Simple Difficult Problems for Automated Theorem Provers

Chad E. Brown

Czech Technical University in Prague

Abstract. We describe a collection of simple first-order theorems about hereditarily finite sets. In spite of their short presentation and obvious theoremhood, they appear to difficult for current automated theorem provers.

Keywords:

1 The Problem Set

We begin by describe the first-order set theoretic language in which the problems are stated. We assume an infinite set of variables $x, y, z, \ldots$. Terms s, t are generated according to the following grammar:

$$x|\emptyset|\wp(t)|\{t\}|s \cup t|s \setminus t|\mathsf{adj}(s,t)|s^+$$

A term is closed if it has no variables. Formulas φ, ψ are generated as follows:

$$s = t|s \in t|s \subseteq t|\mathsf{disj}(s,t)|\mathsf{atleast}_2(s)|\cdots|\mathsf{atleast}_{16}(s)|\neg\varphi|\varphi \Rightarrow \psi|\varphi\wedge\psi|\varphi\vee\psi|\varphi \Leftrightarrow \psi|\forall x.\varphi|\exists x.\varphi$$

We write $s \neq t$, $s \notin t$ and $s \not\subseteq t$ for $\neg(s = t)$, $\neg(s \in t)$ and $\neg(s \subseteq t)$, respectively. We write 0 for $\emptyset$, 1 for 0^+, 2 for 1^+, 3 for 2^+ and 4 for 3^+. Sometimes we will write $s \cup \{t\}$ for $\mathsf{adj}(s,t)$ since these two terms are equal according to one of the axioms below.

We take 25 first-order formulas as axioms.

$\mathsf{A}_{\subseteq}$: $\forall xy.x \subseteq y \Leftrightarrow (forall z.z \in x \Rightarrow z \in y)$
$\mathsf{A}_{\mathsf{ext}}$: $\forall xy.x \subseteq y \Rightarrow y \subseteq x \Rightarrow x = y$
$\mathsf{A}_{\mathsf{disj}}$: $\forall xy.\mathsf{disj}(x,y) \Leftrightarrow \neg\exists z.z \in x \wedge z \in y$
$\mathsf{A}_{\emptyset}$: $\forall x.x \notin \emptyset$
$\mathsf{A}_{\{.\}}$: $\forall xy.x \in \{y\} \Leftrightarrow x = y$
$\mathsf{A}_{\cup}$: $\forall xy.x \in y \cup z \Leftrightarrow x \in y \vee x \in z$
$\mathsf{A}_{\setminus}$: $\forall xy.x \in y \setminus z \Leftrightarrow x \in y \wedge x \notin z$
$\mathsf{A}_{\mathsf{adj}}$: $\forall xy.\mathsf{adj}(x,y) = x \cup \{y\}$
$\mathsf{A}_{\mathsf{nxt}}$: $\forall x.x^+ = \mathsf{adj}(x,x)$
$\mathsf{A}_{\wp}$: $\forall xy.x \in \wp(y) \Leftrightarrow x \subseteq y$
$\mathsf{A}_{\geq 2}$: $\forall x.\mathsf{atleast}_2(x) \Leftrightarrow \exists y.y \in x \wedge x \not\subseteq \wp(y)$.
$\mathsf{A}_{\geq n+1}$: $\forall x.\mathsf{atleast}_{n+1}(x) \Leftrightarrow \exists y.y \subseteq x \wedge \mathsf{atleast}_n(y) \wedge x \not\subseteq y$ for $n \in \{2, \ldots, 15\}$.

Each of the problems consists of the 25 axioms above followed by one conjecture. The conjecture is always of one of five forms:

subq: $s \subseteq t$ where s and t are closed.
nsubq: $s \not\subseteq t$ where s and t are closed.
disj: $\mathsf{disj}(s,t)$ where s and t are closed.
$\mathsf{atleast}_n$: $\mathsf{atleast}_n(s)$ where s is closed and $n \in \{3,\ldots,16\}$
atmost_n: $\neg\mathsf{atleast}_{n+1}(s)$ where s is closed and $n \in \{2,\ldots,10\}$

In addition to knowing the s and t in the conjectures are closed, we also know they evaluate to sets that are semantically of small rank. In every model satisfying the axioms above, the interpretations of s and t will be members of the interpretation of $\wp(\wp(\wp(\wp(\wp(\emptyset)))))$. Using the Ackermann encoding of hereditarily finite sets as a model [1], each s and t will evaluate to a number in $\{0,\ldots,65535\}$.

