Case Study: Saturations as Explicit Models in Equational Theories

Mikoláš Janota[1], Michael Rawson[2] and Stephan Schulz[3]

[1]Czech Technical University in Prague, CIIRC, Czechia
[2]University of Southampton, UK
[3]DHBW Stuttgart, Stuttgart, Germany

Abstract

Automated theorem provers (ATPs) can disprove conjectures by saturating a set of clauses, but the resulting saturated sets are opaque certificates. In the unit equational fragment, a saturated set can in fact be read as a convergent rewrite system defining an explicit, possibly infinite, model — but this is not widely known, even amongst frequent users of ATPs. Moreover, ATPs do not emit these explicit certificates for infinite (counter-)models. We present such a certificate construction in full, implement it in Vampire and E, and apply it to the recent Equational Theories Project [1], where hundreds of implications do not admit finite countermodels. The resulting rewrite systems can be checked for confluence and termination by existing certified tools, yielding trustworthy countermodels.

Keywords

model building, saturation theorem proving, rewriting

1. Introduction

In a September 2024 post on his blog [2], Terence Tao proposed *The Equational Theories Project* (ETP), a collaborative project bringing together mathematicians and automated reasoning researchers. The goal was to classify a set of 22,028,942 implications between universally-quantified equalities over a single binary operation (see Section 4 for details). Each such implication yields a formula that can be processed by a first-order automated theorem prover (ATP). An experimental study shows that ATPs are very successful on such problems [3]. From the initial 22 million, fewer than 1000 are left unsolved by Vampire [4].

The ETP is now completed [1], but at considerable human cost: the report counts 34 authors. A significant obstacle in using modern ATPs is that the mathematicians are also interested in *why* an implication holds or does not: a certificate. If the implication in question holds, ATPs are capable of producing a detailed proof. However, if the implication does not hold, a *counter-model* is required. Current ATPs cannot produce an explicit counter-model, but they can sometimes provide a *saturation* that implicitly describes a model. *Finite* counter-models can be obtained by the use of a *finite model builder* [5, 6]. However, in some cases all models are infinite. Moreover, large finite models are often out of reach — in fact, large finite models can be harder to find than the infinite ones represented by saturations. This is due to the combinatorics of known finite model building algorithms, which typically become too large for domain sizes above around 20. However, nontrivial infinite (counter-) models are often needed for classification of algebraic structures [7, 8]. This motivates our case study, where we modify current tools to produce such models.

Saturation-based ATPs such as E [9], Vampire [4], iProver [10], Zipperposition [11], Twee [12], or Waldmeister [13] search for a refutation by deducing consequences of the input formulae. If falsum is derived, the input was unsatisfiable. Another possibility is that *saturation* is achieved: no more new non-redundant consequences can be derived. If the proof procedure used is *complete* — that is, a proof

Practical Aspects of Automated Reasoning (PAAR) 2026, as part of the Federated Logic Conference (FLoC) 2026, Lisbon, Portugal, July 25, 2026

mikolas.janota@cvut.cz (M. Janota); michael@rawsons.uk (M. Rawson); schulz@eprover.org (S. Schulz)
0000-0003-3487-784X (M. Janota); 0000-0001-7834-1567 (M. Rawson); 0000-0001-6262-8555 (S. Schulz)

published 2026-08-20

of falsum is guaranteed to be found whenever the input is unsatisfiable — by contraposition the input is satisfiable. Typical ATP systems at best dump the saturated set of consequences to report satisfiability.

Saturated sets are unwieldy objects. Unlike proofs, they are opaque and rarely useful as counterexamples in downstream applications such as mathematics or verification. This problem has become more acute as ATP systems improve and saturate more often. Recent calculus improvements such as ground joinability [14] and AVATAR [15], combined with aggressive preprocessing such as blocked clause elimination [16] mean that systems saturate on more inputs than ever before, including some traditionally-difficult inputs like associative and commutative operators.

