max planck institut
informatik

Automatic Reasoning (AR)

Beyond SAT and SMT

Christoph Weidenbach

Automatic Reasoning

The science of developing systems that automatically test (un)satisfiability, validity of a logical formula.

SAT:

$$P \lor Q \lor R$$
$$\neg P \lor Q \lor R$$
$$P \lor \neg Q \lor R$$
$$\neg P \lor \neg Q \lor R$$
$$P \lor Q \lor \neg R$$
$$\neg P \lor Q \lor \neg R$$
$$P \lor \neg Q \lor \neg R$$
$$\neg P \lor \neg Q \lor \neg R$$

FOL:

$$R(\epsilon, \epsilon)$$
$$\neg R(x, y) \lor R(h(x), g(x))$$
$$\neg R(x, y) \lor R(g(x), h(g(y)))$$
$$\neg R(x, y) \lor R(g(x), y)$$
$$\neg R(g(x), g(x))$$

$$R(g(h(g(\epsilon))), g(h(g(\epsilon))))$$

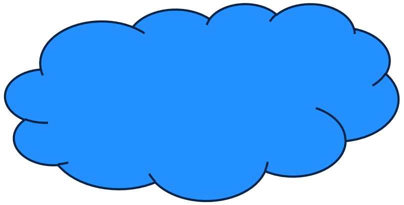
Post Correspondence Problem (PCP) [Post46]

Message

The more expressive the logic the more the need for a sophisticated combination of AR techniques in order to obtain a robust user experience.

Robust:

- Small changes to a problem formulation result in small changes in system solving.
- Easy problems are solved fast.



This is a dream, in general, but achievable in specific settings.

Parts of the AR Landscape

max planck institut informatik

$P \vee \neg Q$		$\exists P. \forall Q.\ P \vee \neg Q$		$P(x) \vee \neg Q(a,y)$		$\forall x. \exists y.\ P(x) \vee \neg Q(a,y)$
SAT	$\subset$	QBF	$\subset$	BS	$\subset$	FOL
NP	$\Longleftarrow$ EXP	**PSPACE**	$\Longleftarrow$ EXP	**NEXPTIME**		**UNDECIDABLE**
Hardware Verification		Hardware Verification		Knowledge Representation		Theorem Proving

+		+
LIA	$$\begin{array}{rcl} 3x - 4y + 1 & \leq & 0 \\ x + y - 5 & \geq & 0 \end{array}$$	LIA
NP		**NP**
=		=
SMT		BS(T)
PSPACE	$\subset$	**UNDECIDABLE**
Software Verification		Universal

[Coo71, Lew79, Lew80, Pap81, Pla84, FLHT01, BHvMW09]

Why does SAT work?

CDCL (Conflict Driven Clause Learning) [SS96, BS97]

$$N = \{\neg S \lor P, \neg S \lor \neg P \lor \neg Q, \ldots\}$$

$$\begin{aligned}(\epsilon, N, \top) &\Rightarrow (Q^1 R^2 S^3 P^{\neg S \lor P}, N, \top)\\ &\Rightarrow (Q^1 R^2 S^3 P^{\neg S \lor P}, N, \neg S \lor \neg P \lor \neg Q)\\ &\Rightarrow (Q^1 R^2 S^3, N, \neg S \lor \neg Q)\\ &\Rightarrow (Q^1 \neg S^{\neg S \lor \neg Q}, N \cup \{\neg S \lor \neg Q\}, \top)\end{aligned}$$

$\neg S \lor \neg Q$ is not redundant:

$$N^{\prec \neg S \lor \neg Q} \not\models \neg S \lor \neg Q$$

No waste of computing time.

Non-Redundant Clauses

Theorem [Wei15]

If $(L_1 \ldots L_n, N, C)$ is a CDCL Backtracking state with eager Conflict and Propagate, then $N^{\prec C} \not\models C$ where $L_1 \prec \ldots \prec L_n$. Non-Redundancy is NP-complete.

