Notes about Successful Discriminator Searches

Chad E. Brown

Draft of April 14, 2020

Contents

1 Thm 1: 2 has Exactly 2 Elements **5**

1.1 Definitions and Axioms 5

1.2 Discriminator and Shallow Rules 6

1.3 Theorem 1 10

1.3.1 Search for Subgoal 1 10

1.3.2 Search for Subgoal 2 17

Chapter 1

Thm 1: 2 has Exactly 2 Elements

1.1 Definitions and Axioms

Let us first make explicit the set of definitions and axioms. We will take primitive names for the first five ordinals: 0, 1, 2, 3 and 4. Note that 0 will also be our notation for the empty set. All the axioms below obviously hold in every (reasonable) set theoretic model. The definition of atleast2 is nonstandard, but can be proven to mean that the set in question has at least two elements.

Definition 1.1. *We define $\subseteq$ to be a binary relation such that $X \subseteq Y$ holds iff (if and only if) $\forall x.x \in X \Rightarrow x \in Y$.*

Definition 1.2. *We define* disj *to be a binary relation such that* disj X Y *holds iff* $X \cap Y = 0$.

Definition 1.3. *We define* atleast2 *to be a unary predicate such that* atleast2 X *holds iff* $\exists Y \in X.X \nsubseteq \wp\, Y$.

Definition 1.4. *We define* atleast3 *to be a unary predicate such that* atleast3 X *holds iff* $\exists Y \subseteq X.X \nsubseteq Y \wedge$ atleast2 Y.

Definition 1.5. *We define* atleast4 *to be a unary predicate such that* atleast4 X *holds iff* $\exists Y \subseteq X.X \nsubseteq Y \wedge$ atleast3 Y.

Definition 1.6. *We define* atleast5 *to be a unary predicate such that* atleast5 X *holds iff* $\exists Y \subseteq X.X \nsubseteq Y \wedge$ atleast4 Y.

Definition 1.7. *We define* atleast6 *to be a unary predicate such that* atleast6 X *holds iff* $\exists Y \subseteq X.X \nsubseteq Y \wedge$ atleast5 Y.

Definition 1.8. *We define* atleast7 *to be a unary predicate such that* atleast7 X *holds iff* $\exists Y \subseteq X.X \nsubseteq Y \wedge$ atleast6 Y.

Definition 1.9. *We define* exactly2 *to be a unary predicate such that* exactly2 X *holds iff* atleast2 $X \wedge \neg$atleast3 X.

Definition 1.10. *We define* $\mathsf{exactly3}$ *to be a unary predicate such that* $\mathsf{exactly3}\ X$ *holds iff* $\mathsf{atleast3}\ X \wedge \neg \mathsf{atleast4}\ X$.

Definition 1.11. *We define* $\mathsf{exactly4}$ *to be a unary predicate such that* $\mathsf{exactly4}\ X$ *holds iff* $\mathsf{atleast4}\ X \wedge \neg \mathsf{atleast5}\ X$.

Definition 1.12. *We define* $\mathsf{exactly5}$ *to be a unary predicate such that* $\mathsf{exactly5}\ X$ *holds iff* $\mathsf{atleast5}\ X \wedge \neg \mathsf{atleast6}\ X$.

Definition 1.13. *We define* $\mathsf{exactly6}$ *to be a unary predicate such that* $\mathsf{exactly6}\ X$ *holds iff* $\mathsf{atleast6}\ X \wedge \neg \mathsf{atleast7}\ X$.

Axiom 1.1. $\forall XY.X \subseteq Y \Rightarrow Y \subseteq X \Rightarrow X = Y$.

Axiom 1.2. $\forall x.x \notin x$.

Axiom 1.3. $\forall xy.x \in y \Rightarrow y \notin x$.

Axiom 1.4. $\forall x.x \notin 0$.

Axiom 1.5. $\forall i.i \in 1 \Leftrightarrow i = 0$.

Axiom 1.6. $\forall i.i \in 2 \Leftrightarrow i = 0 \vee i = 1$.

Axiom 1.7. $\forall i.i \in 3 \Leftrightarrow i = 0 \vee i = 1 \vee i = 2$.

Axiom 1.8. $\forall i.i \in 4 \Leftrightarrow i = 0 \vee i = 1 \vee i = 2 \vee i = 3$.

Axiom 1.9. $\forall XY.Y \in \wp\ X \Leftrightarrow Y \subseteq X$.

Axiom 1.10. $\forall xy.y \in \{x\} \Leftrightarrow y = x$.

Axiom 1.11. $\forall XYz.z \in X \cup Y \Leftrightarrow z \in X \vee z \in Y$.

Axiom 1.12. $\forall XYz.z \in X \cap Y \Leftrightarrow z \in X \wedge z \in Y$.

Axiom 1.13. $\forall XYz.z \in X \setminus Y \Leftrightarrow z \in X \wedge z \notin Y$.

1.2 Discriminator and Shallow Rules

Discriminator is a first-order automated theorem prover. Unlike most first-order automated theorem provers, e.g., Prover9 [8], E [10] and Vampire [7], Discriminator does not use clause normalization, does not use resolution and does not use metavariables. Since there are no metavariables there is no need for unification (or even matching) to instantiate metavariables.

The search procedure Discriminator uses proceeds in three phases, each of which is affected by a variety of parameters. The three phases are the opening phase, the search phase and the closing phase. During the opening phase Discriminator may (optionally) perform operations such as decomposing logical operators, splitting the goal into multiple subgoals and expanding away some abbreviations. The opening phases then passes each subgoal (one at a time) to the main search phase (which we will simply call the *search phase*). The search phase performs a complete search (for first-order logic) based largely on the calculus of Satallax [5, 3, 4], although with no need for $\beta\eta$-normalization since we are in the first-order case. During search DISCRIMINATOR processes propositions and instantiations to create ground instances of rules and information about these instances are passed to the SAT solver MiniSat [6]. The closing phase is (optionally) activated if a certain number of abstract steps have been taken in the search. During the closing phase completeness is purposefully abandoned. The closing phase continues to process the propositions and instantiations generated during the search, but does not generate all the new propositions and instantiations required for completeness.

The main new feature of DISCRIMINATOR (not in Satallax) is the production and use of *shallow rules*. For the moment we consider only a special form of shallow rule: a *linear predicate shallow rule*. These will be given in the following format:

$$\Gamma|\psi \Rightarrow \varphi_1, \ldots, \varphi_n$$

Here Γ is a list of variables and ψ, φ_1, …, φ_n are formulas that may contain free variables from Γ. Here ψ is the *trigger formula* of the rule and will always be one of the following forms:

- $pt_1 \cdots t_m$ where each t_i is either a variable in Γ or of the form $fx_1 \cdots x_k$ where each x_j is a variable in Γ. Every variable in Γ must occur exactly once in $pt_1 \cdots t_m$. The predicate p may be equality.