We would prefer to extract a more explicit representation of a model from saturations, one which allows at least evaluating ground terms. Doing this for full clausal logic is very difficult [17], but for certain fragments it is considerably easier. We consider the *unit equational* fragment, which allows only universally-quantified equations and ground disequations. This fragment is nonetheless very powerful: many algebraic structures can be expressed as sets of unit equations[1]; it suffices for many applications in functional programming; and even full first-order logic can be embedded in it, the Horn fragment very efficiently [18].

For the unit equational fragment, there is a straightforward reading of saturated sets as potentially-infinite models (Section 3), yet this reading is underexploited in practice — in part because it relies on subtleties of rewriting theory that are not obvious to end-users. We aim to bridge this gap, making saturations directly usable as counter-models for non-specialists, in particular mathematicians. For the avoidance of doubt, we consider the following to be the contributions of the present paper:

1. We demonstrate that infinite countermodels can be extracted from saturation-based provers. This is a known consequence of theory, but only to a small group of experts; many *users* of automated reasoning tools are unaware of it.
2. We provide tooling (implemented in Vampire and E) to emit the produced interpretation as an explicit rewrite system, and to verify that it is indeed a model using existing certified checkers.
3. We carry out a case study on the Equational Theories Project, constructing hundreds of counter-models for formulae where finite model builders fail.

2. Preliminaries

We assume familiarity with classical first-order logic [19] with equality. ATP systems take a set of *axiom* formulas Γ and a single conjecture formula C, and try to prove or disprove C from Γ, completely automatically and without human intervention. In the unit equational case, C and every formula in Γ are of the form $\forall x_1 \ldots x_n.\ s \approx t$. ATPs typically proceed by refutation, negating C and trying to reach a contradiction. As C is negated, the negation is pushed inwards and C becomes a disequation $s \not\approx t$, containing no variables after Skolemisation.

The initial set of clauses E is therefore $\Gamma, s \not\approx t$. Saturation-based ATPs then exhaustively apply inferences of a complete calculus to E, adding any consequences back to E. If a contradiction is produced, C is a theorem of Γ. However, if no contradiction is detected and no non-redundant new clauses can be derived that are not already in E, then E is *saturated* and $\Gamma, s \not\approx t$ is *satisfiable*. Otherwise, a proof would have been found by the complete calculus. Therefore there is a model (or counterexample) satisfying $\Gamma, s \not\approx t$, which shows that C is not a theorem of Γ. In Section 3 we show how to construct this implicitly-represented model for saturation-based ATPs that use calculi descended from *completion* [20].

Completion attempts to transform universally-quantified equations into rewrite rules, such that the resulting term rewriting system is confluent. This is done using a *reduction order* $\prec$ over terms. When an equation $s = t$ has $t \prec s$, it is promoted to a rewrite rule $s \to t$. Deduction is still necessary in order to deal with (i) equations which are not thus oriented; and (ii) *critical pairs* t_1, t_2 such that some t can be rewritten to either t_1 or t_2. Completion-based calculi like superposition therefore incrementally construct a confluent term rewriting system.

[1] "Admits a unit equational axiomatisation" is nearly the definition of a *variety*.

The *signature* of a unit equational problem is the set of functions and constants in $\Gamma, s \not\approx t$. A term or equation is *ground* if it contains no variables. We write $s[t]$ to indicate that t is a sub-term of s, not excluding the possibility that s *is* t. The *Herbrand universe* is the set of all ground terms that can be constructed from symbols in the signature. Substitutions σ are mappings from variables to terms. Their application to a term, $\sigma(t)$, replaces every variable x in t with $\sigma(x)$. Equations are considered implicitly symmetric.

3. Constructing Explicit Models

Suppose we are given a saturated set E of universally-quantified equations[2] from an ATP based on unfailing completion [21, 22] or superposition [23]. We are also given the signature, and details of its internal term ordering $\prec$, which is total, well-founded, and decidable on ground terms. We will now explicitly construct a Herbrand model for E under $\prec$ such that ground terms and equations can be evaluated.

The domain D is the Herbrand universe of the signature, i.e. ground terms interpreted only as themselves. We therefore need only define the equality relation $\approx$. The idea is that terms will be equal when they have the same normal form when rewritten by E. To this end, we define a computable normalisation function $[\![\_]\!] : D \to D$.

