Logical reasoning and programming, task II

(November 28, 2023)

A notoriously hard task for humans[1] is to prove formulae in Hilbert-style (also called Frege) proof systems. Despite this fact, they are quite popular among logicians for their nice theoretical properties and hence occur regularly in courses. If the rule modus ponens rings a bell, you have probably already seen such a system.

Your task is to prove some formulae in propositional Hilbert-style proof systems using first-order theorem provers; you encode a propositional proof system into first-order logic. It means that you should produce a TPTP file for each problem and use a theorem prover or a model finder to show that it is provable or not provable, respectively. The task is slightly simplified by the fact that the propositional language is restricted—the only allowed connectives are implications ($\rightarrow$) and negations ($\neg$).

Software

I recommend to use the E prover in `--auto-schedule` mode as a theorem prover; you can also print the proof using the `--proof-object=1` option (it is also possible to produce a GraphViz dot graph using the `--proof-graph` option). To run the E prover for roughly 30 seconds, use the `--cpu-limit=30` option.

If you are unable to find a proof, try to find a counter-example using Paradox (compiled version 2.3 for linux, version 2.3, experimental version 3 from author's web page); you can also print the model using the `--model` option. To run Paradox for roughly 30 seconds, use the `--time 30` option, however, it is really a very soft limit.

As a last resort, you can also use System on TPTP, where many provers are available online (including E and Paradox). An advantage is there is no need to install solver, and it is also possible to visualize your proofs using IDV easily.

Points

The task consists of three parts:

I System $\mathcal{HC}$ (5 points).

II System $\mathcal{HI}$ (5 points).

III The relation between $\mathcal{HC}$ and $\mathcal{HI}$ (5 points).

You are supposed to submit all files in an archive to BRUTE. Ideally, each part should be in a separate directory. Note that this task will be evaluated manually.

Please, do submit even incomplete solutions!

[1] In fact, it is also hard for machines as many seemingly simple problems are already algorithmically undecidable, for example, whether given schemata of axioms together with the rule modus ponens prove the same formulae as the system $\mathcal{HC}$. Hence in Part III you are asked to solve an instance of a really hard problem.

I System $\mathcal{HC}$

Problem

The system $\mathcal{HC}$ has the following three schemata of axioms

$$\varphi \to (\psi \to \varphi) \tag{C1}$$

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{C2}$$

$$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi) \tag{C3}$$

where φ, ψ, and χ are arbitrary propositional formulae. It means that any instance, which is a formula, of (C1), (C2), or (C3) is trivially provable. The system $\mathcal{HC}$ also contains the rule modus ponens that for every formulae φ and ψ says:

If φ is provable and $\varphi \to \psi$ is provable, then also ψ is provable. (MP)

Hence the rule modus ponens extends provability also to formulae that are not instances of (C1–C3). A natural question to ask is whether a given formula is provable in the system $\mathcal{HC}$. Or more generally, we can ask whether formulae of a given shape are provable in $\mathcal{HC}$.

It is possible to encode this problem, which is 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, for example `pr`, that says a term (representing a propositional formula) is provable. Then the schema of axiom (C3) can be encoded, for example, as

```
fof(c3, axiom, ![A,B]: ( pr(i(i(n(B), n(A)), i(A, B))))).
```

or

```
cnf(c3, axiom, pr(i(i(n(B), n(A)), i(A, B)))).
```

in TPTP, where we use the binary function symbol `i` for implication ($\to$) and the unary function symbol `n` for negation ($\neg$). Similarly, we can encode (C1–C2) and the rule modus ponens (MP).

Now we can use a theorem prover to show that $\varphi \to \varphi$ is provable in $\mathcal{HC}$ for every φ (in our propositional language containing only $\to$ and $\neg$). In the language of TPTP this can be encoded as

```
fof(phiphi, conjecture, ![A]: ( pr(i(A,A)))).
```

or

```
cnf(phiphi, negated_conjecture, ~pr(i(a,a))).
```

On the other hand, $\varphi \to \psi$ is not provable in $\mathcal{HC}$, for every φ and ψ, as you can show by producing a counter-example (of course, it is provable if $\psi = \varphi$).