- $\neg pt_1 \cdots t_m$ where each t_i is either a variable in Γ or of the form $fx_1 \cdots x_k$ where each x_j is a variable in Γ. Every variable in Γ must occur exactly once in $pt_1 \cdots t_m$. The predicate p may be equality.

When we are processing a formula Ψ one can easily determine if the trigger formula matches the formula and uniquely determine a substitution θ for all the variables in Γ such that $\theta(\psi) = \Psi$ without doing general first-order matching since all the variables occur (exactly once) in a shallow position of ψ. Using θ we have (potentially) new (ground) propositions $\theta(\varphi_1)$, …, $\theta(\varphi_n)$. If these propositions are new, they will be added to the priority queue to be processed later. Additionally a propositional clause recording the relationship between Ψ and $\theta(\varphi_1)$, …, $\theta(\varphi_n)$ is created and given to MiniSat.

Shallow rules can be seen as analogous to the rules for if-then-else and choice developed for Satallax [2]. In those cases the rules were sufficient to obtain completeness with no extra axioms for if-then-else or choice. In the case of DISCRIMINATOR we generate shallow rules from propositions without reason to believe the shallow rule can completely replace the source proposition during the search. However, as a heuristic we typically assign propositions low priority if they produce at least one shallow rule.

The set theory axioms from the previous section produce many linear predicate shallow rules. We record these here.

Axiom 1.1 produces the following rules:

$$x, y | x \neq y \Rightarrow x \not\subseteq y, y \not\subseteq x \tag{1.1}$$

$$x, y | x \subseteq y \Rightarrow x = y, y \not\subseteq x \tag{1.2}$$

$$x, y | y \subseteq x \Rightarrow x = y, x \not\subseteq y \tag{1.3}$$

Axiom 1.2 produces the following rule:

$$x, y | x \in y \to y \neq x \tag{1.4}$$

The reader may have expected the rule to appear as $x|x \in x \to \cdot$. However, the x occurs twice in $x \in x$, so this would not be a linear shallow predicate rule.

Axiom 1.3 produces the following rule:

$$x, y | x \in y \Rightarrow y \notin x \tag{1.5}$$

Axiom 1.4 produces the following rule:

$$x | x \in 0 \Rightarrow \cdot \tag{1.6}$$

Axiom 1.5 produces the following rules:

$$x | x \in 1 \Rightarrow x = 0 \tag{1.7}$$

$$x | x = 0 \Rightarrow x \in 1 \tag{1.8}$$

$$x | x \notin 1 \Rightarrow x \neq 0 \tag{1.9}$$

$$x | x \neq 0 \Rightarrow x \notin 1 \tag{1.10}$$

Axiom 1.6 produces the following rules:

$$x | x \in 2 \Rightarrow x = 0 \vee x = 1 \tag{1.11}$$

$$x | x = 1 \Rightarrow x \in 2 \tag{1.12}$$

$$x | x = 0 \Rightarrow x \in 2 \tag{1.13}$$

$$x | x \notin 2 \Rightarrow \neg(x = 0 \vee x = 1) \tag{1.14}$$

$$x | x \neq 1 \Rightarrow x = 0, x \notin 2 \tag{1.15}$$

$$x | x \neq 0 \Rightarrow x = 1, x \notin 2 \tag{1.16}$$

There are 8 similar rules produced from Axiom 1.7 and 10 rules produced from Axiom 1.8. We show 2 of the 8 rules produced from Axiom 1.7.

$$x | x \in 3 \Rightarrow x = 0 \vee x = 1 \vee x = 2 \tag{1.17}$$

$$x|x \neq 1 \Rightarrow x = 0, x = 2, x \notin 3 \tag{1.18}$$

We show 2 of the 10 rules produced from Axiom 1.8.

$$x|x \in 4 \Rightarrow x = 0 \vee x = 1 \vee x = 2 \vee x = 3 \tag{1.19}$$

$$x|x = 2 \Rightarrow x \in 4 \tag{1.20}$$

Axiom 1.9 generates four shallow rules:

$$x, y|y \in \wp x \Rightarrow y \subseteq x \tag{1.21}$$

$$x, y|y \notin \wp x \Rightarrow y \nsubseteq x \tag{1.22}$$

$$x, y|y \subseteq x \Rightarrow y \in \wp x \tag{1.23}$$

$$x, y|y \nsubseteq x \Rightarrow y \notin \wp x \tag{1.24}$$

Axiom 1.10 produces the following rules:

$$x, y|y \in \{x\} \Rightarrow y = x \tag{1.25}$$

$$x, y|y = x \Rightarrow y \in \{x\} \tag{1.26}$$

$$x, y|y \notin \{x\} \Rightarrow y \neq x \tag{1.27}$$

$$x, y|y \notin \{x\} \Rightarrow x \neq y \tag{1.28}$$

$$x, y|y \neq x \Rightarrow y \notin \{x\} \tag{1.29}$$

Axiom 1.11 produces the following rules:

$$x, y, z|z \in x \cup y \Rightarrow z \in x \vee z \in y \tag{1.30}$$

$$x, y, z|z \notin x \cup y \Rightarrow \neg(z \in x \vee z \in y) \tag{1.31}$$

Axiom 1.12 produces the following rules:

$$x, y, z|z \in x \cap y \Rightarrow z \in x \wedge z \in y \tag{1.32}$$

$$x, y, z|z \notin x \cap y \Rightarrow \neg(z \in x \wedge z \in y) \tag{1.33}$$

Axiom 1.13 produces the following rules:

$$x, y, z|z \in x \setminus y \Rightarrow z \in x \wedge z \notin y \tag{1.34}$$

$$x, y, z|z \notin x \setminus y \Rightarrow \neg(z \in x \wedge z \notin y) \tag{1.35}$$

1.3 Theorem 1

Theorem 1.1. $\mathsf{exactly2}\ 2$.

Let us first give a quick informal proof to give an idea what an automated theorem prover would need to do to prove this theorem from the axioms.

Proof. We need to prove two things: $\mathsf{atleast2}\ 2$ and $\neg\mathsf{atleast3}\ 2$. We first prove $\mathsf{atleast2}\ 2$. By Definition 1.3 we need to give some $Y \in 2$ such that $2 \not\subseteq \wp Y$. We take Y to be 0. We easily have $0 \in 2$ from Axiom 1.6. It remains to prove $2 \not\subseteq \wp 0$. Assume $2 \subseteq \wp 0$. We have $1 \in 2$ from Axiom 1.6 and so we must have $1 \in \wp 0$ (using Definition 1.1). By Axiom 1.9 we have $1 \subseteq 0$. We know $0 \in 1$ by Axiom 1.5 and so $0 \in 0$. The conclusion $0 \in 0$ contradicts both Axiom 1.2 and Axiom 1.4, providing two ways to complete this subproof.