Definition 1 (rewrite)**.** *A ground term $t[\sigma(l)]$ can be rewritten to $t[\sigma(r)]$ when there is a ground instance $\sigma(l) \approx \sigma(r)$ of an equation $l \approx r \in E$ with $\sigma(l) \succ \sigma(r)$.*

Note that $l \succ r$ without substitution does not necessarily hold when rewriting. Finding a rewrite can be done algorithmically by matching either side of an equation onto any subterm of t, then checking the ordering constraint. If there are unbound variables on the other side of the equation, they can be mapped to any ground term such that $\sigma(l) \succ \sigma(r)$, in practice the smallest constant.

Definition 2 (normalisation)**.** *The normal form $[\![t]\!]$ of a term t under E is the result of rewriting t until an irreducible term is reached.*

Any such computation terminates, since $\prec$ is well-founded. Additionally, the precise order of rewrites found by a particular normalisation function does not matter, since the set of equations resulting if the unfailing completion procedure terminates is confluent on ground terms [22].

Example 1. *Consider a binary function f, and the lexicographic path ordering with $f \succ b \succ a$. f is commutative, and this results in the saturated set $f(x, y) \approx f(y, x)$. One possible computation normalising $f(f(b, a), a)$ is*

$$[\![f(f(b,a),a)]\!] \to [\![f(f(a,b),a)]\!] \to [\![f(a,f(a,b))]\!] \to f(a,f(a,b))$$

We can now define $s \approx t$ to be the equivalence relation $\{(s, t) \mid [\![s]\!] = [\![t]\!]\}$ and we have our model. It remains to show that this is a model of the input. The input disequation $s \not\approx t$ has $[\![s]\!] \neq [\![t]\!]$, otherwise the ATP would have derived falsum. It is therefore satisfied. Input equations $l \approx r$ are true by assumption in the original equational theory, which is invariant under (unfailing) completion. Hence there is a proof of $l \approx r$ in the saturated system, and by the equivalence of Church-Rosser and confluence, a rewrite proof for any ground instance of it. So, $[\![\sigma(l)]\!] = [\![\sigma(r)]\!]$ for any grounding substitution σ.

Ground terms can be easily evaluated as their normal forms in such models, and ground equations are also decided by normalisation. We remark that each saturated clause set and $\prec$ induces a unique model. A limitation of this approach is that there is no obvious way to determine whether two models are equivalent, nor to evaluate non-ground terms or non-ground (dis)equations.

[2] Disequations in the saturation can be ignored for our purposes.

$$
\begin{aligned}
f_2 * f_4 &\to f_0(b, f_4) & f_4 * b &\to f_0(f_3, b)\\
f_0(f_0(f_0(x,x),x),x) &\to x & b * a &\to f_2\\
f_2 * a &\to f_0(b, a) & f_3 * b &\to f_4\\
f_0(f_2, b) &\to f_4 & f_1(y, x) &\to x\\
y * f_0(x, y) &\to x & f_2 * b &\to f_3\\
f_0(x, f_0(x, x)) &\to x & f_0(x, y) * y &\to f_0(x * y, y)\\
x * f_0(x, y) &\to f_0(y, f_0(x, y)) & b * f_4 &\to f_2\\
(x * y) * y &\to f_0(x, y) & x * x &\to f_0(f_0(x, x), x)
\end{aligned}
$$

Figure 1: Confluent and terminating system for $118 \wedge \neg 274$

4. Case Study: Countermodels of Equational Theories

The ETP targets implications between universally-quantified equalities over a single binary operation $*$, which we motivate by the following examples.

$$(x * y) * z \approx x * (y * z) \implies x * y \approx y * x \tag{1}$$

$$x * y \approx y * x \implies (x * y) * z \approx x * (y * z) \tag{2}$$

$$x * y \approx u * w \implies x * y \approx y * x \tag{3}$$

Implication (1) asks whether associativity implies commutativity; it does not hold, e.g., matrix multiplication is associative but not commutative. Likewise, (2) asks whether commutativity implies associativity; this fails because, e.g., $\frac{x+y}{2}$ is commutative but not associative. In contrast, implication (3) holds: the left-hand side forces $*$ to be a constant function, which is necessarily commutative. The ETP considered all equalities where the operation $*$ is applied at most 4 times. There are $n = 4{,}694$ equations, systematically numbered from 1 to 4694, yielding $n^2 - n = 22{,}028{,}942$ possible implications. The ETP has successfully decided all the considered implications, and the results can be explored on the project's website [24].