$N = \{C_1, \ldots, C_n\}$ is satisfiable

$N' = \{C_1 \vee \neg Q, \ldots, C_n \vee \neg P,\ P,\ Q\}$ is satisfiable, P, Q new

$(N' \setminus \{P, Q\})^{\prec \neg P \vee \neg Q} \not\models \neg P \vee \neg Q$ where $\neg P \prec \neg Q$ maximal in $\prec$

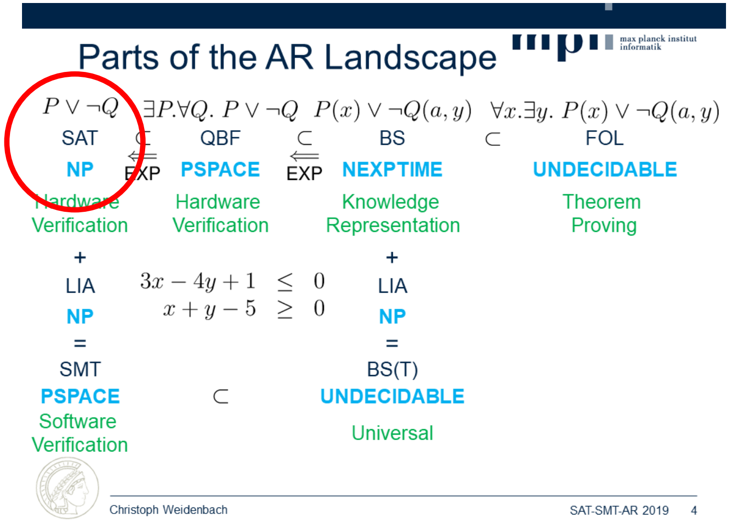
CDCL either finds a model or generates a non-redundant clause with respect to an NP-complete criterion.

No waste of computing time.

Summary

SAT works because:

- Explicit, efficient model generation
- Non-redundant clause learning
- No waste of computing time



Why does SMT work?

max planck institut informatik

SMT (Satisfiability Modulo Theories) [NOT06]

$$N = \{x - y \leq 0,\ x \geq 5,\ y \leq 7,\ 1 - x + y \leq 0 \vee y \geq 6\}$$

$$N' = \{\quad P \quad,\quad Q \quad,\quad R \quad,\quad S \vee T \quad\}$$

$$(\epsilon, N', \top) \Rightarrow^*_{\text{CDCL}} (P^P Q^Q R^R S^1, N', \top)$$

LIA $\quad x - y \leq 0,\ x \geq 5,\ y \leq 7,\ 1 - x + y \leq 0$

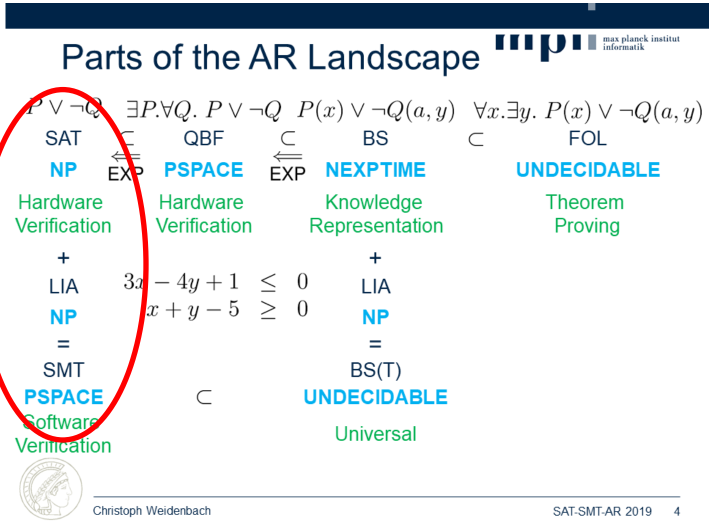
$$\Rightarrow^*_{\text{LIA}}\ x \geq 5 \quad y \geq 6 \quad x \geq 6 \quad \ldots \quad y \geq 8$$

$$\Rightarrow_{\text{LIA}}\ \neg P \vee \neg Q \vee \neg R \vee \neg S$$

$$\Rightarrow^*_{\text{CDCL}} (P^P Q^Q R^R \neg S^{\neg P \vee \neg Q \vee \neg R \vee \neg S} T^{S \vee T}, N'', \top)$$