Next we prove $\neg\mathsf{atleast3}\ 2$. By Definition 1.10 there must be some $Y \subseteq 2$ such that $2 \not\subseteq Y$ and $\mathsf{atleast2}\ Y$. By Definition 1.1 there must be some $a \in 2$ with $a \notin Y$. By Definition 1.9 there must be some $b \in Y$ with $Y \not\subseteq \wp b$. By Definition 1.1 there must be some $c \in Y$ with $c \notin \wp b$. Since $a \notin Y$, $b \in Y$ and $c \in Y$, we know $a \neq b$ and $a \neq c$. Using $c \notin \wp b$ we can also argue $b \neq c$ (using Axioms 1.9 and 1.1). Since $Y \subseteq 2$, we know $b \in 2$ and $c \in 2$. Applying Axiom 1.6 with each of a, b and c we have eight cases, each of which is in contradiction with $a \neq b$, $a \neq c$ and $b \neq c$. □

Let us now informally describe how DISCRIMINATOR searches for and finds a proof of the theorem. DISCRIMINATOR uses the opening to recognize that certain predicates can be considered abbreviations, including $\subseteq$, $\mathsf{exactly2}$, $\mathsf{atleast2}$ and $\mathsf{atleast3}$. These are expanded during the opening and the subgoal is split into two subgoals, corresponding to proving $\mathsf{atleast2}\ 2$ and $\neg\mathsf{atleast3}\ 2$. We describe the search for these two subgoals separately below.

1.3.1 Search for Subgoal 1

The first subgoal asserts the axioms above (with definitional axioms removed and with abbreviations expanded in the remaining axioms) and the additional proposition:

$$\neg\exists Y \in 2.2 \hat{\not\subseteq} \wp Y$$

where we write $s \hat{\subseteq} t$ as notation for $\forall z.z \in s \to z \in t$ and $s \hat{\not\subseteq} t$ for the negation. (Recall that the predicate $\subseteq$ has been expanded away already.) These asserted propositions are analyzed to see which produce shallow rules. In this case *every* asserted proposition produces at least one shallow rule. We saw shallow rules for the axioms in Section 1.2 and these remain mostly the same. The exceptions are caused by the expansion of $\subseteq$ in Axioms 1.1 and 1.9. Due to this expansion the modified Axiom 1.1 yields only the shallow rule

$$x, y | x \neq y \Rightarrow x \hat{\not\subseteq} y, y \hat{\not\subseteq} x \tag{1.36}$$

and the modified Axiom 1.9 yields only the shallow rules

$$x, y | y \in \wp x \Rightarrow y \hat{\subseteq} x \tag{1.37}$$

$$x, y | y \notin \wp x \Rightarrow y \hat{\nsubseteq} x \tag{1.38}$$

A legend assigning numbers to the propositions considered during the search is given in Table 1.1. A figure showing the steps leading to the proof is given in Figure 1.1. For each asserted formula a unit clause is sent to MiniSat. For example, $\boxed{14}$ represents the proposition

$$\exists Y \in 2.2 \hat{\nsubseteq} \wp Y$$

and the unit clause $\boxed{-14}$ (representing the negation of the conclusion) is sent to MiniSat. For $i \in \{1, \dots, 13\}$, $\boxed{i}$ represents one of the axioms (with $\subseteq$ expanded) and the unit clause $\boxed{i}$ is sent to MiniSat. The ones that play a role are $\boxed{5}$, $\boxed{7}$, $\boxed{8}$, $\boxed{11}$, $\boxed{12}$ and $\boxed{13}$ (see Table 1.1).

The negated conclusion

$$\neg \exists Y \in 2.2 \hat{\nsubseteq} \wp Y$$

yields two shallow rules:

$$x | x \in 2 \Rightarrow 2 \hat{\subseteq} \wp x \tag{1.39}$$

$$x, y | x \notin \wp y \Rightarrow x \notin 2, y \notin 2 \tag{1.40}$$

Due to Shallow Rule 1.40, whenever we process a proposition of the form $s \notin \wp t$ we will generate the propositions $s \notin 2$ and $t \notin 2$ along with the propositional information relating these three propositions.

The story of how DISCRIMINATOR proceeds to search for this subgoal is somewhat roundabout.[1]

When a proposition produces at least one shallow rule we often (heuristically) give a lower priority to processing the source proposition. In this case since all the propositions produce shallow rules, all the initial propositions have a low priority. For this reason the search will begin by processing instantiations.

The complete calculus for DISCRIMINATOR only requires instantiating with terms that occur on the left or right hand side of a disequation (*discriminating terms*) or with a default term if no term is discriminating. In this case there are no disequations (yet) so no term is discriminating. One option in DISCRIMINATOR is to seed the initial set of instantiations with all (ground) subterms of the problem. The only ground subterms of the propositions asserted in this subgoal are 0, 1, 2, 3 and 4. There are no discriminating terms in the subgoal, so these are the only instantiations in the beginning.

Processing an instantiation before processing any positive universally quantified propositions or negative existentially quantified propositions only has a bookkeeping effects. The search begins by processing these five instantiations and then looking for one of the (low priority) propositions to process. Steps 0 (processing instantiation 2), 2 (processing instantation 1) and 3 (processing instantiation 0) are the first steps shown in

[1]This is a polite way to refer to what appears to be a drunkard's walk.