The previous experiment [3] by Janota shows that Vampire can prove *all* implications that hold. These already have a certificate in the form of a superposition proof. On the other hand, 13,854,015 problems were shown *not* to hold by Vampire's finite model builder (FMB) [5, 6], which produces a checkable certificate. A further 817 non-theorems were detected by saturation. We ran the finite model builder on these 817 with a longer timeout, leaving us with 304 non-theorems refuted only by saturation (rather than finite model building). Of these, 196 are marked *finitely unsatisfiable* by Infinox [25], so no finite counter-model exists. This means that the remaining 108 *may* have a finite counter-model, but we were not able to obtain it via FMB. A detailed table of the experiments is available on the authors' website[3]. Further reductions are possible by identifying *duals*: implications that can be obtained from one another by swapping order of arguments, which we avoid here for clarity. The numbers are summarized in the following table.

	¬Fin. Unsat	Fin. Unsat	Total
¬FMB	108	196	**304**
FMB	513	0	513

As an example, consider disproving the implication between equations 118 and 274:

$$\forall xy.\, x \approx y * ((x * y) * y) \implies \forall xy.\, x \approx ((y * x) * y) * y$$

[3] https://people.ciirc.cvut.cz/~janotmik/stamp

After negation and Skolemization, we obtain the input clause set

$$x \approx y * ((x * y) * y) \qquad\qquad a \not\approx ((b * a) * b) * b$$

which is satisfiable but admits only infinite models.[4] The rewrite system for the counter-model is shown in Fig. 1. Note that Vampire has introduced definitions

$$f_0(x, y) \triangleq (x * y) * y \qquad\qquad f_1(x, y) \triangleq x * f_0(y, x)$$

$$f_2 \triangleq b * a \qquad\qquad f_3 \triangleq f_2 * b \qquad\qquad f_4 \triangleq f_3 * b$$

in order to achieve saturation. These definitions cannot be easily read off the saturated set, but they are not necessary to define the model. Perhaps surprisingly, in some cases the original problem is already saturated. This is the case for the implication between `477` and `1426`, which also has only infinite counter-models.[5]

$$x \approx y * (x * (y * (y * y))) \qquad\qquad a \not\approx (a * a) * (a * (a * a))$$

Here the positive equation is pre-oriented right-to-left, and no consequences are produced from either clause in the calculus.

5. Checked Models

Ideally, models would be externally checkable. Systems like E or Vampire are large, complex pieces of software, and even their underlying theory can be tricky [26]. One path to more trust we envisage is exporting a model suitable for checking in a proof assistant, such as the Lean formalisation employed in the ETP. This would require:

1. Defining the domain as a Herbrand universe. This is a straightforward inductive definition in most systems.
2. Defining a normalisation function. Many systems require that all user-defined functions terminate, which typically requires showing that $\prec$ is well-founded. This may not be trivial, and depends on fine details of the term ordering.
3. Showing that the resulting model is a model of the input. This should be trivial for ground disequations, but universally-quantified equations will require a more involved argument.
4. Therefore, we will also very likely need to show confluence of the rewrite system. This is guaranteed in theory but will need to be re-proven, perhaps using a confluence checker.

[4]

No finite counterexample to `118` $\implies$ `274`

The following clause set has no finite model.

$$x \approx y * ((x * y) * y) \tag{4}$$

$$a \not\approx (((b * a) * b) * b) \tag{5}$$

Proof. Suppose that there is a finite model for contradiction. First, observe that left-multiplication is surjective: by (4) every x is of the form $y * z$, where z is $(x * y) * y$. Surjective functions on finite sets are injective. Now consider the following instance of (4):

$$b * a \approx b * (((b * a) * b) * b),$$

and obtain

$$a \approx ((b * a) * b) * b,$$

by injectivity. □

[5] By an analoguous argument.