LIA $\quad x - y \leq 0,\ x \geq 5,\ y \leq 7,\ y \geq 6$

Summary



SAT works because:

- Explicit, efficient model generation
- Non-redundant clause learning
- No waste of computing time

SMT works because:

- Abstraction
- SAT works
- Explicit, efficient model generation CDCL(LIA)
- No waste of computing time
- No notion of non-redundant clause learning CDCL(LIA)

Bernays-Schönfinkel (BS)

$P \vee \neg Q$ $\quad$ $P(x) \vee \neg Q(a,y)$

SAT $\subset$ BS

NP $\overset{\Longleftarrow}{\text{EXP}}$ **NEXPTIME**

$$
\begin{array}{l}
P(a) \vee \neg Q(a,a)\\
P(a) \vee \neg Q(a,b)\\
P(b) \vee \neg Q(a,a)\\
P(b) \vee \neg Q(a,b)\\
Q(a,a) \vee Q(b,a)\\
Q(a,b) \vee Q(b,a)\\
Q(b,a) \vee Q(b,b)\\
Q(b,b) \vee Q(b,b)
\end{array}
\quad \overset{\Longleftarrow}{\text{EXP}} \quad
\begin{array}{l}
P(x) \vee \neg Q(a,y)\\
Q(x,y) \vee Q(b,x)
\end{array}
$$

mr^k ground atoms

Reduction to SAT

Answer Set Programming (ASP) [KLPS16]

[BS28,vH67]

BS Explicit Models

$P \vee \neg Q$		$P(x) \vee \neg Q(a, y)$
SAT		BS
NP	$\neq$	**NEXPTIME**

There cannot be an efficient model representation formalism for BS, in general.

There are several:

- ME [BFT06] $P(x, a, y)/\{P(b, a, y), P(c, a, c)\}$ **P**
- DPLL(SX) [PMB10] $P(x, z, y)\{\{z \mapsto a, x \mapsto a\}, \ldots\}$ **NP**
- NRCL [AW15] $P(x, a, y) : x \neq b, x \neq c$ **NP**
- SCL [FW19] $P(a, a, b)$ **P**

BS Model Complications
Lengthy Propagations

$$N = \left\{ \begin{array}{l} 1 : P(0,0,0,0) \\ 2 : \neg P(x_1,x_2,x_3,0) \lor P(x_1,x_2,x_3,1) \\ 3 : \neg P(x_1,x_2,0,1) \lor P(x_1,x_2,1,0) \\ 4 : \neg P(x_1,0,1,1) \lor P(x_1,1,0,0) \\ 5 : \neg P(0,1,1,1) \lor P(1,0,0,0) \\ 6 : \neg P(1,1,1,1) \end{array} \right\}$$

$$\begin{array}{ll} (\epsilon, N', \top) & \Rightarrow_{\text{BS}} (P(0,0,0,0)^{1:}, N, \top) \\ & \Rightarrow_{\text{BS}} (P(0,0,0,0)^{1:} P(0,0,0,1)^{2:}, N, \top) \\ & \Rightarrow_{\text{BS}} (P(0,0,0,0)^{1:} P(0,0,0,1)^{2:} P(0,0,1,0)^{3:}, N, \top) \\ & \ldots \\ & \Rightarrow_{\text{BS}} (\ldots P(1,1,1,1)^{2:}, N, \top) \end{array}$$

