Discriminating Discriminator from iProver-Eq

Chad E. Brown

April 26, 2020

1 Introduction

We consider Example 1 from a paper about iProver-Eq [4], showing five Discriminator proof searches and arguing that none of these proof searches are similar to that performed by iProver-Eq.

2 The Example

We begin by discussing Example 1 from [4]. The example consists of four first-order clauses that form an unsatisfiable set:

$$f(f(u)) = f(u) \tag{1}$$

$$g(f(fx))(fy) = hz \vee g(fx)y \neq hc \tag{2}$$

$$g(fa)(fb) \neq hw \tag{3}$$

$$g(fa)b = hc \tag{4}$$

Note that u, w, x, y and z are variables implicitly universally quantified over the clauses in which they appear.

Let us quickly consider why this set is unsatisfiable. Clauses (2) and (4) together easily yield $g(f(fa))(fb) = hz$ (for arbitrary z). Simplifying with Clause (1) we have $g(fa)(fb) = hz$. If z and w are chosen to be the same term, then we have a conflict with Clause (3).

Neither iProver-Eq nor Discriminator follow this proof precisely, though the instantiations a and b are always present. The way iProver-Eq and Discriminator arrive at these instantiations are very different, as we we will see. In addition, Discriminator has no way of "simplifying" (or rewriting) $f(fa)$ to fa using Clause (1) in the equation $g(f(fa))(fb) = hz$. The reason is that the occurrence of $f(fa)$ in $g(f(fa))(fb) = hz$ is considered "too deep" for the rules used by Discriminator.

The iProver-Eq proof described in [4] proceeds as follows. First iProver-Eq instantiates all variables to a default element $\perp$. The resulting ground clauses are clearly satisfiable and the following satisfiable set of ground literals is selected:

- $f(f\bot) = f\bot$
- $g(f(f\bot))(f\bot) = h\bot$
- $g(fa)(fb) \neq h\bot$
- $g(fa)b = hc$

iProver-Eq then considers the first-order unit clauses (consisting only of the selected literals). Effectively Clause (2) is temporarily replaced with the following unit clause:

$$g(f(fx))(fy) = hz \tag{5}$$

The four unit clauses are shown to be unsatisfiable via a US (unit-superposition) calculus. The unit equation given by Clause (1) is used to rewrite $f(fx)$ to fx in Clause (5) yielding $g(fx)(fy) = hz$. Performing this inference required unification of $f(fx)$ in Clause (5) and $f(fu)$ in Clause (1). Unifying gives the substituion $u \mapsto x$. This inferred unit equation is used to rewrite $g(fa)(fb)$ to hz in Clause (3) yielding $hz \neq hw$. Again, the inference used unification on terms $g(fa)(fb)$ and $g(fx)(fy)$ yielding the substitution $x \mapsto a, y \mapsto b$. (This is how iProver-Eq first considers the instantiations a and b.) Finally a contradiction can be inferred from $hz \neq hw$ by unifying hz and hw to obtain $w \mapsto z$.

After proving this special case is unsatisfiable, iProver-Eq examines the instantiations used in each clause and adds new ground clauses based on these instantiations. In particular the following four ground clauses are now part of the set of ground clauses:

- $f(fa) = fa$
- $g(f(fa))(fb) = h\bot \vee g(fa)b \neq hc$
- $g(fa)(fb) \neq h\bot$
- $g(fa)b = hc$

These are unsatisfiable, as pointed out in [4]. However, it is worth noting that they are propositionally satisfiable. The unsatisfiability relies on the semantics of equality to determine that $f(fa) = fa$, $g(f(fa))(fb) = h\bot$ and $g(fa)(fb) \neq h\bot$ cannot all be true. This is decidable, of course, and can be determined using congruence closure.

The iProver-Eq proof above makes use of unification to instantiate free variables in clauses. Discriminator does not use unification and only works on closed formulas, so it must proceed differently. In order to compare different Discriminator proofs to the iProver-Eq proof above, let us list all ground instantiations and closed formulas considered during the iProver-Eq proof. We make implicit quantifications explicit so that all formulas are closed. The only ground instantiations appearing in the proof are $\bot$, a and b. The closed formulas are listed in Table 1. We omit the closed formulas corresponding to the original clauses, since these will be shared by every proof. This leaves us with 10 propositions.

$$\begin{gathered}
f(f\bot) = f\bot \\
g(f(f\bot))(f\bot) = h\bot \lor g(f\bot)\bot \neq hc \\
g(f(f\bot))(f\bot) = h\bot \\
g(fa)(fb) \neq h\bot \\
\forall xyz.g(f(fx))(fy) = hz \\
\forall xyz.g(fx)(fy) = hz \\
\forall zw.hz \neq hw \\
f(fa) = fa \\
g(f(fa))(fb) = h\bot \lor g(fa)b \neq hc \\
g(fa)(fb) \neq h\bot
\end{gathered}$$

Table 1: Propositions considered in iProver-Eq Proof

3 Proof Search 1

The first DISCRIMINATOR proof we consider was found using the default settings within 20s. Since DISCRIMINATOR insists on closed formulas, the clauses are given with explicit quantifiers:

$$\forall u.f(f(u)) = f(u) \tag{6}$$

$$\forall xyz.g(f(fx))(fy) = hz \lor g(fx)y \neq hc \tag{7}$$

$$\forall w.g(fa)(fb) \neq hw \tag{8}$$

$$g(fa)b = hc \tag{9}$$

The proof makes use of two shallow rules (described below) and, interestingly, makes use of not just instantiations a and b, but also c, hc, fb and $h(fb)$. All but the last of these instantiations are subterms of the original formulas, and are indeed also subterms of Proposition (9). The last instantiation $h(fb)$ is produced as a *discriminating term* [2] during the proof search.

When DISCRIMINATOR initially asserts the four propositions above, it also analyzes what shallow rules can be produced. Three shallow rules are produced from Proposition (6) and one shallow rule is produced from Proposition (8).

Only one of the shallow rules from Proposition (6) is used in the proof. It has the following form:

$$x, y | fx = fy \Leftarrow x = fy \tag{10}$$

The reader should verify that Proposition (6) implies $\forall xy.x = fy \rightarrow fx = fy$. The way the rule is used in practice is if a proposition of the form $fs \neq ft$ is processed, then the proposition $s \neq ft$ will be produced along with a propositional clause relating these propositions in an obvious way. Note that determining the substitution $x \mapsto s, y \mapsto t$ to apply the rule does not require unification or even general matching since the variables x and y both have shallow occurrences in $fx = fy$. (By contrast, x has no shallow occurence in $g(f(fx))(fy) = hz \lor g(fx)y \neq hc$ which is why Proposition (7) did not produce a shallow rule.)