Such a system would be quite involved. However, to demonstrate the concept, we have modified Vampire[6] and E[7] to print the rewrite system on saturation, provided that every equation $l \approx r$ is orientable, i.e. $l \succ r$. Of the 304 saturations of interest, to our surprise 261 are orientable. We used the tool CSI [27, 28] to certify (ground) confluence of the rewrite system, and TTT2 [29] for termination. Both tools produce certificates independently verifiable by the tool CeTA [30], which itself relies on the IsaFoR Isabelle library. All 261 orientable rewrite systems were certified as confluent and terminating by CSI and TTT2. In the other 43 cases, a model can still be constructed, but it is not yet clear to us how to automatically certify (ground-)confluence and termination.

6. Related Work

Bachmair-Ganzinger model construction [23] builds a Herbrand model from a saturated set by generating rewrite rules from so-called *productive* ground instances of clauses. More recently, Lynch [31] revisits model construction, giving a more explicit and algorithmic account. The construction in Section 3 is a specialisation of this framework to the unit equational fragment, where the saturated set of equations directly forms a convergent ground rewrite system and the model is the quotient of the Herbrand universe by normalisation. Unit equations allow us to dispense with clause-level machinery and use results from unfailing completion [22] directly.

Peltier [32] develops methods for extracting Herbrand models from clause sets saturated under ordered resolution, and extends them to refinements such as hyperresolution [33], as do Fermüller and Leitsch [34]. Peltier also builds infinite models from equational saturated sets, defining *non-ambiguous* formulae such that model construction is well-defined [35]. In our unit equational setting, non-ambiguity is guaranteed by the convergence of the rewrite system obtained from saturation. Horbach and Weidenbach study decidability results for saturation-based model building in full first-order logic [17]. Infinite models can also be constructed by looking for a model of a fixed template [36, 37], by identifying specific forms of the formula [38], or by enumeration [39].

The TSTP format for ATP solutions has recently been extended [40, 41] to allow for reporting both finite and infinite models. In principle, the models we construct could be reported in this format. Unfailing completion has been extended to record how rules in a terminating and confluent rewrite system were derived [42], producing certificates that can be formally checked [43].

7. Summary and Future Work

We have shown that saturated sets produced by ATPs in the unit equational fragment can be read as (ground-)convergent rewrite systems defining explicit, potentially infinite models. The construction relies on known results in rewriting but has not been widely accessible to users of ATPs; we present it here in detail. Applying the construction to the Equational Theories Project, we modified the saturation-based ATPs Vampire and E to emit models in the form of ground-convergent rewrite systems, even for problems where no finite counter-model exists. We demonstrated that all resulting saturations with orientable equations can be verified for confluence and termination by certified tools, yielding 261 fully verified counter-models. This independent verification also corroborates that the ATPs emit correct saturations, even for the 43 problems with unoriented equations, which are currently beyond our verification approach.

Extending the approach beyond the unit equational fragment to richer clausal logics is a challenging but important goal. At least for the Horn case, this should be possible either by embedding Horn problems into UEQ [18], or by treating the axioms as conditional rewrite rules in the style of Dershowitz [44]. A further question is whether (potentially large) *finite* models could be constructed explicitly from a convergent rewrite system.

[6] https://github.com/vprover/vampire/tree/michael-preordered-saturations
[7] https://github.com/eprover/eprover

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] M. Bolan, J. Breitner, J. Brox, N. Carlini, M. Carneiro, F. van Doorn, M. Dvorak, A. Goens, A. Hill, H. Husum, H. I. Mejia, Z. A. Kocsis, B. L. Floch, A. L. Bar-on, L. Luccioli, D. McNeil, A. Meiburg, P. Monticone, P. P. Nielsen, E. O. Osazuwa, G. Paolini, M. Petracci, B. Reinke, D. Renshaw, M. Rossel, C. Roux, J. Scanvic, S. Srinivas, A. R. Tadipatri, T. Tao, V. Tsyrklevich, F. Vaquerizo-Villar, D. Weber, F. Zheng, The equational theories project: Advancing collaborative mathematical research at scale, 2025. URL: https://arxiv.org/abs/2512.07087. arXiv:2512.07087.