ME, DPLL(SX), NRCL, SCL

BS Model Complications
Short Resolution Proof

$$N = \left\{ \begin{array}{l} 1: P(0,0,0,0) \\ 2: \neg P(x_1,x_2,x_3,0) \lor P(x_1,x_2,x_3,1) \\ 3: \neg P(x_1,x_2,0,1) \lor P(x_1,x_2,1,0) \\ 4: \neg P(x_1,0,1,1) \lor P(x_1,1,0,0) \\ 5: \neg P(0,1,1,1) \lor P(1,0,0,0) \\ 6: \neg P(1,1,1,1) \end{array} \right\}$$

2.2 Res 3.1	$7: \neg P(x_1,x_2,0,0) \lor P(x_1,x_2,1,0)$
7.2 Res 2.1	$8: \neg P(x_1,x_2,0,0) \lor P(x_1,x_2,1,1)$
8.2 Res 4.1	$9: \neg P(x_1,0,0,0) \lor P(x_1,1,0,0)$
9.2 Res 8.1	$10: \neg P(x_1,0,0,0) \lor P(x_1,1,1,1)$
10.2 Res 5.1	$11: \neg P(0,0,0,0) \lor P(1,0,0,0)$
11.2 Res 10.1	$12: \neg P(0,0,0,0) \lor P(1,1,1,1)$
12.1 Res 6.1	$13: \bot$

BS Model Complications
Immediate Conflict

$$N = \{\neg R(a,x) \vee \neg R(b,x), \ldots\}$$

$$(\epsilon, N', \top) \Rightarrow_{\text{BS}} (R(y,z)^1, N, \top)$$
$$\Rightarrow_{\text{BS}} (R(y,z)^1, N, \neg R(a,x) \vee \neg R(b,x))$$

Theorem

There is always a decision without immediate conflict.

ME, DPLL(SX), NRCL, SCL

BS Model Complications
Inconsistent Model Representation

$$N = \{\neg R(a,x) \lor P(x), \neg P(x) \lor \neg R(b,x), \ldots\}$$

$$(\epsilon, N', \top) \Rightarrow_{\mathsf{BS}} (R(y,z)^1, N, \top)$$

$$\Rightarrow^*_{\mathsf{BS}} (R(y,z)^1 P(x)^{\neg R(a,x) \lor P(x)} \neg R(b,x)^{\neg P(x) \lor \neg R(b,x)}, N, \top)$$

Theorem

There is always a way to repair the model.

ME, DPLL(SX), NRCL, SCL

BS Model Complications Equality

$$N = \{\neg R(a,b) \lor P(a), \neg P(a) \lor \neg R(b,a), \neg P(a) \lor a \approx b, \ldots\}$$

$$(\epsilon, N', \top) \Rightarrow_{\text{BS}} (R(a,b)^1, N, \top)$$

$$\Rightarrow^*_{\text{BS}} (R(a,b)^1 P(a)^{\neg R(a,b)\lor P(a)} \neg R(b,a)^{\neg P(a)\lor\neg R(b,a)} a \approx b^{\neg P(a)\lor a\approx b}, N, \top)$$

There is currently no “nice” solution to BSR.

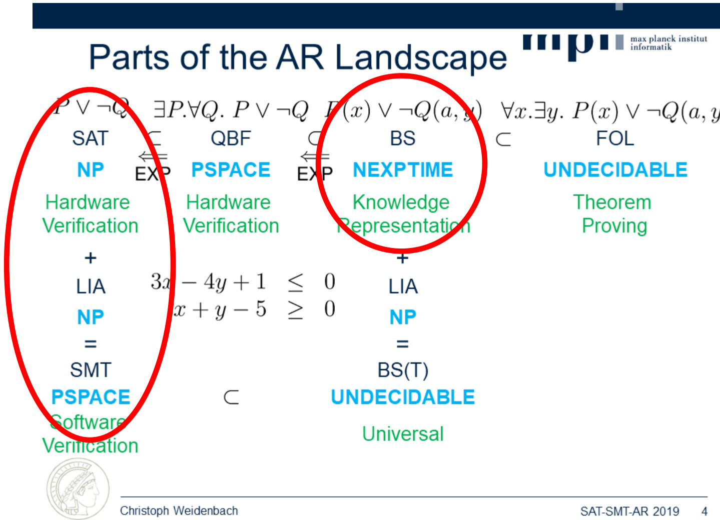
Non-Redundant Clauses

Theorem [AW15,FW19]

If $(L_1 \ldots L_n, N, C)$ is a BS Backtracking state with eager Conflict and Propagate, then $N^{\prec C} \not\models C$ where $L_1 \prec \ldots \prec L_n$. Non-Redundancy is NEXPTIME-complete.