The shallow rule produced by Proposition (8) has the following form:

$$x, y, z | gxy \neq hz \Leftarrow x = fa, y = fb \tag{11}$$

$\boxed{11}$	$\forall yz.(((g(f(fa))(fy)) = (hz)) \vee (g(fa)y) \neq (hc))$
$\boxed{12}$	$\forall yz.(((g(f(f(hc)))(fy)) = (hz)) \vee (g(f(hc))y) \neq (hc))$
$\boxed{14}$	$\forall z.(((g(f(f(hc)))(fb)) = (hz)) \vee (g(f(hc))b) \neq (hc))$
$\boxed{124}$	$((fa) = (fa))$
$\boxed{125}$	$(b = (fb))$
$\boxed{139}$	$(((g(f(f(hc)))(fb)) = (h(hc))) \vee (g(f(hc))b) \neq (hc))$
$\boxed{141}$	$((g(f(f(hc)))(fb)) = (h(hc)))$
$\boxed{142}$	$((fb) = (fb))$
$\boxed{169}$	$\forall z.(((g(f(fa))(fb)) = (hz)) \vee (g(fa)b) \neq (hc))$
$\boxed{173}$	$\forall z.(((g(f(fa))(fc)) = (hz)) \vee (g(fa)c) \neq (hc))$
$\boxed{175}$	$\forall z.(((g(f(fa))(f(hc))) = (hz)) \vee (g(fa)(hc)) \neq (hc))$
$\boxed{183}$	$(((g(f(fa))(f(hc))) = (h(hc))) \vee (g(fa)(hc)) \neq (hc))$
$\boxed{185}$	$((g(f(fa))(f(hc))) = (h(hc)))$
$\boxed{186}$	$((f(fa)) = (fa))$
$\boxed{215}$	$(((g(f(fa))(fc)) = (h(fb))) \vee (g(fa)c) \neq (hc))$
$\boxed{223}$	$((g(f(fa))(fc)) = (h(fb)))$
$\boxed{237}$	$(b = (h(fb)))$
$\boxed{1707}$	$(((g(f(fa))(fb)) = (h(h(fb)))) \vee (g(fa)b) \neq (hc))$
$\boxed{2298}$	$((g(f(fa))(fb)) = (h(h(fb))))$

Table 2: Propositions in Proof Search 1

The four propositions are assigned numbers and these numbers are sent to MiniSat as unit clauses. Proposition (6) is assigned $\boxed{4}$, Proposition (7) is assigned $\boxed{3}$, Proposition (8) is assigned $\boxed{2}$ and Proposition (9) is assigned $\boxed{1}$. Propositions (6) and (8) are not processed during the search and only appear in relation to their corresponding shallow rules.

The remaining propositions and clauses are generated during the search. Table 2 lists these propositions along with their assigned number used to communicate with MiniSat.

Let us now consider the steps DISCRIMINATOR took leading to a successful proof. These steps are shown in Figure 1. A total of 2401 steps were taken before the proof was found, but most of these did not contribute to the proof.

The first steps that contribute to the proof (Steps 0-3 and Step 6) process subterms of the initial asserted propositions as instantiations. Step 0 processes hc, Step 1 processes a, Step 2 processes c, Step 3 processes fb and Step 6 processes b. The result of processing these instantiations is to make them available to use as instantiations when processing a universal quantifier. Note that DISCRIMINATOR has already obtained the instantiations a and b simply by including all subterms of the problem, rather than as a result of any unification. (This will be true for all the DISCRIMINATOR proofs we consider until the last one.)

Step 8 processes Proposition (7) instantiating it with all instantiations processed so far. Two instances, corresponding to instantiating x with a and hc will play a role in the successful proof search. In the case of the instance with a the corresponding MiniSat clause will also play a role in the final propositional unsatisfiable. This clause records that if Proposition (7) is true, then so is the instance with a for x.

Step 9 processes the instance of Proposition (7) with hc, i.e.,

$$\forall yz.(((g(f(f(hc)))(fy)) = (hz)) \vee (g(f(hc))y) \neq (hc)).$$

Here y is instantiated with all the instantiations so far. The one that will play a role later is the instance with $y \mapsto b$. We return to process this instance in Step 114.

Step 100 processes Proposition (9) using Shallow Rule (11) producing propositions $fa \neq fa$ and $b \neq fb$. The proposition $fa \neq fa$ is processed in Step 101 yielding a unit clause for MiniSat indicating $fa = fa$ is true. Propositions of the form $s \neq s$ are typically eagerly processed to record that $s = s$ is true.

In Step 114 we return to process the proposition

$$\forall z.(((g(f(f(hc)))(fb)) = (hz)) \vee (g(f(hc))b) \neq (hc)).$$

As a universally quantified formula we instantiate with all the instantiations so far. The one that will be built upon is the instance with $z \mapsto hc$. This instance is processed in Step 115 producing two propositions (from the two disjuncts), only one of which will contribute to the successful part of the proof search, the equation $g(f(f(hc)))(fb) = h(hc)$. In Step 116 we process this equation applying Shallow Rule (11) to produce $fb \neq fb$. This is then eagerly processed in Step 117 so that we now know $fb = fb$.

The reader may already observe that some steps DISCRIMINATOR takes only "contribute to the proof" by giving roundabout ways to consider a proposition of the form $s \neq s$ for a particular s and then determine $s = s$ is true.

Step 149 returns to the instance of Proposition (7) using $x \mapsto a$ (the more reasonable instance). Three instances of this contribute to the proof, one directly (with $y \mapsto b$), and two indirectly (with $y \mapsto c$ and $y \mapsto hc$). A MiniSat clause recording the relationship between the proposition and its instance with $y \mapsto b$ will contribute to proposition unsatisfiability. Unfortunately DISCRIMINATOR does not process this instance until Step 253. It first processes the other two instances. Step 150 processes the instance with $y \mapsto hc$, instantiating it with $z \mapsto hc$. The resulting disjunction is processed in Step 151 producing the equation $g(f(fa))(f(hc)) = h(hc)$. Shallow Rule (11) applied to this equation in Step 152 produces the disequation $f(fa) \neq fa$. Note that the disquation $f(fa) \neq fa$ is relevant to the proof, even though it was arrived at in a roundabout way. Indeed DISCRIMINATOR ultimately processes $f(fa) \neq fa$ in the last step of the proof, Step 2400, yielding propositional unsatisfiability.

