General Tactics and Matching

Review

- The state of NO prover can be characterized

$$F_1 \wedge \ldots \wedge F_n \stackrel{?}{\Rightarrow} \bot$$

(trying to derive a contradiction from a set of literals)

- A satisfiability procedure can also declare subgoals

G is a subgoal if

$$F_1 \wedge \ldots \wedge F_n \wedge G \Rightarrow \bot$$ (G helps prove a contradiction)

This makes the prover try to prove

$$F_1 \wedge \ldots \wedge F_n \stackrel{?}{\Rightarrow} G$$

- This mechanism generalizes the equality sharing

$$F_1 \wedge \ldots \wedge F_n \Rightarrow E \quad \text{or.} \quad F_1 \wedge \ldots \wedge F_n \wedge \neg E \Rightarrow \bot$$

$\neg E$ is a subgoal

Next state is $F_1 \wedge \ldots \wedge F_n \wedge E \stackrel{?}{\Rightarrow} \bot$

- It also generalizes disjunction of equality sharing

$$F_1 \wedge \ldots \wedge F_n \Rightarrow E_1 \vee E_2$$

→ two subgoal states

$$F_1 \wedge \ldots \wedge F_n \wedge E_1 \stackrel{?}{\Rightarrow} \bot$$

$$F_1 \wedge \ldots \wedge F_n \wedge E_2 \stackrel{?}{\Rightarrow} \bot$$

Question: Is it possible to generate a tactic automatically from the axiomatization of a theory?

Answer: Yes, in some cases

Consider the axiomatization in the form of clauses $\dfrac{H_1 \dots H_m}{C}$

If this rule is appropriate for backwards chaining ($Var(H_i) \subseteq Var(C)$) then after instantiating the conclusion we have instances of the hypotheses

Thus, if the state is $F_1 \wedge \dots \wedge F_n \stackrel{?}{=} \bot$ and there is an instantiation θ such that

$$\theta(C) = F_i$$

then we can announce the subgoal $\theta(H_1) \wedge \dots \wedge \theta(H_m)$

This leads to m - subgoals

$$F_1 \wedge \dots \wedge F_n \stackrel{?}{\Longrightarrow} \theta(H_i) \quad i = 1 \dots m$$

This is in fact how Prolog interprets a set of clauses

- it is somewhat simpler here because of the condition on variables

But this form of matching is not enough.
Consider:

$$\frac{q(y)}{l(s(x,y),x)}$$

and trying to prove $p(a) = s(a,b) \wedge q(b) \Rightarrow l(p(a),a)$

There is no θ such that $\theta(\neg l(s(x,y),x)) = F_i$

- if we do regular matching.

But if we do E-DAG matching then

$$[a/x, b/y](\neg l(s(x,y),x)) = \neg l(p(a),a)$$

(because $p(a) = s(a,b)$)

Matching in the E-DAG is a slight modification on the regular matching

- instead of checking that $\theta(e) \equiv e'$
  check that $\theta(e)$ is congruent to e'

- take into consideration all members in an equivalence class
- the problem of determining whether there is a match is NP-complete
  - But experience suggest it is worth the cost

Still, this form of matching is too weak
- expects to find an exact instance of the negation of the conclusion among the literals
- still has the restriction that the conclusion has all the variables

Extension 1

- $\theta(H_i) = F_j$ then $\theta(H_1) \wedge \ldots \wedge \theta(H_{i-1}) \wedge \theta(H_i) \ldots \wedge \theta(\neg C)$ is a subgoal

- this might be a good idea when H_i contains all the variables
- The point here is to view the rule in a symmetric way as $\neg H_1 \vee \ldots \vee \neg H_n \vee C$

and to separate among H_i, C those that contain all variables

But what about a rule of the form

$$\frac{l(x,y) \quad l(y,z)}{l(x,z)}$$

- no hypothesis or conclusion contains all variables

Extension 2

- match a few H_i and/or $\neg C$ among the $\overline{F}$'s such that all variables are instantiated.

E.g. $$\frac{H_1 \ldots H_k \; H_{k+1} \ldots H_n}{C}$$

and $Var(C) \cup \bigcup_{i=1..k} Var(H_i) \supseteq \bigcup_{j=k+1..n} Var(H_j)$

If there is θ such that

$$\{\theta(\neg C), \theta(H_1), \ldots, \theta(H_k)\} \subseteq \overline{F}$$

then add the subgoal

$$\theta(H_{k+1}) \wedge \ldots \wedge \theta(H_n)$$

C and $H_1 \ldots H_k$ are trigger terms.

E.g. $$\frac{\ell(x,y) \quad \ell(y,z)}{\ell(x,z)}$$

triggers if $\ell(x,y) \wedge \ell(y,z) \dashrightarrow \neg \ell(x,z)$

or if $\ell(x,y) \wedge \neg \ell(x,z) \dashrightarrow \ell(y,z)$

...

which is how it should be

E.g.

$$\frac{g(x)}{l(p(x),x)}\ ① \qquad \frac{g(y)}{l(x,s(x,y))}\ ② \qquad \frac{l(x,y)\quad l(y,z)}{l(x,z)}\ ③$$

try to prove

$$g(a) \wedge g(b) \wedge \neg l(p(a), s(a,b))$$

② and ③ cannot fire

① can fire and produces $\boxed{l(p(a),a)}$ ← does not enable anything $l(p(b),b)$

Now ② still cannot fire

but ③ can fire $l(p(a),a) \wedge \neg l(p(a), s(a,b)) \Rightarrow \neg l(a, s(a,b))$

so we add literal $\boxed{\neg l(a, s(a,b))}$

(Note that 3 will never fire again with instantiation $x \to p(a)$, $y \to a$, $z \to s(a,b)$)

Now 2 can fire to add literal $\neg g(b)$

Which leads to a contradiction.

- But this is still not enough because we require a hypothesis, or conclusion to match completely an existing literal.
- We need instead a notion of "close" match.
- Several theorem provers say that the instantiation θ is a close match for clause $C_1 \vee \dots \vee C_n$ if for all variables x there is a subterm t of C_i that properly contains x such that $\theta(t)$ is <u>represented</u> in the E-DAG
  - we don't want to match the entire C_i but just a subterm
  - we take into consideration the equalities in the E-DAG

<u>Example</u>

$$\frac{g(x)}{l(p(x),x)}\ (1) \qquad \frac{g(y)}{l(x, s(f(x),y))}\ (2) \qquad \frac{l(x,y) \quad l(y,z)}{l(x,z)}\ (3)$$

- Two close matches for Ax 1 $x \to a$ and $x \to b$

~~No close match for Ax 2~~

Trying to falsify

$$c = f(a) \wedge g(a) \wedge g(b) \wedge \neg l(p(a), s(c,b))$$

- No close match for Ax. 3

One close match for Ax. 2

$x \to a$ ($f(x)$ is represented

$y \to b$ $g(x)$ is represented.

also $s(f(x), y)$ is represented

Say we do Ax 1 first

- we add literal $l(p(a), a)$

Now Ax. 3 has a close instance

$x \to p(a)$

$y \to a$

$z \to s(c, b)$

- add literal $\neg l(a, s(c, b))$

Now Ax. 2 has a close match. (same as before)

- add literal $\neg g(b)$

$\Rightarrow$ contradiction

But

- easy to go in a infinite loop
  $\Rightarrow$ do breadth first
- a multitude of other heuristics can be used

Once we have a good solution to matching we can think of extending the language of predicates that we handle.

- we can add
  - implication
  - universal quantification to the ~~right~~ left-hand side of $\Rightarrow$

  and - existential quantification to the right-hand side.