[2] T. Tao, A pilot project in universal algebra to explore new ways to collaborate and use machine assistance?, 2024. URL: https://terrytao.wordpress.com/2024/09/25/a-pilot-project-in-universal-algebra-to-explore-new-ways-to-collaborate-and-use-machine-assistance/.

[3] M. Janota, Experimental results for Vampire on the equational theories project, 2025. URL: https://arxiv.org/abs/2508.15856. arXiv:2508.15856.

[4] F. Bártek, A. Bhayat, R. Coutelier, M. Hajdú, M. Hetzenberger, P. Hozzová, L. Kovács, J. Rath, M. Rawson, G. Reger, M. Suda, J. Schoisswohl, A. Voronkov, The Vampire diary, in: R. Piskac, Z. Rakamaric (Eds.), Computer Aided Verification - 37th International Conference, CAV 2025, Zagreb, Croatia, July 23-25, 2025, Proceedings, Part III, volume 15933 of *Lecture Notes in Computer Science*, Springer, 2025, pp. 57–71. doi:10.1007/978-3-031-98682-6_4.

[5] K. Claessen, N. Sörensson, New techniques that improve MACE-style model finding, in: Proc. of Workshop on Model Computation (MODEL), 2003.

[6] G. Reger, M. Riener, M. Suda, Symmetry avoidance in MACE-style finite model finding, in: A. Herzig, A. Popescu (Eds.), Frontiers of Combining Systems - 12th International Symposium, FroCoS 2019, London, UK, September 4-6, 2019, Proceedings, volume 11715 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 3–21. doi:10.1007/978-3-030-29007-8_1.

[7] J. D. Phillips, P. Vojtěchovský, The varieties of loops of Bol-Moufang type, Algebra universalis 54 (2005) 259–271. doi:10.1007/s00012-005-1941-1.

[8] M. Kinyon, R. Veroff, P. Vojtěchovský, Loops with Abelian Inner Mapping Groups: An Application of Automated Deduction, Springer Berlin Heidelberg, 2013, p. 151–164. doi:10.1007/978-3-642-36675-8_8.

[9] S. Schulz, S. Cruanes, P. Vukmirović, Faster, higher, stronger: E 2.3, in: P. Fontaine (Ed.), Proc. of the 27th CADE, Natal, Brasil, number 11716 in LNAI, Springer, 2019, pp. 495–507. doi:10.1007/978-3-030-29436-6_29.

[10] A. Duarte, K. Korovin, Implementing superposition in iProver (system description), in: N. Peltier, V. Sofronie-Stokkermans (Eds.), Proc. of the 10th IJCAR, Paris (Part II), volume 12167 of *LNAI*, Springer, 2020, pp. 158–166. doi:10.1007/978-3-030-51054-1_24.

[11] P. Vukmirović, A. Bentkamp, J. Blanchette, S. Cruanes, V. Nummelin, S. Tourret, Making higher-order superposition work, Journal of Automated Reasoning 66 (2022) 541–564. doi:10.1007/s10817-021-09613-z.

[12] N. Smallbone, Twee: An equational theorem prover, in: Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2021, p. 602–613. doi:10.1007/978-3-030-79876-5_35.

[13] T. Hillenbrand, B. Löchner, The next Waldmeister loop, in: A. Voronkov (Ed.), Automated Deduction - CADE-18, 18th International Conference on Automated Deduction, volume 2392 of *Lecture Notes in Computer Science*, 2002, pp. 486–500. doi:10.1007/978-3-540-45085-6_27.

[14] A. Duarte, K. Korovin, Ground joinability and connectedness in the superposition calculus, in: J. Blanchette, L. Kovács, D. Pattinson (Eds.), Automated Reasoning - 11th International Joint

Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings, volume 13385 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 169–187. doi:10.1007/978-3-031-10769-6_11.