Step 184 processes the instance using $y \mapsto c$ from Step 149. Here the instance $z \mapsto fb$ is produced and then processed in Step 191. Processing this disjunction produces the equation $g(f(fa))(fc) = h(fb)$ which is processed in Step 192. Processing this equation makes it available to be used later. In particular it will be used to confront a disequation in the next step we describe, Step 200.

Step 200 processes the disequation $b \neq fb$ produced in Step 100. The equation $g(f(fa))(fc) = h(fb)$ confronts this disequation producing the disequation $b \neq h(fb)$. (Readers unfamiliar with the confrontation rule can consult [3, 2, 1].)

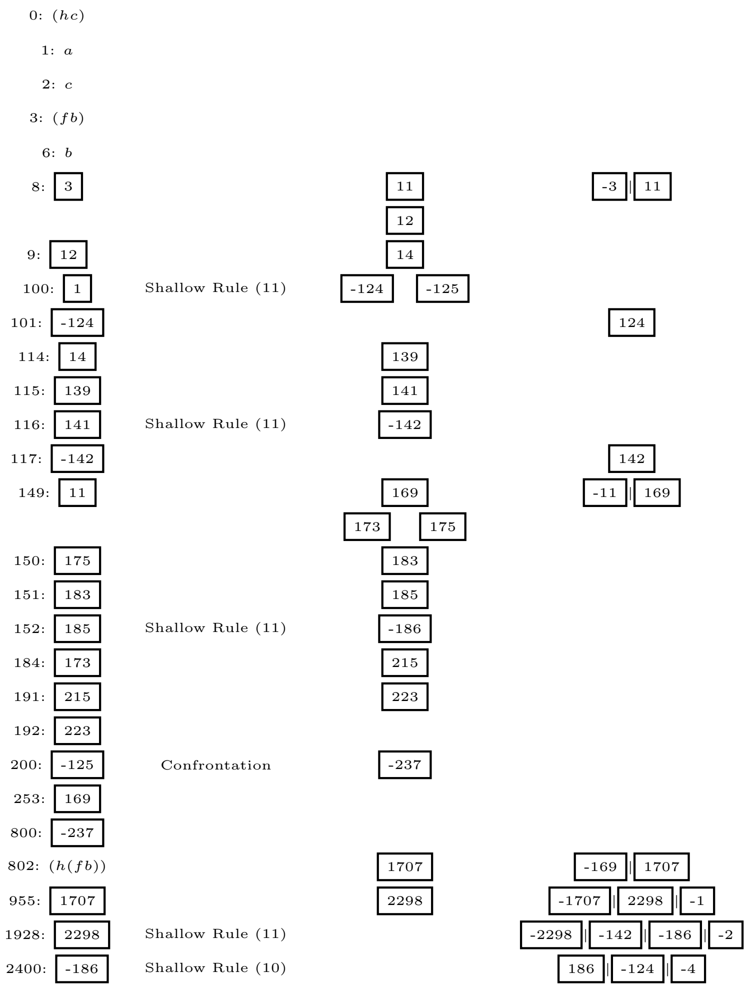

0: (hc)
1: a
2: c
3: (fb)
6: b
8: 3 11 -3 | 11
12
9: 12 14
100: 1 Shallow Rule (11) -124 -125
101: -124 124
114: 14 139
115: 139 141
116: 141 Shallow Rule (11) -142
117: -142 142
149: 11 169 -11 | 169
173 175
150: 175 183
151: 183 185
152: 185 Shallow Rule (11) -186
184: 173 215
191: 215 223
192: 223
200: -125 Confrontation -237
253: 169
800: -237
802: $(h(fb))$ 1707 -169 | 1707
955: 1707 2298 -1707 | 2298 | -1
1928: 2298 Shallow Rule (11) -2298 | -142 | -186 | -2
2400: -186 Shallow Rule (10) 186 | -124 | -4

Figure 1: Search Steps Leading to Proof 1

Step 253 finally processes the instance of Proposition (8) with $x \mapsto a$ and $y \mapsto b$ produced in Step 149. The proposition being processed is

$$\forall z.(((g(f(fa))(fb)) = (hz)) \vee (g(fa)b) \neq (hc)).$$

Discriminator instantiates this with all instantiations processed so far, but none of these end up being used as part of the ultimate proof (though any of them could have been, in principle). Instead the quantified formula is kept to be instantiated with later instantiations.

Step 800 processes the disequation $b \neq h(fb)$. A side effect of processing the disequation is to note $h(fb)$ is now a discriminating term (occurs on one side of a disequation) and can be potentially used for instantiations. Step 802 processes $h(fb)$ as an instantiation with the effect of instantiating every previously processed universally quantified formula with this new instantiation. In particular the proposition processed in Step 253 is instantiated with $h(fb)$ yielding the disjunction

$$g(f(fa))(fb) = h(h(fb)) \vee g(fa)b \neq hc$$

and a MiniSat clause relating the quantified proposition to the instance. Step 955 processes the disjunction. This could have produced two propositions $g(f(fa))(fb) = h(h(fb))$ and $g(fa)b \neq hc$, but $g(fa)b \neq hc$ is not new (see Proposition (9) and Step 100). The proposition $g(f(fa))(fb) = h(h(fb))$ is new and is produced along with a MiniSat clause relating the disjunction to its disjuncts. If we read the MiniSat clauses along with the initial four unit clauses, at this point we effectively know the equation $g(f(fa))(fb) = h(h(fb))$ must be true. Step 1928 processes $g(f(fa))(fb) = h(h(fb))$ applying Shallow Rule (11) to produce a clause that says $g(f(fa))(fb) = h(h(fb))$ is false if $f(fa) = fa$ and $fb = fb$. We already know $fb = fb$ from Step 117. Since we know $g(f(fa))(fb) = h(h(fb))$, we must have $f(fa) \neq fa$. This will conflict with the final step.

