and (dis)prove them using the E prover. Assuming the following group axioms

$$\begin{aligned} &e \cdot X = X, \\ &X^{-1} \cdot X = e, \\ &(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z) \end{aligned}$$

your task is to (dis)prove

(a) $X \cdot e = X$,
(b) $X \cdot X^{-1} = e$,
(c) $X \cdot Y = Y \cdot X$,
(d) $X \cdot Y = Y^{-1} \cdot X^{-1}$.

XII.3 Use the model finder Paradox to produce counterexamples for unprovable claims in the previous exercise (**XII.2**).

XII.4 Formalize in the TPTP format a simple example with the following axioms

$$\begin{aligned} &\forall X \neg r(X,X), \\ &\forall X \forall Y \forall Z (r(X,Y) \wedge r(Y,Z) \rightarrow r(X,Z)), \\ &\forall X \exists Y r(X,Y) \end{aligned}$$

and check how fast can Paradox generate possible finite models for this simple problem. Clearly, it will never find a model, because the problem has only infinite models.

XII.5 Try the Vampire prover on the problem GRP140-1 from the TPTP library. We demonstrate the effect of the limited resource strategy (LRS), which discards unprocessed clauses that will be unlikely processed in a given time limit, by this example. For the intended behavior you need a special setting—age:weight ratio is 5:1 and the forward subsumption is turned off:

```
vampire -awr 5:1 -fsr off -t 30 GRP140-1.p
```

First, try the timelimit 30s, then try 15s, 7s, ... You can try even shorter times than 1s, e.g., `-t 5d` means 5 deciseconds.

For comparison you can try the competition mode on the same problem

```
vampire --mode casc GRP140-1.p
```

XII.6 Try the E prover on the problem GRP001-1 from the TPTP library. Compare how can the use of a literal selection strategy influence the behavior of the prover:

```
eprover --literal-selection-strategy=NoSelection GRP001-1.p
eprover --literal-selection-strategy=SelectLargestNegLit \
          GRP001-1.p
```

XII.7 A notoriously hard task for humans is to prove formulae in Hilbert style proof systems. We have the following schemata of axioms

$$\varphi \to (\psi \to \varphi) \tag{1}$$
$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{2}$$
$$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi) \tag{3}$$

where φ, ψ, and χ are propositional formulae. It means that any instance of them is trivially provable. We also have a rule, called modus ponens, which says that if φ and $\varphi \to \psi$ are provable, then also ψ is provable.

We can encode this whole problem about propositional provability as a first-order problem. We can treat propositional formulae as terms in first-order logic and we can introduce a new unary predicate, say `pr`, which says that a term is provable. Then (3) can be encoded as

```
cnf(ax3, axiom, pr(i(i(n(B), n(A)), i(A, B)))).
```

where we use a binary function symbol `i` for implication and a unary function symbol `n` for negation. Similarly, we can encode (1–2) and the rule modus ponens. Now we can ask the E prover whether the following formulae are provable in our system

(a) $\varphi \to \varphi$,
(b) $\neg\neg\varphi \to \varphi$,
(c) $((\varphi \to \psi) \to \varphi) \to \varphi)$,
(d) $((\varphi \to \psi) \to \varphi) \to \varphi)$ with (3) removed,
(e) $(\neg\varphi \to \psi) \to ((\neg\varphi \to \neg\psi) \to \varphi)$,
(f) $(\neg\varphi \to \psi) \to (\neg\psi \to \varphi)$,
(g) $(\neg\varphi \to \psi) \to (\psi \to \varphi)$.

Try also `--auto-schedule` mode and if you are unable to find a proof, try to find a counterexample using Paradox.