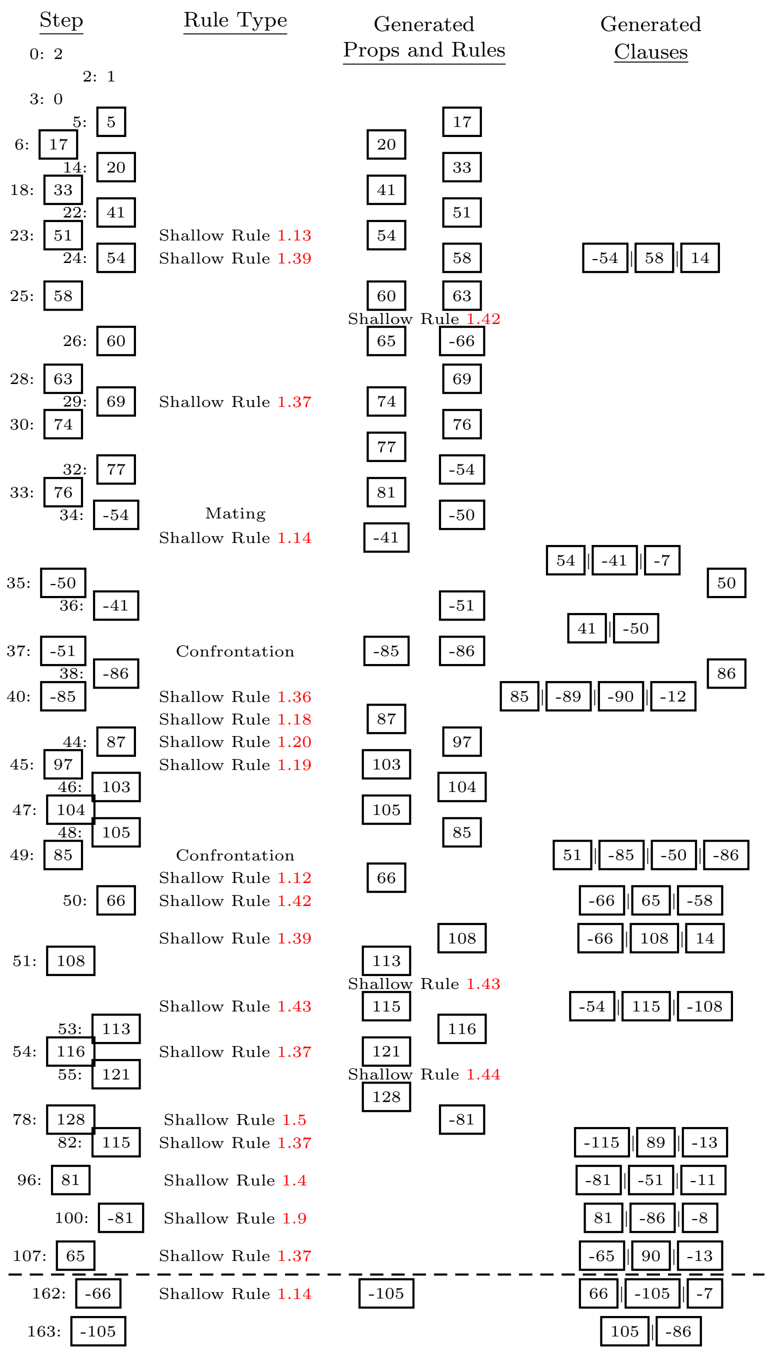
Step
Rule Type
Generated Props and Rules
Generated Clauses
0: 2
2: 1
3: 0
5: 5 | 17
6: 17 | 20
14: 20 | 33
18: 33 | 41
22: 41 | 51
23: 51 | Shallow Rule 1.13 | 54
24: 54 | Shallow Rule 1.39 | 58 | -54 | 58 | 14
25: 58 | 60 | 63
Shallow Rule 1.42
26: 60 | 65 | -66
28: 63 | 69
29: 69 | Shallow Rule 1.37 | 74
30: 74 | 76
77
32: 77 | -54
33: 76 | 81
34: -54 | Mating | -50
Shallow Rule 1.14 | -41
54 | -41 | -7
35: -50 | 50
36: -41 | -51
41 | -50
37: -51 | Confrontation | -85 | -86
38: -86 | 86
40: -85 | Shallow Rule 1.36 | 85 | -89 | -90 | -12
Shallow Rule 1.18 | 87
44: 87 | Shallow Rule 1.20 | 97
45: 97 | Shallow Rule 1.19 | 103
46: 103 | 104
47: 104 | 105
48: 105 | 85
49: 85 | Confrontation | 51 | -85 | -50 | -86
Shallow Rule 1.12 | 66
50: 66 | Shallow Rule 1.42 | -66 | 65 | -58
Shallow Rule 1.39 | 108 | -66 | 108 | 14
51: 108 | 113
Shallow Rule 1.43
Shallow Rule 1.43 | 115 | -54 | 115 | -108
53: 113 | 116
54: 116 | Shallow Rule 1.37 | 121
55: 121 | Shallow Rule 1.44
128
78: 128 | Shallow Rule 1.5 | -81
82: 115 | Shallow Rule 1.37 | -115 | 89 | -13
96: 81 | Shallow Rule 1.4 | -81 | -51 | -11
100: -81 | Shallow Rule 1.9 | 81 | -86 | -8
107: 65 | Shallow Rule 1.37 | -65 | 90 | -13
162: -66 | Shallow Rule 1.14 | -105 | 66 | -105 | -7
163: -105 | 105 | -86

Figure 1.1: Search Steps Leading to Proof of Subgoal 1

$\boxed{5}$	$\forall x.x \in 4 \leftrightarrow x = 0 \vee x = 1 \vee x = 2 \vee x = 3$	$\boxed{7}$	$\forall i.i \in 2 \Leftrightarrow i = 0 \vee i = 1$
$\boxed{8}$	$\forall i.i \in 1 \Leftrightarrow i = 0$	$\boxed{11}$	$\forall x.x \notin x$
$\boxed{12}$	$\forall XY.X \hat{\subseteq} Y \Rightarrow Y \hat{\subseteq} X \Rightarrow X = Y$	$\boxed{13}$	$\forall XY.Y \in \wp\, X \Leftrightarrow Y \hat{\subseteq} X$
$\boxed{14}$	$\exists Y \in 2.2 \hat{\nsubseteq} \wp Y$	$\boxed{17}$	$(1 \in 4 \leftrightarrow 1 = 0 \vee 1 = 1 \vee 1 = 2 \vee 1 = 3)$
$\boxed{20}$	$1 = 0 \vee 1 = 1 \vee 1 = 2 \vee 1 = 3$	$\boxed{33}$	$1 = 0 \vee 1 = 1 \vee 1 = 2$
$\boxed{41}$	$1 = 0 \vee 1 = 1$	$\boxed{50}$	$1 = 1$
$\boxed{51}$	$1 = 0$	$\boxed{54}$	$1 \in 2$
$\boxed{58}$	$2 \hat{\subseteq} \wp 1$	$\boxed{60}$	$0 \in 2 \rightarrow 0 \in \wp 1$
$\boxed{63}$	$2 \in 2 \rightarrow 2 \in \wp 1$	$\boxed{65}$	$0 \in \wp 1$
$\boxed{66}$	$0 \in 2$	$\boxed{69}$	$2 \in \wp 1$
$\boxed{74}$	$2 \hat{\subseteq} 1$	$\boxed{76}$	$0 \in 2 \rightarrow 0 \in 1$
$\boxed{77}$	$1 \in 2 \rightarrow 1 \in 1$	$\boxed{81}$	$0 \in 1$
$\boxed{85}$	$0 = 1$	$\boxed{86}$	$0 = 0$
$\boxed{87}$	$0 = 2$	$\boxed{89}$	$1 \hat{\subseteq} 0$
$\boxed{90}$	$0 \hat{\subseteq} 1$	$\boxed{97}$	$0 \in 4$
$\boxed{103}$	$0 = 0 \vee 0 = 1 \vee 0 = 2 \vee 0 = 3$	$\boxed{104}$	$0 = 0 \vee 0 = 1 \vee 0 = 2$
$\boxed{105}$	$0 = 0 \vee 0 = 1$	$\boxed{108}$	$2 \hat{\subseteq} \wp 0$
$\boxed{113}$	$2 \in 2 \rightarrow 2 \in \wp 0$	$\boxed{115}$	$1 \in \wp 0$
$\boxed{116}$	$2 \in \wp 0$	$\boxed{121}$	$2 \hat{\subseteq} 0$
$\boxed{128}$	$1 \in 0$		

Table 1.1: Propositions in Subgoal 1 Search

Figure 1.1. No rules apply and no propositions, instantiations or clauses are generated. The side effect (not shown in Figure 1.1) is that these instantiations are now available when future propositions are processed.