[15] A. Voronkov, AVATAR: the architecture for first-order theorem provers, in: A. Biere, R. Bloem (Eds.), Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings, volume 8559 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 696–710. doi:10.1007/978-3-319-08867-9_46.

[16] B. Kiesl, M. Suda, M. Seidl, H. Tompits, A. Biere, Blocked clauses in first-order logic, in: T. Eiter, D. Sands (Eds.), LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana, May 7-12, 2017, volume 46 of *EPiC Series in Computing*, EasyChair, 2017, pp. 31–48. doi:10.29007/C3WQ.

[17] M. Horbach, C. Weidenbach, Decidability results for saturation-based model building, in: R. A. Schmidt (Ed.), Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction, Montreal, Canada, August 2-7, 2009. Proceedings, volume 5663 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 404–420. doi:10.1007/978-3-642-02959-2_30.

[18] K. Claessen, N. Smallbone, Efficient encodings of first-order Horn formulas in equational logic, in: D. Galmiche, S. Schulz, R. Sebastiani (Eds.), Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, volume 10900 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 388–404. doi:10.1007/978-3-319-94205-6_26.

[19] R. M. Smullyan, First-Order Logic, Springer Berlin Heidelberg, 1968. doi:10.1007/978-3-642-86718-7.

[20] D. Knuth, P. Bendix, Simple word problems in universal algebras, in: J. Leech (Ed.), Computational Algebra, Pergamon Press, 1970, pp. 263–297.

[21] J. Hsiang, M. Rusinowitch, On Word Problems in Equational Theories, in: Proc. of the 14th ICALP, Karlsruhe, volume 267 of *LNCS*, 1987, pp. 54–71.

[22] L. Bachmair, N. Dershowitz, D. Plaisted, Completion without failure, in: H. Ait-Kaci, M. Nivat (Eds.), Resolution of Equations in Algebraic Structures, volume 2, Academic Press, 1989, pp. 1–30. doi:10.1016/B978-0-12-046371-8.50007-9.

[23] L. Bachmair, H. Ganzinger, Rewrite-based equational theorem proving with selection and simplification, Journal of Logic and Computation 4 (1994) 217–247. doi:10.1093/logcom/4.3.217, report: https://pure.mpg.de/rest/items/item_1834970/component/file_1857487/content.

[24] T. Tao, Equational theories project, ???? URL: https://teorth.github.io/equational_theories/dashboard/.

[25] K. Claessen, A. Lillieström, Automated inference of finite unsatisfiability, Journal of Automated Reasoning 47 (2011) 111–132. doi:10.1007/s10817-010-9216-8.

[26] A. Schlichtkrull, J. C. Blanchette, D. Traytel, U. Waldmann, Formalizing Bachmair and Ganzinger's ordered resolution prover, in: D. Galmiche, S. Schulz, R. Sebastiani (Eds.), Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, volume 10900 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 89–107. doi:10.1007/978-3-319-94205-6_7.

[27] J. Nagele, B. Felgenhauer, A. Middeldorp, CSI: New Evidence – A Progress Report, Springer International Publishing, 2017, p. 385–397. doi:10.1007/978-3-319-63046-5_24.

[28] H. Zankl, B. Felgenhauer, A. Middeldorp, CSI – A Confluence Tool, Springer Berlin Heidelberg, 2011, p. 499–505. doi:10.1007/978-3-642-22438-6_38.

[29] M. Korp, C. Sternagel, H. Zankl, A. Middeldorp, Tyrolean Termination Tool 2, Springer Berlin Heidelberg, 2009, p. 295–304. doi:10.1007/978-3-642-02348-4_21, preprint http://cl-informatik.uibk.ac.at/users/mkorp/research/papers/KSZM09.pdf.

[30] R. Thiemann, C. Sternagel, Certification of Termination Proofs Using CeTA, Springer Berlin Heidelberg, 2009, p. 452–468. doi:10.1007/978-3-642-03359-9_31.