This holds for NRCL, SCL but probably also for variants of DPLL(SX) and ME.

Summary



SAT works because:

- Explicit, efficient model generation
- Non-redundant clause learning
- No waste of computing time

SMT works because:

- Abstraction
- SAT works
- Explicit, efficient model generation CDCL(LIA)
- No waste of computing time
- No notion of non-redundant clause learning CDCL(LIA)

BS works because:

- Non-redundant clause learning
- In general, no efficient model generation
- No waste of computing time with SCL
- Exhaustive Propagation, Equality

BS Approximation Refinement

BS clause set $N \quad \Rightarrow_{\text{Abstr}} N'$ reasoning in N' is less complex

use reasoning on N' to check satisfiability of N

Instgen [KG03,K13]

$$P(x) \lor Q(x,y) \quad \Rightarrow_{\text{Apr}} P(c) \lor \neg Q(c,c)$$

Approximation to SAT solver: unsat

sat

Refine N such that the spurious model vanishes

SUP(AR) [TW17]

$$P(x) \lor Q(x,y) \quad \Rightarrow_{\text{Apr}} P(x) \lor \neg Q(z,y)$$

Approximation to MSLH solver: unsat

sat

Refine N such that the spurious proof vanishes

BS Ordered Resolution

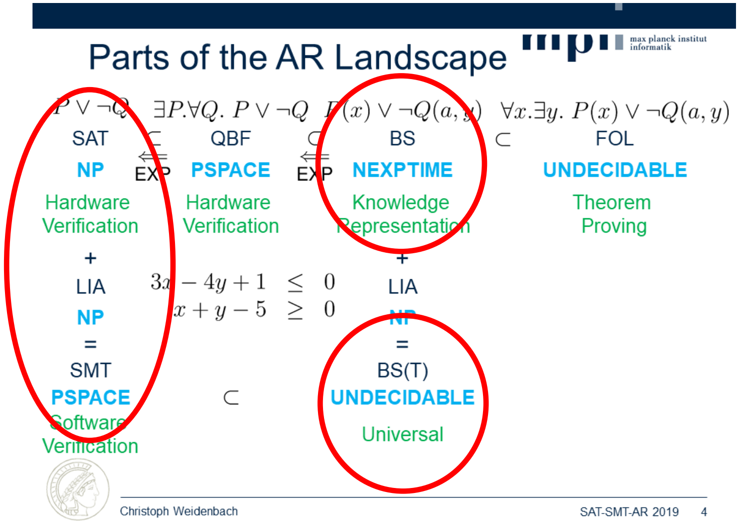
$$N = \left\{ \begin{array}{l} 1 : \underline{P(0,0,0,0)} \\ 2 : \neg P(x_1,x_2,x_3,0) \lor \underline{P(x_1,x_2,x_3,1)} \\ 3 : \neg P(x_1,x_2,0,1) \lor \underline{P(x_1,x_2,1,0)} \\ 4 : \neg P(x_1,0,1,1) \lor \underline{P(x_1,1,0,0)} \\ 5 : \neg P(0,1,1,1) \lor \underline{P(1,0,0,0)} \end{array} \right\}$$

Take an ordering with $1 \succ 0$ then all positive literals are maximal.

There is no ordered resolution inference, N is satisfiable.

Explicit model building: exponential propagation.

BS(T)



Already undecidable for LRA and one monadic predicate P.