The first proposition processed is Axiom 1.8 ($\boxed{5}$ in Table 1.1 and Figure 1.1). This is instantiated with each of 0, 1, 2, 3 and 4. The instance that will play a role in the successful proof is the one given by 1:

$$\boxed{17} : 1 \in 4 \leftrightarrow 1 = 0 \vee 1 = 1 \vee 1 = 2 \vee 1 = 3.$$

This may seem surprising since this seems to have nothing directly to do with proving 2 has at least 2 elements. However, processing the proposition and subsequent subformulas leads to processing the propositions $1 = 0$ and $1 = 1$. Since the proposition above is an equivalence, $1 = 0$ and $1 = 1$ occur both positively and negatively as subformulas. When processing $1 = 0$ (a positive proposition), the proposition $1 \in 2$ is generated due to Shallow Rule 1.13. This illustrates the somewhat unexpected way propositions are sometimes generated. If 1 were equal to 0, then 1 would be in 2 so we consider $1 \in 2$. Of course, $1 \in 2$ is both true and a reasonable proposition to consider, even though it was generated by considering the false proposition $1 = 0$. The reader should keep in mind that "processing" a proposition has nothing to do with whether the proposition is true or false, but only to do with determining its relationship to other propositions.

When processing $1 \in 2$, Shallow Rule 1.39 leads Discriminator to generate $2 \hat{\subseteq} \wp 1$. Again, this is a somewhat unexpected way to obtain $2 \hat{\subseteq} \wp 1$ since it is true, but Discriminator is using the negation of atleast2 2 to obtain it. In Table 1.1 and Figure 1.1,

$1 \in 2$ is $\boxed{54}$ and $2\hat{\subseteq}\wp 1$ is $\boxed{58}$. Shallow Rule 1.39 also yields the propositional clause $\boxed{-54}\|\boxed{58}\|\boxed{14}$ to send to MiniSat. (This is the first clause generated during the search that will play a role in the unsatisfiability of the clauses sent to MiniSat.) Recall that $\boxed{-14}$ is a unit clause from the initialization of the search.

Recall that $2\hat{\subseteq}\wp 1$ is notation for $\forall z.z \in 2 \to z \in \wp 1$. When Discriminator processes this universally quantified formula it generates the new shallow rules

$$y, z | z \notin \wp y \Rightarrow y \neq 1, z \notin 2 \tag{1.41}$$

and

$$z | z \in 2 \Rightarrow z \in \wp 1 \tag{1.42}$$

Only this second rule will play a role in the eventual proof. Discriminator also instantiates the formula with all the five instantations 0, 1, 2, 3 and 4. Two instances play a role in the proof. The instance with 0 (i.e., $0 \in 2 \to 0 \in \wp 1$, abbreviated as $\boxed{60}$) plays a direct role in the proof. The instance with 2 (i.e., $2 \in 2 \to 2 \in \wp 1$, abbreviated as $\boxed{63}$) plays an indirect, but still important, role during the search. Processing the instance with 0 (in Step 26) generates propositions $0 \notin 2$ and $0 \in \wp 1$, and both will participate in the proof. Processing the instance with 2 generates propositions $2 \notin 2$ and $2 \in \wp 1$, but only $2 \in \wp 1$ will participate in the proof.

Step 29 in Figure 1.1 corresponds to processing $2 \in \wp 1$. Shallow Rule 1.37 generates the proposition $2\hat{\subseteq}1$. Processing (in Step 30) this yields $0 \in 2 \to 0 \in 1$ ($\boxed{76}$) and $1 \in 2 \to 1 \in 1$ ($\boxed{77}$) leading Discriminator to generate propositions $0 \in 1$ and $1 \notin 2$. (Recall above Discriminator had already processed the positive proposition $1 \in 2$, see Step 24 in Figure 1.1.) Discriminator processes $1 \notin 2$ in Step 34, delaying processing $0 \in 1$ until Step 96.

Two important actions happen when processing $1 \notin 2$. Since $1 \in 2$ was processed before, it is mated with $1 \notin 2$[2] producing disequations $1 \neq 1$ and $2 \neq 2$. In addition Shallow Rule 1.14 gives that if $1 \notin 2$, then $\neg(1 = 0 \vee 1 = 1)$. This results in generating the proposition $\neg(1 = 0 \vee 1 = 1)$ ($\boxed{-41}$) and the clause $\boxed{54}\|\boxed{-41}\|\boxed{-7}$ sent to MiniSat. Since $\boxed{7}$ is a unit clause from the initialization, we must have either $1 \in 2$ or $\neg(1 = 0 \vee 1 = 1)$.

Processing $1 \neq 1$ yields the unit clause $\boxed{50}$ representing the fact that $1 = 1$ (by an equality rule of the calculus). Processing $\neg(1 = 0 \vee 1 = 1)$ yields an obvious relationship with $1 = 1$ represented by the clause $\boxed{41}\|\boxed{-50}$. Essentially at this point we know $1 = 0 \vee 1 = 1$ holds, and so $1 \in 2$ holds. Using the information represented by the clause $\boxed{-54}\|\boxed{58}\|\boxed{14}$ we can further infer that $2\hat{\subseteq}\wp 1$ must hold.

[2] This is weird. It's almost certainly unnecessary to mate a proposition with its literal complement. In this case it happens to help. Since writing this description I changed the code to prevent mating formulas with their signed counterpart and prevent confrontations with their signed counterparts, with parameters to allow such "self matings" and "self confrontations." The proof described here depends on allowing self matings and self confrontations, but since changing the code alternative proofs have (allegedly) been found that do not require self matings or self confrontations. I need to look into these more closely to make sure there is nothing suspicious about the new proofs.

When the proposition $\neg(1 = 0 \vee 1 = 1)$ was processed, the proposition $1 \neq 0$ was generated. In Step 37, $1 \neq 0$ is processed. Recall that the positive proposition $1 = 0$ was processed above. When $1 \neq 0$ is processed the disequation $1 = 0$ is *confronted* by the equation $1 = 0$[3] yielding as a side effect the disequations $0 \neq 1$ and $0 \neq 0$. Processing $0 \neq 0$ (in Step 38) produces the unit clause $\boxed{86}$ form MiniSat, representing the truth of $0 = 0$.

When $0 \neq 1$ is processed (in Step 40) two relevant independent actions occur. Shallow Rule 1.36 yields the clause $\boxed{85}\|\boxed{-89}\|\boxed{-90}\|\boxed{-12}$ recording that if $0 \neq 1$, then either $0 \hat{\not\subseteq} 1$ or $1 \hat{\not\subseteq} 1$. Shallow Rule 1.18 generates the proposition $0 = 2$. When $0 = 2$ is processed Shallow Rule 1.20 generates the proposition $0 \in 4$. When $0 \in 4$ is processed the proposition $0 = 0 \vee 0 = 1 \vee 0 = 2 \vee 0 = 3$ is generated. By processing it and its subformulas eventually $0 = 1$ is processed.