[31] C. Lynch, Constructing Bachmair-Ganzinger models, in: A. Voronkov, C. Weidenbach (Eds.), Programming Logics - Essays in Memory of Harald Ganzinger, volume 7797 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 285–301. doi:10.1007/978-3-642-37651-1_12.

[32] N. Peltier, Model building with ordered resolution: extracting models from saturated clause sets, J. Symb. Comput. 36 (2003) 5–24. doi:10.1016/S0747-7171(03)00028-2.
[33] N. Peltier, Extracting models from clause sets saturated under semantic refinements of the resolution rule, Inf. Comput. 181 (2003) 99–130. doi:10.1016/S0890-5401(03)00016-6.
[34] C. G. Fermüller, A. Leitsch, Hyperresolution and automated model building, J. Log. Comput. 6 (1996) 173–203. doi:10.1093/LOGCOM/6.2.173.
[35] N. Peltier, Building infinite models for equational clause sets: Constructing non-ambiguous formulae, Logic Journal of the IGPL 11 (2003) 97–129. doi:10.1093/jigpal/11.1.97.
[36] C. E. Brown, M. Janota, C. Kaliszyk, Abstract: Challenges and solutions for higher-order SMT proofs, in: D. Déharbe, A. E. J. Hyvärinen (Eds.), Proceedings of the 20th Internal Workshop on Satisfiability Modulo Theories co-located with the 11th International Joint Conference on Automated Reasoning (IJCAR 2022) part of the 8th Federated Logic Conference (FLoC 2022), Haifa, Israel, August 11-12, 2022, volume 3185 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, p. 128. URL: https://ceur-ws.org/Vol-3185/abstract697.pdf.
[37] N. Elad, O. Padon, S. Shoham, An infinite needle in a finite haystack: Finding infinite counter-models in deductive verification, Proceedings of the ACM on Programming Languages 8 (2024) 970–1000. doi:https://doi.org/10.1145/3632875.
[38] Y. Ge, L. M. de Moura, Complete instantiation for quantified formulas in satisfiability modulo theories, in: Computer Aided Verification, 21st International Conference, CAV, 2009, pp. 306–320. doi:10.1007/978-3-642-02658-4_25.
[39] J. Parsert, C. E. Brown, M. Janota, C. Kaliszyk, Experiments on infinite model finding in SMT solving, in: R. Piskac, A. Voronkov (Eds.), LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Manizales, Colombia, 4-9th June 2023, volume 94 of *EPiC Series in Computing*, EasyChair, 2023, pp. 317–328. doi:10.29007/SLRM.
[40] G. Sutcliffe, A. Steen, P. Fontaine, The new TPTP format for interpretations, CoRR abs/2406.06108 (2024). doi:10.48550/ARXIV.2406.06108. arXiv:2406.06108.
[41] A. Steen, G. Sutcliffe, P. Fontaine, J. McKeown, Representation, verification, and visualization of tarskian interpretations for typed first-order logic, in: R. Piskac, A. Voronkov (Eds.), LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, volume 94 of *EPiC Series in Computing*, EasyChair, 2023, pp. 369–385. doi:10.29007/1RHX.
[42] T. Sternagel, S. Winkler, H. Zankl, Recording completion for certificates in equational reasoning, in: X. Leroy, A. Tiu (Eds.), Proceedings of the 2015 Conference on Certified Programs and Proofs, CPP 2015, Mumbai, India, January 15-17, 2015, ACM, 2015, pp. 41–47. URL: https://doi.org/10.1145/2676724.2693171. doi:10.1145/2676724.2693171.
[43] C. Sternagel, S. Winkler, Certified equational reasoning via ordered completion, in: P. Fontaine (Ed.), Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings, Lecture Notes in Computer Science, Springer, 2019, pp. 508–525. URL: https://doi.org/10.1007/978-3-030-29436-6_30. doi:10.1007/978-3-030-29436-6_30.
[44] N. Dershowitz, Ordering-based strategies for Horn clauses, in: J. Mylopoulos (Ed.), Proc. of the 12th IJCAI, Sydney, volume 1, Morgan Kaufmann, 1991, pp. 118–124. URL: https://www.ijcai.org/Proceedings/91-1/Papers/020.pdf.