In the final step, Step 2400, the disequation $f(fa) \neq fa$ (produced in Step 152) is processed. Shallow Rule (10) is triggered yielding a clause that says $f(fa) = fa$ is true if $fa = fa$ is true. We know $fa = fa$ from Step 100 and so $f(fa) = fa$ must be true. This conflicts with Step 1928. Technically MiniSat has noted propositional unsatisfiability of the clauses produced by Discriminator.

Let us briefly consider how similar or different the proof search performed by Discriminator is from that of iProver-Eq in this instance. Discriminator made use of six instantiations a, b, c, hc, fb and $h(fb)$ and iProver-Eq made use of three instantiations $\perp$, a and b. These do have a and b in common, as one would expect, but are clearly different sets. As for the propositions used in the (successful part) of the proof searches, comparing the 10 propositions in Table 1 to the 19 propostions in Table 2 we no common propositions. The closest match is that iProver-Eq made use of the equation $f(fa) = fa$ and Discriminator made use of the disequation $f(fa) \neq fa$.

$\boxed{5}$	$((fa) = (fa))$
$\boxed{43}$	$\forall yz.(((g(f(fb))(fy)) = (hz)) \vee (g(fb)y) \neq (hc))$
$\boxed{44}$	$\forall z.(((g(f(fb))(fb)) = (hz)) \vee (g(fb)b) \neq (hc))$
$\boxed{66}$	$(((g(f(fb))(fb)) = (hb)) \vee (g(fb)b) \neq (hc))$
$\boxed{70}$	$((g(f(fb))(fb)) = (hb))$
$\boxed{72}$	$((fb) = (fb))$
$\boxed{264}$	$\forall yz.(((g(f(fa))(fy)) = (hz)) \vee (g(fa)y) \neq (hc))$
$\boxed{290}$	$\forall z.(((g(f(fa))(fb)) = (hz)) \vee (g(fa)b) \neq (hc))$
$\boxed{297}$	$(((g(f(fa))(fb)) = (hc)) \vee (g(fa)b) \neq (hc))$
$\boxed{334}$	$((g(f(fa))(fb)) = (hc))$
$\boxed{339}$	$((f(fa)) = (fa))$

Table 3: Propositions in Proof Search 2

4 Proof Search 2

The first proof was quite roundabout and took almost 20s to search for this proof. This was simply a consequence of the order in which propositions and instantiations were processed by default. A *priority salt* parameter can optionally be given to DISCRIMINATOR in order to reorder the search by pseudorandomly modifying the priority of options in the priority queue. By simply setting the priority salt to 2, DISCRIMINATOR can find a proof within 1s. We consider this second proof in this section. It produces and uses the same shallow rules as the previous proof. As in the previous proof, the initial four propositions are assigned numbers $\boxed{4}$, $\boxed{3}$, $\boxed{2}$ and $\boxed{1}$ and these are given as unit clauses to MiniSat. The remaining propositions generated during the search (and contributing to the ultimate success) are listed in Table 3. The main search only requires 123 steps (as opposed to 2401 steps for the first proof) before the clauses given to MiniSat become unsatisfiable. The relevant steps of the proof search are given in Figure 2.

Step 0 processes Proposition (9) triggering Shallow Rule (11) to produce $fa \neq fa$ which is processed in Step 1 to produce a clause recording that $fa = fa$ is true.

Step 2 processes the instantiation c (available as a subterm of one of the four propositions) making it available for later instantiations. Step 3 processes Proposition (7) making it available to be instantiated. Step 27 processes the instantiation b instantiating it into Proposition (7) to give the proposition

$$\forall yz.g(f(fb))(fy) = hz \vee g(fb)y \neq hc.$$

Step 28 processes this new proposition instantiating it with b as well giving

$$\forall z.g(f(fb))(fb) = hz \vee g(fb)b \neq hc.$$

Step 51 processes this new proposition instantiating it with b giving

$$g(f(fb))(fb) = hb \vee g(fb)b \neq hc.$$

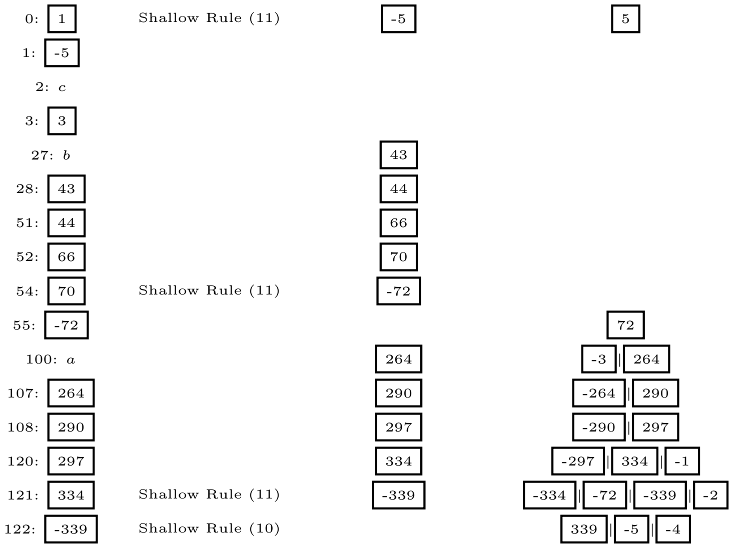


Figure 2: Search Steps Leading to Proof 2

Step 52 processes this disjunction producing the disjunct $g(f(fb))(fb) = hb$. Step 54 processes $g(f(fb))(fb) = hb$ triggering Shallow Rule (11) to produce the disequation $fb \neq fb$ which is processed in Step 55 yielding a clause recording the truth of $fb = fb$.

Step 100 processes the instantiation a instantiating into Proposition (7) giving

$$\forall yz.g(f(fa))(fy) = hz \vee g(fa)y \neq hc$$

along with a clause implying this proposition is true. Step 107 processes this proposition instantiating it with b giving

$$\forall z.g(f(fa))(fb) = hz \vee g(fa)b \neq hc$$

along with a clause implying this new proposition is true. The new proposition is processed in Step 108 and is instantiated with c giving

$$g(f(fa))(fb) = hc \vee g(fa)b \neq hc$$

along with a clause indicating this disjunction is true. The disjunction is processed in Step 120 giving the new proposition $g(f(fa))(fb) = hc$ and a clause relating the disjunction to its disjuncts. Since we know the disjunction as true and the right disjunct is the negation of Proposition (9) we know $g(f(fa))(fb) = hc$ must be true. Step 121 processes $g(f(fa))(fb) = hc$ triggering Shallow Rule (11) to produce the disequation $f(fa) \neq fa$ and a clause implying $f(fa) \neq fa$ must be true (since $g(f(fa))(fb) = hc$ is known from Step 120 and $fb = fb$ is known from Step 55). The final step, Step 122, processes $f(fa) \neq fa$ triggering Shallow Rule (10) to produce a clause implying $f(fa) = fa$ must be true (since $fa = fa$ is known from Step 1). This conflict completes the proof.