Two important actions occur when $0 = 1$ is processed in Step 49. First $0 = 1$ confronts $1 \neq 0$ (processed earlier) to record the propositional information that both cannot be true. This is represented by the clause $\boxed{51}\|\boxed{-85}\|\boxed{-50}\|\boxed{-86}$. Second Shallow Rule 1.12 is triggered and generates the propositional formula $0 \in 2$.

When $0 \in 2$ is processed in Step 50 two shallow rules apply. Shallow Rule 1.42 produces the clause $\boxed{-66}\|\boxed{65}\|\boxed{-58}$, essentially representing that if $0 \in 2$, then $0 \in \wp 1$. Shallow Rule 1.39 produces the new proposition $2 \hat{\subseteq} \wp 0$ and the clause $\boxed{-66}\|\boxed{108}\|\boxed{14}$. Since $\boxed{-14}$ is a unit clause from the initialization, this clause means that if $0 \in 2$, then $2 \hat{\subseteq} \wp 0$.

Processing $2 \hat{\subseteq} \wp 0$ in Step 51 yields the proposition $2 \in 2 \rightarrow 2 \in \wp 0$ ($\boxed{113}$) by instantiating with 2. In addition the following new shallow rule is created:

$$x | x \in 2 \Rightarrow x \in \wp 0 \tag{1.43}$$

After creating this new shallow rule DISCRIMINATOR checks if it is triggered by any previously processed propositions. In fact $1 \in 2$ (processed in Step 24) triggers the rule leading to the new proposition $1 \in \wp 0$ ($\boxed{115}$) and the clause $\boxed{-54}\|\boxed{115}\|\boxed{-108}$.

Processing $2 \in 2 \rightarrow 2 \in \wp 0$ leads to processing $2 \in \wp 0$. Shallow Rule 1.37 generates $2 \hat{\subseteq} 0$. Processing $2 \hat{\subseteq} 0$ creates the following new shallow rule:

$$x | x \in 2 \Rightarrow x \in 0 \tag{1.44}$$

This new shallow rule is triggered by $1 \in 2$ generating the proposition $1 \in 0$. Processing $1 \in 0$ (in Step 78) generates the proposition $0 \notin 1$ by Shallow Rule 1.5.

In Step 82 $1 \in \wp 0$ is processed. Shallow rule 1.37 yields clause $\boxed{-115}\|\boxed{89}\|\boxed{-13}$ representing that if $1 \in \wp 0$, then $1 \hat{\subseteq} 0$ (since $\boxed{13}$ is a unit clause from the initial assumptions).

In Step 96 $0 \in 1$ (generated in Step 33) is finally processed. Shallow Rule 1.4 yields the clause $\boxed{-81}\|\boxed{-51}\|\boxed{-11}$ meaning that if $0 \in 1$, then $1 \neq 0$.

[3]Confrontation is essentially the Mating rule for equality, making use of symmetry of equality. See [2].

In Step 100 $0 \notin 1$ (generated in Step 78) is processed. Shallow Rule 1.9 yields the clause $\boxed{81}\|\boxed{-86}\|\boxed{-8}$ essentially giving that if $0 \notin 1$, then $0 \neq 0$. Since we already know $0 = 0$ from Step 38, we now know $0 \in 1$. Combining this with the information from Step 96, we know $1 \neq 0$.

In Step 107 $0 \in \wp 1$ (generated in Step 26) is processed. Shallow Rule 1.37 yields the clause $\boxed{-65}\|\boxed{90}\|\boxed{-13}$, representing that if $0 \in \wp 1$, then $0 \hat{\subseteq} 1$.

Discriminator was called with a middle abstract time limit of 128. As a consequence Discriminator continues searching as usual by processing propositions for 20 more steps. After these steps Discriminator enters the closing phase. This is represented by the dashed line in Figure 1.1. In the closing phase, Discriminator continues to process propositions (and instantiations, if there are any) on the priority queue but no longer generates all new propositions and adds them to the priority queue for later processing. (This is, of course, very incomplete.) Depending on parameter settings Discriminator might not add any new propositions to the priority queue. For this search a parameter is set so that Discriminator will add a new proposition to the priority queue only if it is an implication, a conjunction, a disjunction or an equivalence (or the negation of one of these).

During the closing phase only two steps will be required to complete the proof. In Step 162 the formula $0 \notin 2$ (generated in Step 26) is processed. Shallow Rule 1.14 yields the proposition $\neg(0 = 0 \vee 0 = 1)$ and the clause $\boxed{66}\|\boxed{-105}\|\boxed{-7}$, representing that if $0 \notin 2$, then $\neg(0 = 0 \vee 0 = 1)$. Since $\neg(0 = 0 \vee 0 = 1)$ is the negation of a disjunction, it is added to the priority queue.

In the final step, Step 163, $\neg(0 = 0 \vee 0 = 1)$ is processed. The clause $\boxed{105}\|\boxed{-86}$ is sent to MiniSat and MiniSat reports unsatisfiability, completing the proof of the subgoal.

Before ending the discussion of this subgoal, let us reconsider the argument by using the MiniSat clauses to indicate how a refutation is reached. In the beginning there are 14 unit clauses: $\boxed{-14}$ and $\boxed{i}$ for $i \in \{1, \ldots, 13\}$. Two more unit clauses are produced during the search: 50 (for $1 = 1$) and 86 (for $0 = 0$). These unit clauses can be propagated (leading directly to unsatisfiability) as follows. The clause $\boxed{41}\|\boxed{-50}$ gives the unit $\boxed{41}$ (for $1 = 0 \vee 1 = 1$). The clause $\boxed{54}\|\boxed{-41}\|\boxed{-7}$ gives $\boxed{54}$ (for $1 \in 2$). The clause $\boxed{-54}\|\boxed{58}\|\boxed{14}$ gives $\boxed{58}$ (for $2 \hat{\subseteq} \wp 1$).

The clause $\boxed{81}\|\boxed{-86}\|\boxed{-8}$ gives the unit $\boxed{81}$ (for $0 \in 1$) and the clause $\boxed{105}\|\boxed{-86}$ gives the unit $\boxed{105}$ (for $0 = 0 \vee 0 = 1$). Now $\boxed{66}\|\boxed{-105}\|\boxed{-7}$ gives the unit $\boxed{66}$ (for $0 \in 2$). Now $\boxed{-66}\|\boxed{65}\|\boxed{-58}$ gives the unit $\boxed{65}$ (for $0 \in \wp 1$) and $\boxed{-66}\|\boxed{108}\|\boxed{14}$ gives the unit $\boxed{108}$ (for $2 \hat{\subseteq} \wp 0$). Note that $2 \hat{\subseteq} \wp 0$ is actually false, but it is implied by the negation of the conclusion we with to prove (represented by $\boxed{-14}$).