Table 1 indicates the number of problems in each of the five categories.

subq	19437	21.4%
nsubq	12662	13.9%
disj	23831	26.2%
$\mathsf{atleast}_*$	21918	24.1%
atmost_*	13045	14.4%
Total	90893	100%

Table 1. Number of problems in each category

2 Preliminary ATP Results

We began by running two top first-order ATPs (automated theorem provers), Vampire [6] and E [7] in CASC mode for 60 seconds on each of the problems. We later also ran Prover9 for 60s on each problem with default settings. The results are in Table 2.

	Vampire	E	Prover9
subq	12371 (63.6%)	1067 (5.5%)	324 (1.7%)
nsubq	7925 (62.6%)	3566 (28.2%)	55 (0.4%)
disj	4402 (18.5%)	153 (0.6%)	0
$\mathsf{atleast}_*$	122 (0.6%)	15 (0.07%)	0
atmost_*	20 (0.2%)	0	0
Total	24840 (27.3%)	4801 (5.3%)	379 (0.4%)

Table 2. First results using Vampire, E and Prover9

In Table 3 we give a simple example from each category which neither Vampire nor E could prove in CASC mode within 60s.

$\mathsf{subq}^{24,30}$	$\{2\} \cup \{\{1\}\} \subseteq \{1\} \cup (\{2\} \cup \{\{1\}\}^+)$
$\mathsf{nsubq}^{50,48}$	$\wp(\wp(\{1\})) \setminus 1 \not\subseteq \{\{\{1\}\}\} \cup \wp(\{1\})$
$\mathsf{disj}^{9,96}$	$\mathsf{disj}(1 \cup 2, \{\wp(\{1\})\} \cup \{1\}^+)$
$\mathsf{atleast}_3^7$	$\mathsf{atleast}_3(1 \cup \{1\}^+)$
atmost_2^3	$\neg\mathsf{atleast}_3(2)$

Table 3. Simple examples not proven by Vampire or E in CASC mode in 60s

3 Including Enumeration Lemmas

Natural lemmas to include in the problems would be lemmas that list the members of the sets in question. An example of such a lemma would be

$$\forall x.x \in 2 \Leftrightarrow x = 0 \vee x = 1.$$

Based on the conjecture we can refine such lemmas further into the two directions of implication and only include the direction that seems the most helpful. Furthermore, a lemma like

$$\forall x.x = 0 \vee x = 1 \Rightarrow x \in 2$$

can be more naturally given as two lemmas: $0 \in 2$ and $1 \in 2$.

For all closed terms s used in the conjectures, we associate m closed terms closed terms $s_1, \ldots, s_m$ where semantically (and deductively) $s_1, \ldots, s_m$ list out all the m distinct members of s. We call

$$\forall x.x \in s \Rightarrow x = s_1 \vee \ldots \vee s = s_m$$

the *elimination lemma for* s. We call each of the formulas $s_i \in s$ an *introduction lemma for* s.

Based on this idea we created modified versions of the 90895 problems in which we included such lemmas as follows:

1. If the conjecture is $s \subseteq t$, we include the elimination lemma for s and the introduction lemmas for t.
2. If the conjecture is $s \not\subseteq t$, we include the introduction lemmas for s and the elimination lemma for t.
3. If the conjecture is $\mathsf{disj}(s, t)$ we include the elimination lemmas for s and t.
4. If the conjecture is $\mathsf{atleast}_n(s)$, we include the introduction lemmas for s.
5. If the conjecture is $\neg\mathsf{atleast}_n(s)$, we include the elimination lemma for s.

We ran E over these modified versions in CASC mode for 60s. The results (compared with the results without lemmas) is shown in Table 4. A significant improvement was seen only in the nsubq category, in which the results were almost as good as Vampire's results without lemmas.

	E with lemmas	E without lemmas
subq	967 (5%)	1067 (5.5%)
nsubq	7597 (6%)	3566 (28.2%)
disj	405 (1.7%)	153 (0.6%)
atleast$_*$	22 (0.1%)	15 (0.07%)
atmost$_*$	0	0
Total	8991 (9.9%)	4801 (5.3%)

Table 4. E with and without lemmas

4 Preliminary Satelo Results

A different approach than adding lemmas is to use a theorem prover that builds set theoretic reasoning into rules. A new theorem prover, Satelo, is under construction that follows this principle. Satelo is a theorem prover for higher-order set theory and is a fork of the higher-order theorem prover Satallax [4,3]. Preliminary results running Satelo on 3 different modes is reported in Table 5.