This second DISCRIMINATOR proof is more straightforward than the first. The only instantiations used are a, b and c, which is closer to the three ($\perp$, a and b) used in the iProver-Eq proof. Among the 13 propositions in Table 3 there are again none in common with Table 1, with the closest match being $f(fa) = fa$ vs. $f(fa) \neq fa$. It is worth noting that even this this formula, iProver-Eq obtained $f(fa) = fa$ as an instance of Clause (1) used to rewrite a subterm of (essentially) Clause (2). On the contrary, DISCRIMINATOR obtained $f(fa) \neq fa$ as a result of applying a shallow rule derived from Proposition (8) – which corresponds to Clause (3), not Clause (2). A shallow rule derived from Proposition (6) is used at the end of both of these first two DISCRIMINATOR proofs simply to obtain a conflict with $f(fa) \neq fa$ and not to positively derive or use $f(fa) = fa$.

5 Proof Search 3

The third proof was found by DISCRIMINATOR in less than a second with parameters set that limited the shallow rules produced to functional ones. Functional shallow rules are of the form

$$x_1, \ldots, x_n | s \rightsquigarrow t \Leftarrow \phi_1, \ldots, \phi_m$$

where each of the variables $x_1, \ldots, x_n$ has a unique shallow occurrence in s. The shallow rule is triggered when processing a disequation of the form $t_1 \neq t_2$ where either t_1 or t_2 has the form θs for some θ. As before, this θ can be easily computed without general matching due to the shallow occurrences of the variables in s. Suppose t_1 has the form θs. In this case the shallow rule produces the disequation $\theta t \neq t_2$, propositions $\theta\phi_1, \ldots, \theta\phi_m$ and a MiniSat clause indicating that if all of $\theta\phi_1, \ldots, \theta\phi_m$ are true and $t_1 \neq t_2$ is true, then $\theta t \neq t_2$ is true. If t_2 has the form θs, then the disequation $\theta t \neq t_1$ plays the role of $\theta t \neq t_2$ in the description of the previous case.

The priority salt for the search was set to 1.

Two functional shallow rules produced and used. The first is produced from Proposition (6) and has the form:

$$x | fx \rightsquigarrow f(fx) \tag{12}$$

This rule is triggered when processing a disequation $fs \neq t$ or $t \neq fs$. The rule produces $f(fs) \neq t$ and a MiniSat clause relating the new disequation to the one being processed.

The second shallow rule is not produced by one of the initial four propositions, but by a proposition processed during the search. In particular, after instantiating Proposition (7) with $x \mapsto a$ and $y \mapsto b$, we have the proposition

$$\forall z. g(f(fa))(fb) = hz \vee g(fa)b \neq hc.$$

$\boxed{7}$	$\forall yz.(((g(f(fa))(fy)) = (hz)) \vee (g(fa)y) \neq (hc))$
$\boxed{9}$	$\forall z.(((g(f(fa))(fb)) = (hz)) \vee (g(fa)b) \neq (hc))$
$\boxed{22}$	$((g(fa)(fb)) = (ha))$
$\boxed{25}$	$((g(f(fa))(fb)) = (g(fa)(fb)))$
$\boxed{26}$	$((f(fa)) = (fa))$
$\boxed{27}$	$((fb) = (fb))$
$\boxed{31}$	$((f(fa)) = (f(fa)))$

Table 4: Propositions in Proof Search 3

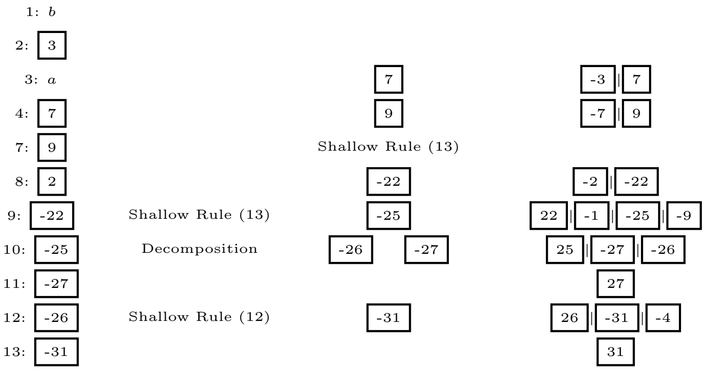


Figure 3: Search Steps Leading to Proof 3

This can be read as an implication that if $g(fa)b = hc$, then $hz = g(f(fa))$ for every z. This z has a shallow occurrence in hz, justifying the creation of the following shallow rule:

$$z|hz \rightsquigarrow g(f(fa))(fb) \Leftarrow g(fa)b = hc \tag{13}$$

The propositions, other than the initial four, produced during the search are given in Table 4. This search uses the least number of extra propositions, namely, 7. The only instantiations used are a and b. This is arguably the most straightforward of the five Discriminator proofs we consider. The steps are shown in Figure 3.

Step 1 processes b (again, available as a subterm of the original propositions). Step 2 processes Proposition (7). Step 3 processes a instantiating it into Proposition (7) giving the instance and a clause recording its truth. Step 4 processes this instance instantiating it with b giving the instance and a clause recording its truth. This instance is processed in Step 7 producing Shallow Rule (13) described above.

Step 8 processes Proposition (8) instantiating it with a to obtain the disequation $g(fa)(fb) \neq ha$ along with a clause indicating the disequation is true. Step

9 processes the disequation triggering Shallow Rule (13) to produce the disequation $g(f(fa))(fb) \neq g(fa)(fb)$ and a clause implying $g(f(fa))(fb) \neq g(fa)(fb)$ must be true (since the side condition of Shallow Rule (13) is Proposition (9)). The disequation $g(f(fa))(fb) \neq g(fa)(fb)$ is processed in Step 10 and decomposed into $f(fa) \neq fa$ and $fb \neq fb$ with a clause indicating that since $g(f(fa))(fb) \neq g(fa)(fb)$ is true one of $f(fa) \neq fa$ or $fb \neq fb$ must be true. Step 11 processes $fb \neq fb$ determining $fb = fb$ is true. Hence we must have $f(fa) \neq fa$. Step 12 processes $f(fa) \neq fa$ triggering Shallow Rule (12) to produce $f(fa) \neq f(fa)$ and a clause indicating $f(fa) \neq f(fa)$ must be true. Step 13 processes $f(fa) \neq f(fa)$ to determine $f(fa) = f(fa)$ is true, a conflict that completes the proof.