At this point $\boxed{-54}\|\boxed{115}\|\boxed{-108}$ gives the unit $\boxed{115}$ (for $1 \in \wp 0$). Now $\boxed{-115}\|\boxed{89}\|\boxed{-13}$ gives the unit $\boxed{89}$ (for $1 \hat{\subseteq} 0$). Additionally $\boxed{-65}\|\boxed{90}\|\boxed{-13}$ gives the unit $\boxed{90}$ (for $0 \hat{\subseteq} 1$). These last two units combine with $\boxed{85}\|\boxed{-89}\|\boxed{-90}\|\boxed{-12}$ to give $\boxed{85}$ (for $0 = 1$). Now $\boxed{51}\|\boxed{-85}\|\boxed{-50}\|\boxed{-86}$ gives $\boxed{51}$ (for $1 = 0$).

We are now in conflict with the clause $\boxed{-81}\|\boxed{-51}\|\boxed{-11}$. That is, we have derived

$0 \in 1$ and $1 = 0$ above, but also derived that both of these cannot be true.

1.3.2 Search for Subgoal 2

When considering the second subgoal the opening phase can break the problem down further after expanding atleast3, atleast2 and $\subseteq$. As a consequence the opening creates four fresh eigenvariables Y, a, b and c. In addition to the axioms (with abbreviations expanded), the search includes the following propositions:

1. $Y \hat{\subseteq} 2$
2. $a \in 2$,
3. $b \in Y$,
4. $c \in Y$,
5. $c \notin \wp b$ and
6. $a \notin Y$.

The proposition $Y \hat{\subseteq} 2$ produces the following shallow rule:

$$x | x \in Y \Rightarrow x \in 2 \tag{1.45}$$

The other propositions above do not produce a shallow rule and so they are given priority when the search begins. Other shallow rules that are produced from the axioms (with $\subseteq$ expanded) and are used in the search are Shallow Rule 1.11 and 1.37. The important propositions used in the search are given in Table 1.2 and the steps leading to a proof are shown in Figure 1.2. MiniSat is initially given unit clauses corresponding to the axioms and assumptions of the subgoal. These unit clauses are 17 positive unit clauses $\boxed{i}$ for $i \in \{1, \ldots, 17\}$ and the two negative unit clauses $\boxed{\text{-18}}$ and $\boxed{\text{-19}}$.

In Step 1 $c \in Y$ is processed. Shallow Rule 1.45 generates the formula $c \in 2$ and a clause that essentially tells us that $c \in 2$ is true. In Step 2 $c \in 2$ is processed. Shallow Rule 1.11 generates the formula $c = 0 \vee c = 1$ and a clause that essentially tells us $c = 0 \vee c = 1$ is true. In Step 5 $c = 0 \vee c = 1$ is processed. The disjunction rule gives us the formulas $c = 0$ and $c = 1$ and a propositional clause telling us one of these must be true. Although the formulas $c = 0$ and $c = 1$ are generated, we do not list them as generated in Figure 1.2 since they will not need to be processed in order to obtain a proof. (They are both processed during the search, but this does not contribute to the ultimate success.)

In Step 6 $c \notin \wp b$ is processed. Shallow Rule 1.37 gives the formula $c \hat{\not\subseteq} b$ and a clause that essentially means $c \hat{\not\subseteq} b$ is true. We return to process in $c \hat{\not\subseteq} b$ at Step 27.

In Step 11 $a \notin Y$ is processed and is mated with $c \in Y$. This generates two disequations $c \neq a$ and $Y \neq Y$, as well as a clause essentially indicating that either

Step	Rule Type	Generated Props	Generated Clauses
1: 15	Shallow Rule 1.45	22	-15 ‖ 22 ‖ -12
2: 22	Shallow Rule 1.11	25	-22 ‖ 25 ‖ -7
5: 25			-25 ‖ 33 ‖ 32
6: -18	Shallow Rule 1.36	-37	18 ‖ -37 ‖ -17
11: -19	Mating	-45 -47	-15 ‖ 19 ‖ -47 ‖ -45
12: -47			47
27: -37		-95	37 ‖ -95
28: -95		96 -97	95 ‖ 96
			95 ‖ -97
29: 96	Mating	-101	
33: -101			
39: 14	Mating	-133	-14 ‖ 19 ‖ -47 ‖ -133
	Shallow Rule 1.45	139	-14 ‖ 139 ‖ -12
40: -133			
45: 139	Shallow Rule 1.11	165	-139 ‖ 165 ‖ -7
46: 165		142 146	-165 ‖ 142 ‖ 146
47: 142	Confrontation	-167	133 ‖ -142 ‖ -167 ‖ -141
	Confrontation	-168	
48: -167			167
58: 146	Confrontation		133 ‖ -146 ‖ -167 ‖ -145
60: 13	Mating	-194	
	Shallow Rule 1.11	199	-13 ‖ 199 ‖ -7
61: -194			194
62: 199		141 145	-199 ‖ 141 ‖ 145
63: 145			
71: 141			
75: -97	Mating	-226	-96 ‖ 97 ‖ -168 ‖ -226
78: -226			226
164: -45	Confrontation		45 ‖ -141 ‖ -33 ‖ -194
			45 ‖ -145 ‖ -32 ‖ -194
176: -168	Confrontation		168 ‖ -146 ‖ -32 ‖ -167
			168 ‖ -142 ‖ -33 ‖ -167

Figure 1.2: Search Steps Leading to Proof of Subgoal 2

$\boxed{7}$	$\forall i.i \in 2 \Leftrightarrow i = 0 \vee i = 1$	$\boxed{12}$	$Y \hat{\subseteq} 2$
$\boxed{13}$	$a \in 2$	$\boxed{14}$	$b \in Y$
$\boxed{15}$	$c \in Y$	$\boxed{17}$	$\forall XY.Y \in \wp X \Leftrightarrow Y \hat{\subseteq} X$
$\boxed{18}$	$c \in \wp b$	$\boxed{19}$	$a \in Y$
$\boxed{22}$	$c \in 2$	$\boxed{25}$	$c = 0 \vee c = 1$
$\boxed{32}$	$c = 1$	$\boxed{33}$	$c = 0$
$\boxed{37}$	$c \hat{\subseteq} b$	$\boxed{45}$	$c = a$
$\boxed{47}$	$Y = Y$	$\boxed{77}$	$1 = 1$
$\boxed{95}$	$d \in c \to d \in b$	$\boxed{96}$	$d \in c$
$\boxed{97}$	$d \in b$	$\boxed{101}$	$d = c$
$\boxed{133}$	$b = a$	$\boxed{139}$	$b \in 2$
$\boxed{141}$	$a = 0$	$\boxed{142}$	$b = 0$
$\boxed{145}$	$a = 1$	$\boxed{146}$	$b = 1$
$\boxed{165}$	$b = 0 \vee b = 1$	$\boxed{167}$	$b = b$
$\boxed{168}$	$c = b$	$\boxed{194}$	$a = a$
$\boxed{199}$	$a = 0 \vee a = 1$	$\boxed{226}$	$d = d$

Table 1.2: Propositions in Subgoal 2 Search

$c \neq a$ or $Y \neq Y$ must be true. In Step 12 $Y \neq Y$ is processed resulting in a unit clause corresponding to the truth of $Y = Y$. Hence we now know $c \neq a$.