BS(T) is decidable if:

T is only simple bounds: $x \neq 3, x < 7$

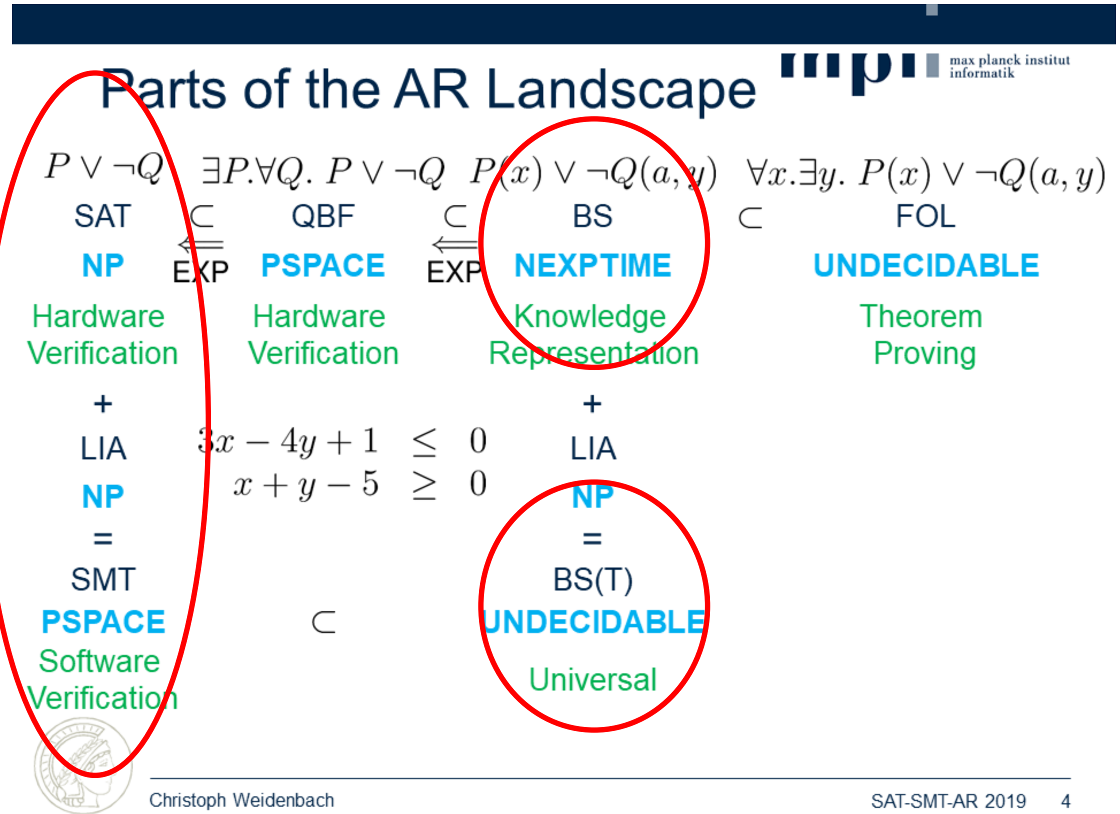
No recursive structure on the BS side

T is over the integers and all variables have an upper and lower bound

T is ??

Thanks for Your Attention

max planck institut
informatik



References do not reflect history.

References

[AW15] Gábor Alagi and Christoph Weidenbach. NRCL - A model building approach to the bernays-schönfinkel fragment. In Carsten Lutz and Silvio Ranise, editors, *Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings*, volume 9322 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2015.

[BFT06] Peter Baumgartner, Alexander Fuchs, and Cesare Tinelli. Lemma learning in the model evolution calculus. In *LPAR*, volume 4246 of *Lecture Notes in Computer Science*, pages 572–586. Springer, 2006.

[BGG96] Egon Börger, Erich Grädel, and Yuri Gurevich. *The classical decision problem*. Perspectives in mathematical logic. Springer, 1996.