In this case no extra instantiation terms are used, and so the set $\{a, b\}$ if instantiations is very close to the set $\{\perp, a, b\}$ used by iProver-Eq. It is still, however, the case that these instantiations were arrived at by including all ground subterms as initial instantiations and not through a process of unification during the search. There is again no overlap with the formulas in Tables 1 and 4 with the closest matches being $f(fa) = fa$ vs. $f(fa) \neq fa$. Note that in this case the conflict with $f(fa) \neq fa$ was not obtained by either instantiating Proposition (6) or by using Proposition (6) to reduce $f(fa)$ to fa and using $fa = fa$. Instead a shallow rule produced from Proposition (6) is used to expand $f(fa) \neq fa$ to be $f(fa) \neq f(fa)$ which is in conflict with $f(fa) = f(fa)$. The propositions $fa = fa$ and $fa \neq fa$ are never considered in this proof.

6 Proof Search 4

The remaining two searches are longer as they do not make use of shallow rules. We do not describe them in the same level of detail as the previous three proofs, relying on the reader to fill in the details from the given tables and figures.

The fourth proof was found automatically by Discriminator within a second with parameters set so that no shallow rules would be produced or used. The priority salt was set to 44. The most notable difference between this proof and the others is the use of congruence closure in the final step.

The instantiations used in this proof will be a, b and $g(fa)a$. While a and b are subterms of the original problem, $g(fa)a$ is produced as a discriminating term during the search. The propositions produced during the search are shown in Table 5. The steps of the proof are shown in Figure 4.

Step 0 processes b as a potential instantiation. Steps 1 and 2 process Propositions (7) and (6). Step 3 processes a. In summary these steps produce $f(fb) = fb$, $f(fa) = fa$ and

$$\forall yz.g(f(fa))(fy) = hz \vee g(fa)y \neq hc$$

with corresponding information given to MiniSat. Steps 4, 6 and 11 instantiates

$$\forall yz.g(f(fa))(fy) = hz \vee g(fa)y \neq hc$$

6	$((f(fb)) = (fb))$
7	$((f(fa)) = (fa))$
8	$\forall yz.(((g(f(fa))(fy)) = (hz)) \vee (g(fa)y) \neq (hc))$
9	$\forall z.(((g(f(fa))(fa)) = (hz)) \vee (g(fa)a) \neq (hc))$
10	$\forall z.(((g(f(fa))(fb)) = (hz)) \vee (g(fa)b) \neq (hc))$
11	$(((g(f(fa))(fb)) = (ha)) \vee (g(fa)b) \neq (hc))$
12	$(((g(f(fa))(fb)) = (hb)) \vee (g(fa)b) \neq (hc))$
13	$((g(f(fa))(fb)) = (ha))$
14	$((g(f(fa))(fb)) = (hb))$
16	$(((g(f(fa))(fa)) = (hb)) \vee (g(fa)a) \neq (hc))$
17	$((g(fa)a) = (hc))$
32	$(((g(f(fa))(fb)) = (h(g(fa)a))) \vee (g(fa)b) \neq (hc))$
34	$((f(f(g(fa)a))) = (f(g(fa)a)))$
82	$((g(f(fa))(fb)) = (h(g(fa)a)))$
120	$((g(fa)(fb)) = (hb))$

Table 5: Propositions in Proof Search 4

further with $y \mapsto a, z \mapsto b$, $y \mapsto b, z \mapsto a$ and $y \mapsto b, z \mapsto b$. The result disjunctions are processed in Steps 7, 8 and 12. In particular, Step 12 processes $g(f(fa))(fa) = hb \vee g(fa)a \neq hc$ to produce the disequation $g(fa)a \neq hc$ which, after being processed in Step 14, makes $g(fa)a$ a discriminating term, processed as an instantiation in Step 15. Step 15 instantiates

$$\forall z.g(f(fa))(fb) = hz \vee g(fa)b \neq hc$$

and Proposition (6) with $g(fa)a$ giving

$$g(f(fa))(fb) = h(g(fa)a) \vee g(fa)b \neq hc$$

and

$$f(f(g(fa)a)) = f(g(fa)a).$$

Processing

$$g(f(fa))(fb) = h(g(fa)a) \vee g(fa)b \neq hc$$

produces the equation $g(f(fa))(fb) = h(g(fa)a)$.

Step 75 processes Proposition (8) instantiating it with b to give the disequation $g(fa)(fb) \neq hb$.

Unless instructed otherwise, when DISCRIMINATOR processes equations and disequations, it uses the MiniSat clauses so far to determine which equations must be true and adds these to a graph for testing congruence closure [5] and tests the sides of disequations processed for congruence. If the two sides must be equal, a clause is produced for MiniSat recording that the equations known so far imply the disequation is false. This is what happens in the final step, Step 78, in Figure 4. In particular, congruence closure determines that the disequation $g(fa)(fb) \neq hb$ is in conflict with the following (ground) equations:

