15-815 Automated Theorem Proving
Frank Pfenning, Spring 2004
Out Jan 29, due Feb 5

Assignment 2: A Certifying Decision Procedure

In this assignment you may work by yourself or with a partner. Library code and signatures can be found at
`http://www.cs.cmu.edu/~fp/courses/atp/assignments/asst2.sml`.

We explore an implementation of Dyckhoff's contraction-free sequent calculus **G4ip** [Dyc92, DN00] as a certifying decision procedure for intuitionistic propositional logic with the usual connectives. P stands for atomic propositions; Negation $\neg A$ is defined as $A \supset \bot$.

As in [DN00], we limit initial sequents to being atomic. This yields the following inference system, which has only three trivial differences to the original **G4ip** by including $\top$R, $\top$L, and $\bot\supset$L. To be precise one should label hypotheses with distinct variables, but we elide this here as usual in sequent calculi.

$$\frac{}{\Gamma, P \Longrightarrow P}\,\text{init}$$

$$\frac{\Gamma, A, B \Longrightarrow E}{\Gamma, A \wedge B \Longrightarrow E}\,\wedge\text{L} \qquad \frac{\Gamma \Longrightarrow A \qquad \Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \wedge B}\,\wedge\text{R}$$

$$\frac{\Gamma, A \Longrightarrow E \qquad \Gamma, B \Longrightarrow E}{\Gamma, A \vee B \Longrightarrow E}\,\vee\text{L} \qquad \frac{\Gamma \Longrightarrow A}{\Gamma \Longrightarrow A \vee B}\,\vee\text{R}_1 \qquad \frac{\Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \vee B}\,\vee\text{R}_2$$

$$\frac{}{\Gamma, \bot \Longrightarrow A}\,\bot\text{L} \qquad \text{no } \bot\text{R rule}$$

$$\frac{\Gamma \Longrightarrow E}{\Gamma, \top \Longrightarrow E}\,\top\text{L} \qquad \frac{}{\Gamma \Longrightarrow \top}\,\top\text{R}$$

$$\frac{\Gamma, P, B \Longrightarrow E}{\Gamma, P, P \supset B \Longrightarrow E}\,P\supset\text{L} \qquad \frac{\Gamma, A \Longrightarrow B}{\Gamma \Longrightarrow A \supset B}\,\supset\text{R}$$

$$\frac{\Gamma, C \supset (D \supset B) \Longrightarrow E}{\Gamma, (C \wedge D) \supset B \Longrightarrow E}\,\wedge\supset\text{L} \qquad \frac{\Gamma, B \Longrightarrow E}{\Gamma, \top \supset B \Longrightarrow E}\,\top\supset\text{L}$$

$$\frac{\Gamma, C \supset B, D \supset B \Longrightarrow E}{\Gamma, (C \vee D) \supset B \Longrightarrow E}\,\vee\supset\text{L} \qquad \frac{\Gamma \Longrightarrow E}{\Gamma, \bot \supset B \Longrightarrow E}\,\bot\supset\text{L}$$

$$\frac{\Gamma, C, D \supset B \Longrightarrow D \qquad \Gamma, B \Longrightarrow E}{\Gamma, (C \supset D) \supset B \Longrightarrow E}\,\supset\supset\text{L}$$

Question 1: Proof Term Assignment

Give a proof term assignment for **G4ip**, using the notation in lecture (as in Section 2.4 of the notes, without type labels). As in the proof of soundness for the sequent calculus, your proof term assignment should witness the soundness of the rules of **G4ip**. Are the proof terms you generate normal, that is, do they ever contain an introduction of a connective followed by its elimination? Explain why or why not.

Question 2: Proof Search

Provide an implementation of a decision procedure following the rules of **G4ip**. You should exploit the property that all rules except $\vee R_1$, $\vee R_2$ and $\supset\supset L$ are invertible. This means you can freely choose among multiple applicable invertible rules without the necessity to backtrack over these choices. When all applicable rules are non-invertible you may have to backtrack over the choices in order to make sure your procedure is complete.

Your implementation should be submitted as a single file with a `structure G_`*`yourid`*` :> G4IP` and a `functor T_`*`yourid`*` (D : G4IP) :> TEST`, but not include the provided library code. When the functor is applied to the structure it should generate an exception if one of the test cases produces an incorrect answer.

Efficiency is not a concern in this implementation; you should strive for correctness and elegance, in that order.

Question 3: Certification

Extend your implementation from Question 2 to generate proof terms for theorems. It should be submitted as a single file with a `structure GC_`*`yourid`*` :> G4IP_CERT` and `functor TC_`*`yourid`*` (D : G4IP_CERT) :> TEST` but not include the provided library code. When the functor is applied to the structure it should generate an exception if one of the test cases produces an incorrect answer.

The signature `ND` for proof terms is explained in Section 3.2 of the notes, except that we provide an additional construct $\mathbf{let}\, u = E\, \mathbf{in}\, I$ which represents a (postponed) substitution $[E/u]I$. The **let** form is easy to check by synthesizing a type A for E, and then checking I under the additional assumption that u has type A. The **let** form has the great advantage of allowing explicit sharing without introducing terms that are not normal (in the sense of introducing connectives that are then eliminated).

[DN00] Roy Dyckhoff and Sara Negri. Admissibility of structural rules for contraction-free systems of intuitionistic logic. *Journal of Symoblic Logic*, 65:1499–1518, 2000.

[Dyc92] Roy Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *Journal of Symbolic Logic*, 57:795–807, 1992.