[BHvMW09] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[BS28] Paul Bernays and Moses Schönfinkel. Zum entscheidungsproblem der mathematischen logik. *Mathematische Annalen*, 99:342–372, 1928.

[Coo71] S.A. Cook. The complexity of theorem proving procedures. In *Proceedings Third ACM Symposium on the Theory of Computing, STOC*, pages 151–158. ACM, 1971.

[FLHT01] Christian G. Fermüller, Alexander Leitsch, Ullrich Hustadt, and Tanel Tamet. Resolution decision procedures. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 25, pages 1791–1849. Elsevier, 2001.

[FW19] Alberto Fiori and Christoph Weidenbach. Scl clause learning from simple models. In Pascal Fontaine, editor, *27th International Conference on Automated Deduction, CADE-27*, volume 11716 of *LNAI*. Springer, 2019.

[GK03] Harald Ganzinger and Konstantin Korovin. New directions in instatiation–based theorem proving. In Samson Abramsky, editor, *18th Annual IEEE Symposium on Logic in Computer Science, LICS'03*, LICS'03, pages 55–64. IEEE Computer Society, 2003.

[BS97] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In Benjamin Kuipers and Bonnie L. Webber, editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA.*, pages 203–208, 1997.

[KLPS16] Benjamin Kaufmann, Nicola Leone, Simona Perri, and Torsten Schaub. Grounding and solving in answer set programming. *AI Magazine*, 37(3):25–32, 2016.

[Kor13] Konstantin Korovin. Inst-gen - A modular approach to instantiation-based automated reasoning. In Andrei Voronkov and Christoph Weidenbach, editors, *Programming Logics - Essays in Memory of Harald Ganzinger*, volume 7797 of *Lecture Notes in Computer Science*, pages 239–270. Springer, 2013.

[Lew79] Harry R. Lewis. *Unsolvable Classes of Quantifica-*

tional Formulas. Addison-Wesley, 1979.

[Lew80] Harry R. Lewis. Complexity results for classes of quantificational formulas. *Journal of Compututer and System Sciences*, 21(3):317–353, 1980.

[NOT06] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving sat and sat modulo theories: From an abstract davis–putnam–logemann–loveland procedure to dpll(t). *Journal of the ACM*, 53:937–977, November 2006.

[Pap81] Christos H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981.

[PMB10] Ruzica Piskac, Leonardo Mendonça de Moura, and Nikolaj Bjørner. Deciding effectively propositional logic using DPLL and substitution sets. *Journal of Automated Reasoning*, 44(4):401–424, 2010.

[Pla84] David A. Plaisted. Complete problems in the first-order predicate calculus. *Journal of Computer and System Sciences*, 29:8–35, 1984.

[Pos46] Emil L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946.

[SS96] João P. Marques Silva and Karem A. Sakallah. Grasp - a new search algorithm for satisfiability. In *International Conference on Computer Aided Design, ICCAD*, pages 220–227. IEEE Computer Society Press, 1996.

[TW17] Andreas Teucke and Christoph Weidenbach. Decidability of the monadic shallow linear first-order fragment with straight dismatching constraints. In Leonardo de Moura, editor, *Automated Deduction - CADE 26 - 26th International Conference on Au-*

tomated Deduction, Gothenburg, Sweden, August 6-11, 2017, Proceedings, volume 10395 of *Lecture Notes in Computer Science*, pages 202–219. Springer, 2017.

[vH67] van Heijenoort. *From Frege to Goedel - A Source Book in Mathematical Logic, 1979-1931.* Source Books in the History of the Sciences. Harvard University Press, Cambridge - Massachusetts, London - England, 1967.

[Wei15] Christoph Weidenbach. Automated reasoning building blocks. In Roland Meyer, André Platzer, and Heike Wehrheim, editors, *Correct System Design - Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday, Oldenburg, Germany, September 8-9, 2015. Proceedings*, volume 9360 of *Lecture Notes in Computer Science*, pages 172–188. Springer, 2015.