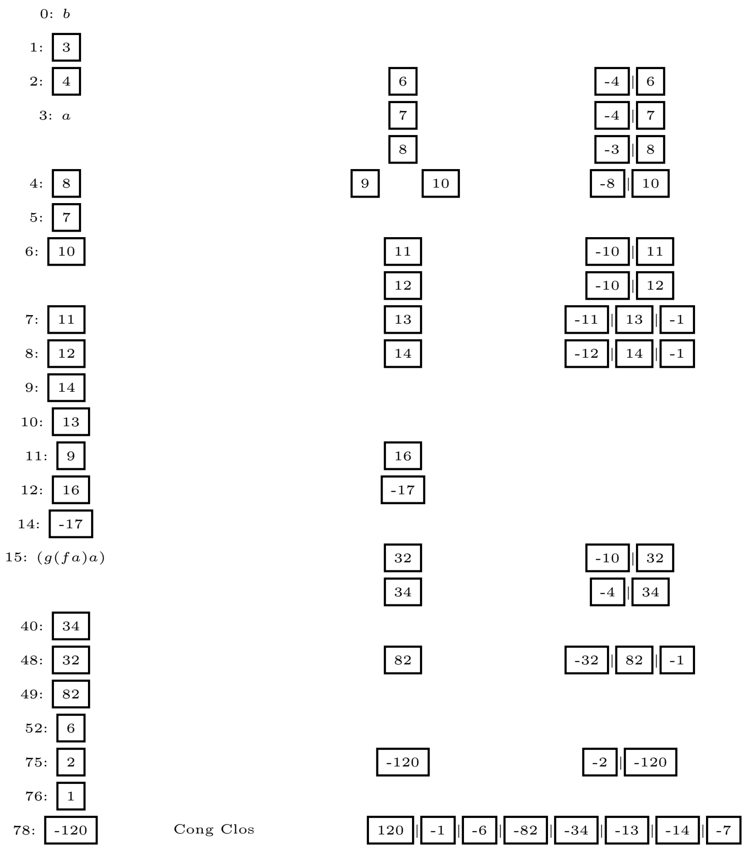


Figure 4: Search Steps Leading to Proof 4

- $f(fa) = fa$
- $g(f(fa))(fb) = hb$
- $g(f(fa))(fb) = ha$
- $f(f(g(fa)a)) = f(g(fa)a)$
- $g(f(fa))(fb) = hb$
- $g(f(fa))(fb) = h(g(fa)a)$
- $f(fb) = fb$
- $g(fa)b = hc$

Clearly only the first two of these equations are necessary to determine $g(fa)(fb)$ must equal hb, but this level of precision is not maintained. With more precision, most of the steps of this proof attempt would (likely) be considered irrelevant.

Unlike the previous three proofs the proposition $f(fa) = fa$ is derived positively from Proposition (6), providing one formula in common in Table 1 and 5. The three instantiations a, b and $g(fa)a$ is close to $\bot$, a and b used by iProver-Eq. While the structure of the proof search clearly differs from that of iProver-Eq, this case is arguable the closest of the five Discriminator searches to the iProver-Eq proof.

7 Proof Search 5

The final proof we consider was not found automatically by Discriminator. Instead "hints" were given in a file indicating some propositions and instantiations to prefer processing. The proof is longer than the others, but only uses basic rules of the calculus and no extra heuristics. In particular, it uses no shallow rules, no congruence closure, and does not seed the instantiations with subterms of the original problem. Instead a default instantiation $\bot$ (analogous to that used by iProver-Eq) is used in the beginning and instantiations like a and b are only allowed after they have occurred as discriminating terms. The relevant propositions produced during the proof search are given in Table 6. The relevant steps are shown in Figure 5.

The proof search begins by creating and processing a default instantiation $\bot$, since there are no disequations and hence no discriminating terms in the problem and we have instructed Discriminator not to use subterms as initial instantiations. As a consequence the disequation $g(fa)(fb) \neq h\bot$ is derived (so that there are now discriminating terms, though neither will prove useful) and confronted by $g(fa)b = hc$ to produce $g(fa)(fb) \neq g(fa)b$. After decomposition we have disequations $fa \neq fa$ and $fb \neq b$. After processing $fb \neq b$, fb and b are discriminating and can be processed as instantiations. First b is processed in Step 7 leading to $g(fa)(fb) \neq hb$, processed in Step 8.

5	$((g(fa)(fb)) = (h\perp))$
7	$((g(fa)(fb)) = (g(fa)b))$
10	$((fa) = (fa))$
11	$((fb) = b)$
17	$((g(fa)(fb)) = (hb))$
23	$\forall yz.(((g(f(f\perp))(fy)) = (hz)) \vee (g(f\perp)y) \neq (hc))$
25	$((f(fb)) = (fb))$
46	$((fb) = (fb))$
65	$\forall z.(((g(f(f\perp))(f\perp)) = (hz)) \vee (g(f\perp)\perp) \neq (hc))$
67	$(((g(f(f\perp))(f\perp)) = (hb)) \vee (g(f\perp)\perp) \neq (hc))$
82	$((g(f(f\perp))(f\perp)) = (hb))$
83	$((hb) = (hb))$
217	$\forall yz.(((g(f(f(fb)))(fy)) = (hz)) \vee (g(f(fb))y) \neq (hc))$
221	$\forall z.(((g(f(f(fb)))(fb)) = (hz)) \vee (g(f(fb))b) \neq (hc))$
225	$(((g(f(f(fb)))(fb)) = (hb)) \vee (g(f(fb))b) \neq (hc))$
228	$((g(f(f(fb)))(fb)) = (hb))$
230	$((g(fa)(fb)) = (g(f(f(fb)))(fb)))$
236	$((fa) = (f(f(fb))))$
256	$(a = (f(fb)))$
352	$((f(fa)) = (fa))$
353	$\forall yz.(((g(f(fa))(fy)) = (hz)) \vee (g(fa)y) \neq (hc))$
358	$\forall z.(((g(f(fa))(fb)) = (hz)) \vee (g(fa)b) \neq (hc))$
363	$(((g(f(fa))(fb)) = (hb)) \vee (g(fa)b) \neq (hc))$
365	$((g(f(fa))(fb)) = (hb))$
390	$((g(fa)(fb)) = (g(f(fa))(fb)))$
396	$((fa) = (f(fa)))$

Table 6: Propositions in Proof Search 5

Many of the steps from Step 9 to Step 85 are roundabout ways of determining $fa = fa$, $fb = fb$ and $hb = hb$. We leave the reader to go through the details.

Step 85 processes fb as an instantiation leading ultimately to a being a discriminating term so that it is available as an instantiation. We sketch how this occurs. Using fb and b as instantiations in Proposition 7 leads to considering the equation $g(f(f(fb)))(fb) = hb$. Confronting $g(fa)(fb) \neq hb$ with $g(f(f(fb)))(fb) = hb$ gives the disequation $g(fa)(fb) \neq g(f(f(fb)))(fb)$. Decomposition leads to $fa \neq f(f(fb))$ and then $a \neq f(fb)$, making a a discriminating term.