In Step 27 $c \hat{\nsubseteq} b$ is processed. Recall that this proposition is actually

$$\neg(\forall x.x \in c \to x \in b).$$

When DISCRIMINATOR processes an existentially quantified proposition or a negated universally quantified proposition, it creates a fresh eigenvariable as a witness. Let us call this fresh eigenvariable d. The new proposition $\neg(d \in c \to d \in b)$ is generated as well as a clause that essentially says $d \in c \to d \in b$ is false. Note that this is different to the way such quantifiers are treated by most first-order automated theorem provers. Most first-order automated theorem provers eliminate these quantifiers using Skolem functions in a preprocessing step. The treatment of such quantifiers by DISCRIMINATOR is even different from other automated theorem provers that do not (always) use Skolem functions. For example TPS [1] (optionally) uses *selected variables* and, in order to maintain soundness, ensures acyclicity of an ordering relating selected variables and existential variables (see [9] for more information). DISCRIMINATOR does not use existential variables and so there is no need to maintain such an ordering.

In Step 28 $\neg(d \in c \to d \in b)$ is processed generating the propositions $d \in c$ and $d \notin b$ along with clauses recording that $d \in c$ is true and $d \notin b$ is true. In Step 29 $d \in c$ is processed and mated with $c \in \wp b$ generating the disequation $d \neq c$. In Step 33 $d \neq c$ is processed making it available for future confrontations.

In Step 39 $b \in Y$ is processed resulting in two important actions. First $b \in Y$ is mated with $a \notin Y$ resulting in the disequation $b \neq a$ and a clause that allows us to conclude that $b \neq a$ is true. Second the Shallow Rule 1.45 gives us the proposition $b \in 2$ and a clause allowing us to conclude $b \in 2$ is true. In Step 40 $b \neq a$ is processed, but this only has the bookkeeping effect of making $b \neq a$ available for future confrontations with positive equations. In Step 45 $b \in 2$ is processed and Shallow Rule 1.11 generates the disjunction $b = 0 \vee b = 1$ and a clause allowing us to conclude this disjunction is true. In Step 46 the disjunction is processed giving us equations $b = 0$ and $b = 1$ and a clause meaning one of these equations must be true.

In Step 47 the equation $b = 0$ is processed and confronts two previously processed disequations: $b \neq a$ and $d \neq c$. The confrontation with $b \neq a$ produces the disequation $b \neq b$ and a clause telling us that if $b = 0$ is true, then $a = 0$ is false or $b = b$ is false. The confrontation with $d \neq c$ produces the disequation $c \neq b$ which will be processed in the final step of the proof. In Step 48 $b \neq b$ is processed giving a unit clause corresponding to the truth of $b = b$. Now we know that if $b = 0$ is true, then $a = 0$ is false.

In Step 58 the equation $b = 1$ is processed and confronts $b \neq a$. This generates a clause that essentially says that if $b = 1$, then $a \neq 1$.

In Step 60 $a \in 2$ is processed giving two relevant results. Mating with $a \notin Y$ (processed in Step 11) generates the new disequation $a \neq a$. Shallow Rule 1.11 generates the disjunction $a = 0 \vee a = 1$ and a clause saying the disjunction is true. Step 61 processes $a \neq a$ producing the unit clause saying $a = a$ is true. Step 62 processes $a = 0 \vee a = 1$ producing the propositions $a = 0$ and $a = 1$ and producing a clause saying one of these propositions must be true. Steps 63 and 71 processes these two equations making them available for later confrontations.

Step 75 processes $d \notin b$ (from Step 28). Mating $d \notin b$ with $d \in c$ produces the disequation $d \neq d$ and a clause saying that either $c \neq b$ or $d \neq d$ must be true. Step 78 processes $d \neq d$ giving the unit clause saying $d = d$ is true. At this point we know $c \neq b$ must be true.

Discriminator continues searching as above until 128 steps have been reached. As with the first subgoal at Step 128 the closing phase begins. After this point the only new propositions that will be added to the priority queue for future processing are implications, conjunctions, disjunctions or equivalences (or negations of one of these). During the closing phase two relevant steps will complete the proof.

Before describing these final two steps, let us summarize what we know so far. After Step 5 we know either $c = 0$ or $c = 1$. After Step 46 we know either $b = 0$ or $b = 1$. After Step 48 we know if $b = 0$, then $a \neq 0$. After Step 58 we know if $b = 1$, then $a \neq 1$. After Step 62 we know either $a = 0$ or $a = 1$. The only remaining possibilities to be ruled are when c has the same value as either a or b. We know $c \neq a$ from Step 12 and $c \neq b$ from Step 78, but the propositional clauses so far are still satisfiable.

In Step 164 $c \neq a$ is processed and is confronted by $a = 0$ and $a = 1$. The confrontation with $a = 0$ yields a clause meaning that if $c = 0$, then $a \neq 0$. The confrontation with $a = 1$ yields a clause meaning that if $c = 1$, then $a \neq 1$.

In the final step, Step 176, $c \neq b$ is processed and is confronted by $b = 0$ and $b = 1$. The confrontation with $b = 0$ yields a clause meaning that if $c = 0$, then $b \neq 0$. The confrontation with $b = 1$ yields a clause meaning that if $c = 1$, then $b \neq 1$. The set of propositional clauses is now unsatisfiable, completing the proof.

Bibliography

[1] Andrews, P.B., Brown, C.E.: TPS: A hybrid automatic-interactive system for developing proofs. Journal of Applied Logic 4(4), 367–395 (2006)

[2] 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

[3] Brown, C.E.: Satallax: An automatic higher-order prover. In: Gramlich, B., Miller, D., Sattler, U. (eds.) IJCAR. LNCS, vol. 7364, pp. 111–117. Springer (2012)

[4] Brown, C.E.: Reducing higher-order theorem proving to a sequence of sat problems. Journal of Automated Reasoning 51(1), 57–77 (Mar 2013)

[5] Brown, C.E., Smolka, G.: Analytic tableaux for simple type theory and its first-order fragment. Logical Methods in Computer Science 6(2) (Jun 2010)

[6] Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT. LNCS, vol. 2919, pp. 502–518. Springer (2003)

[7] Kovács, L., Voronkov, A.: First-order theorem proving and Vampire. In: Sharygina, N., Veith, H. (eds.) CAV. LNCS, vol. 8044, pp. 1–35. Springer (2013)

[8] McCune, W.: Prover9 and mace4 (2005–2010), `http://www.cs.unm.edu/~mccune/prover9/`

[9] Miller, D.A.: A compact representation of proofs. Studia Logica 46(4), 347–370 (1987)

[10] Schulz, S.: System Description: E 1.8. In: McMillan, K., Middeldorp, A., Voronkov, A. (eds.) Proc. of the 19th LPAR, Stellenbosch. LNCS, vol. 8312. Springer (2013)