	MODE0	MODEB1	MODEB2	MODEB3	MODEB4
subq	9944 (51.2%)	9912 (51%)	13368 (68.7%)	655 (3.4%)	974 (5%)
nsubq	106 (0.8%)	94 (0.7%)	3695 (29.2%)	537 (4.2%)	1218 (9.6%)
disj	5687 (23.9%)	5048 (21.2%)	10679 (44.8%)	368 (1.5%)	375 (1.6%)
atleast$_*$	0	0	0	0	0
atmost$_*$	0	0	21 (0.2%)	0	0
Total	15737 (17.3%)	15054 (16.5%)	27763 (30.5%)	1560 (1.7%)	2567 (2.8%)

Table 5. Satelo Results

Satelo ignores the 25 axioms and only works on the conjecture. The constants $\in$, $\subseteq$, $\emptyset$, $\{\cdot\}$, $\cup$, $\setminus$ and $\wp$ are treated as built in, the same way as logical constants like $\vee$ and $=$. In addition to the usual tableau rules of Satallax [2], Satelo has set theory rules used instead of the axioms $\mathsf{A}_{\subseteq}$, $\mathsf{A}_{\mathsf{ext}}$, $\mathsf{A}_{\emptyset}$, $\mathsf{A}_{\{\cdot\}}$, $\mathsf{A}_{\cup}$, $\mathsf{A}_{\setminus}$ and $\mathsf{A}_{\wp}$. These set theory rules are shown in Figure 1. The remaining axioms are recognized as giving definitions for disj, adj, $^+$ and $\mathsf{atleast}_n$. The axioms are deleted and every occurrence of these names are expanded in favor of their definitions before search begins.

MODE0 uses the default flag settings for Satelo. The other modes differ from the default flag settings in the following ways:

- MODEB1 allows Satelo to produce simple first-order clauses (with existential variables) for which resolution with unification can be used to make inferences.
- MODEB2 delays working on sentences whose propositional literal has not been set to true in the latest propositional assignment found by MiniSat [5].

– MODEB3 delays working of sentences which evaluate to true in the Ackermann encoding.
– MODEB4 combines the two flag settings of MODEB2 and MODEB3.

Clearly MODEB2 performs best on this problem set, even outperforming Vampire.

$$\mathcal{T}_{\mathsf{ext}}\ \frac{s = t}{s \subseteq t \mid t \subseteq s} \qquad \mathcal{T}_{\subseteq}\ \frac{s \subseteq t}{\forall x.x \in s \Rightarrow x \in t} \qquad \mathcal{T}_{\not\subseteq}\ \frac{s \not\subseteq t}{w \in s, w \notin t}\ w \text{ FRESH}$$

$$\mathcal{T}_{\in\{\cdot\}}\ \frac{s \in \{t\}}{s = t} \qquad \mathcal{T}_{\notin\{\cdot\}}\ \frac{s \notin \{t\}}{s \neq t} \qquad \mathcal{T}_{\in\wp}\ \frac{s \in \wp t}{s \subseteq t} \qquad \mathcal{T}_{\notin\wp}\ \frac{s \notin \wp t}{s \not\subseteq t}$$

$$\mathcal{T}_{\in\cup}\ \frac{s \in t_1 \cup t_2}{s \in t_1 \mid s \in t_2} \qquad \mathcal{T}_{\notin\cup}\ \frac{s \notin t_1 \cup t_2}{s \notin t_1, s \notin t_2} \qquad \mathcal{T}_{\in\setminus}\ \frac{s \in t_1 \setminus t_2}{s \in t_1, s \notin t_2}$$

$$\mathcal{T}_{\notin\setminus}\ \frac{s \notin t_1 \setminus t_2}{s \notin t_1 \mid s \in t_2}$$

Fig. 1. Satelo Set Theory Rules

References

1. Ackermann, W.: Die Widerspruchsfreiheit der allgemeinen Mengenlehre. Mathematische Annalen 114(1), 305–315 (1937)
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. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT. LNCS, vol. 2919, pp. 502–518. Springer (2003)
6. 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)
7. 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)