Step 96 processes a as an instantiation giving $f(fa) = fa$ and

$$\forall yz.g(f(fa))(fy) = hz \vee g(fa)y \neq hc.$$

Instiating this with $y \mapsto b, z \mapsto b$ (in two steps) gives

$$g(f(fa))(fb) = hb \vee g(fa)b \neq hc$$

and then $g(f(fa))(fb) = hb$. Step 101 processes $g(f(fa))(fb) = hb$ confronting $g(fa)(fb) \neq hb$ (from Steps 7 and 8) to produce the disequation $g(fa)(fb) \neq g(f(fa))(fb)$. Decomposition leads to $fa \neq f(fa)$. The final step, Step 105, processes the equation $f(fa) = fa$ and confronts the disequation $fa \neq f(fa)$ leading to propositional unsatisfiability.

This final proof has the most potential to be close to the iProver-Eq proof since both begin with a default instantiation $\perp$. The instantiations used by Discriminator are $\perp$, a, b and fb, which is the same as the iProver-Eq proof except for fb. Nevertheless, the varied propositions produced by Discriminator give clear evidence, that the proof search is quite different. A close examination of Tables 1 and 6 reveals that still the only proposition in common is $f(fa) = fa$.

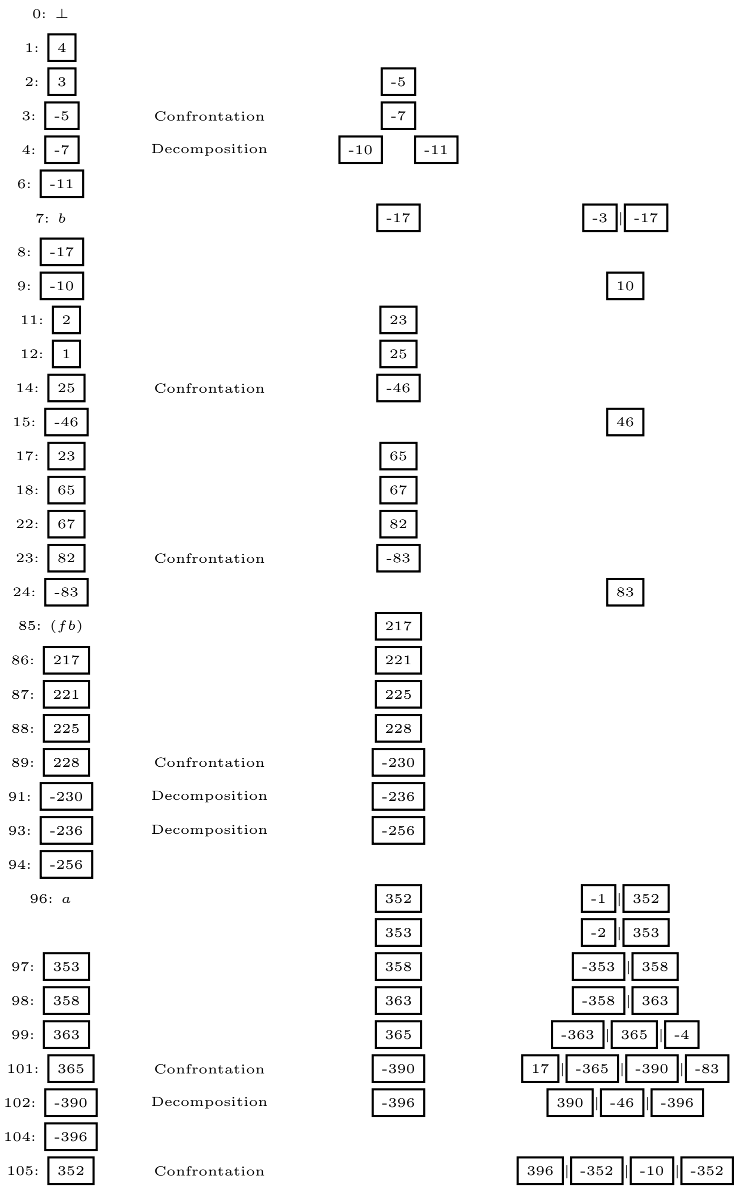
8 Conclusion

We have considered an example proof for iProver-Eq given in [4] and five proofs given by Discriminator. Although both provers can be said to be "instantiation-based," it is clear from the example proof searches that the techniques used by Discriminator are fundamentally different from those used by iProver-Eq.

References

[1] Backes, J., Brown, C.E.: Analytic tableaux for higher-order logic with choice. Journal of Automated Reasoning 47(4), 451–479 (2011), dOI 10.1007/s10817-011-9233-2

[2] Brown, C.E., Smolka, G.: Extended first-order logic. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Proceedings. LNCS, vol. 5674, pp. 164–179. Springer (Aug 2009)



0: ⊥			
1: 4			
2: 3		-5	
3: -5	Confrontation	-7	
4: -7	Decomposition	-10 -11	
6: -11			
7: *b*		-17	-3 | -17
8: -17			
9: -10			10
11: 2		23	
12: 1		25	
14: 25	Confrontation	-46	
15: -46			46
17: 23		65	
18: 65		67	
22: 67		82	
23: 82	Confrontation	-83	
24: -83			83
85: (*f b*)		217	
86: 217		221	
87: 221		225	
88: 225		228	
89: 228	Confrontation	-230	
91: -230	Decomposition	-236	
93: -236	Decomposition	-256	
94: -256			
96: *a*		352	-1 | 352
		353	-2 | 353
97: 353		358	-353 | 358
98: 358		363	-358 | 363
99: 363		365	-363 | 365 | -4
101: 365	Confrontation	-390	17 | -365 | -390 | -83
102: -390	Decomposition	-396	390 | -46 | -396
104: -396			
105: 352	Confrontation		396 | -352 | -10 | -352

Figure 5: Search Steps Leading to Proof 5

[3] Brown, C.E., Smolka, G.: Terminating tableaux for the basic fragment of simple type theory. In: Giese, M., Waaler, A. (eds.) TABLEAUX 2009. LNCS (LNAI), vol. 5607, pp. 138–151. Springer (Jul 2009)

[4] Korovin, K., Sticksel, C.: iProver-Eq: An instantiation-based theorem prover with equality. In: Giesl, J., Hähnle, R. (eds.) 5th International Joint Conference, IJCAR 2010. Lecture Notes in Computer Science, vol. 6173, pp. 196–202. Springer (2010)

[5] Nelson, G., Derek, Oppen, C.: Fast decision procedures based on congruence closure. Journal of the ACM 27, 356–364 (1980)