Task

Let φ, ψ, and χ be arbitrary propositional formulae in our language containing only $\rightarrow$ and $\neg$. Decide for each of the following schemata whether it is provable in $\mathcal{HC}$ or not:

(Ia) $(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$,

(Ib) $\neg\neg\varphi \rightarrow \varphi$,

(Ic) $\varphi \rightarrow \neg\neg\varphi$,

(Id) $((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi$,

(Ie) $(\neg\varphi \rightarrow \psi) \rightarrow ((\neg\varphi \rightarrow \neg\psi) \rightarrow \varphi)$,

(If) $(\neg\varphi \rightarrow \psi) \rightarrow (\neg\psi \rightarrow \varphi)$,

(Ig) $(\neg\varphi \rightarrow \psi) \rightarrow (\psi \rightarrow \varphi)$.

Output

For each schema, upload an input file in the TPTP language that encodes the problem. Upload also the output of either a theorem prover or a model finder used on the input file that shows that the given schema is either provable or not in $\mathcal{HC}$.

II System $\mathcal{HI}$

Problem

The system $\mathcal{HI}$ has the following four schemata of axioms

$$\varphi \rightarrow (\psi \rightarrow \varphi) \tag{I1}$$
$$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi)) \tag{I2}$$
$$(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi) \tag{I3}$$
$$\varphi \rightarrow (\neg\varphi \rightarrow \psi) \tag{I4}$$

where φ, ψ, and χ are arbitrary propositional formulae. It means that any instance, which is a formula, of (I1–I4) is trivially provable. The system $\mathcal{HI}$ also contains the rule modus ponens (MP).

Similarly, as in I, we can use a theorem prover to show that $\varphi \rightarrow \varphi$ is provable in $\mathcal{HI}$ for every φ (in our propositional language containing only $\rightarrow$ and $\neg$). On the other hand, $\varphi \rightarrow \psi$ is not provable in $\mathcal{HI}$, for every φ and ψ, as you can show by producing a counter-example.

Task

Let φ, ψ, and χ be arbitrary propositional formulae in our language containing only $\to$ and $\neg$. Decide for each of the following schemata whether it is provable in $\mathcal{HI}$ or not

(IIa) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$,

(IIb) $\neg\neg\varphi \to \varphi$,

(IIc) $\varphi \to \neg\neg\varphi$,

(IId) $((\varphi \to \psi) \to \varphi) \to \varphi$,

(IIe) $(\neg\varphi \to \psi) \to ((\neg\varphi \to \neg\psi) \to \varphi)$,

(IIf) $(\neg\varphi \to \psi) \to (\neg\psi \to \varphi)$,

(IIg) $(\neg\varphi \to \psi) \to (\psi \to \varphi)$.

Output

For each schema, upload an input file in the TPTP language that encodes the problem. Upload also the output of either a theorem prover or a model finder used on the input file that shows that the given schema is either provable or not in $\mathcal{HI}$.

III The relation between $\mathcal{HC}$ and $\mathcal{HI}$

Problem

There are two natural questions about the relation between $\mathcal{HC}$ and $\mathcal{HI}$, namely whether

(IIIa) all the formulae provable in $\mathcal{HC}$ are also provable in $\mathcal{HI}$, and

(IIIb) all the formulae provable in $\mathcal{HI}$ are also provable in $\mathcal{HC}$.

You should already know a partial answer from I and II. Here we should disclose that $\mathcal{HC}$ is a system for (classical) propositional logic and $\mathcal{HI}$ is a system for propositional intuitionistic logic.

Note that $\mathcal{HC}$ and $\mathcal{HI}$ share the rule modus ponens (MP) and some axiom schemata. Hence, for example, if you prove (I3) and (I4) in $\mathcal{HC}$, then it shows that everything provable in $\mathcal{HI}$ is also provable in $\mathcal{HC}$. On the other hand, if (I3) or (I4) is not provable in $\mathcal{HC}$, then this shows that not everything provable in $\mathcal{HI}$ is provable in $\mathcal{HC}$. Analogously you can decide whether everything provable in $\mathcal{HC}$ is also provable in $\mathcal{HI}$.

Output

For each problem (IIIa) and (IIIb), upload an input file (or files) in the TPTP language that encodes the problem. Upload also the output of either a theorem prover or a model finder used on the input file(s) that show(s) that the given problem is either provable or not.

Note

Maybe, you are wondering here why we prove it in such a complicated way. It should be easy to prove (IIIa) and (IIIb); we can introduce two separate provability predicates for $\mathcal{HC}$ and $\mathcal{HI}$, for example, *pr_hc* and *pr_hi*. And then, for example, proving/refuting $\forall X(pr\_hc(X) \to pr\_hi(X))$ answers (IIIa). However, this simple approach fails badly; the formula has a trivial counter-example, because there is a model where *pr_hc* is always true.