3 Undecidability

A logical calculus, i.e. a formal (purely syntactical) proof system for an underlying logic, is *complete* if every (semantically) valid formula is (syntactically) provable from its premises by the calculus. In other words, every sentence that holds in all possible models for its premises is derivable from its premises by applying rules of the formal system only. Additional we expect a useful calculus to be *sound*, that is, every (syntactically) provable formula (semantically) holds in any model for its premises. Without premises a sentence has to be provable if and only if it holds in any interpretation of its signature.

We first state some completeness, undecidability, and other fundamental theorems about first order logic in Section 3.1. Then we enumerate decidable fragments of first order logic which can be described purely syntactically in Section 3.2 on the next page. We conclude this chapter with a look at decidable first-order theories in Section 3.3 on page 17, which are not necessarily contained in one of the syntactically describable and decidable fragments of first-order logic.

3.1 Theorems about First Order Logic

The fundamental theorems about first order logic were already proven in the first half of the 20th century. They outline the possibilities and limitations of any attempt to create a general procedure that checks the validity of arbitrary first order sentences.

Theorem 3.1 (Soundness)**.** *The inference rules of natural deduction (see Definition 2.26 on page 7) are sound.*

Proof. We can prove the soundness of each inference rule by case distinction and the use of the semantic definition of validity. □

Theorem 3.2 (Gödels Vollständigkeitssatz [11], 1929/1930)**.** *„Es gibt einen Kalkül der Prädikatenlogik erster Stufe derart, dass für jede Formelmenge* Γ *und für jede Formel* φ *gilt:* φ *folgt genau dann aus* Γ*, wenn* φ *im Kalkül aus* Γ *hergeleitet werden kann.“*

Theorem 3.3 ([3])**.** *Natural deduction is a complete calculus, i.e. a proof* $\Gamma \vdash G$ *exists, whenever* $\Gamma \models G$*.*

Lemma 3.4 (Refutation)**.** *A formula is valid if and only if its negation is not satisfiable.*

Proof. By definition of the semantics of negation, validity, and satisfiability. □

Theorem 3.5 (Undecidability [6, 17], Church 1936, Turing 1937.)**.** *The satisfiability problem for first-order logic is undecidable.*

Theorem 3.6 (Trakhtenbort 1950, Craig 1950)**.** *The satisfiability problem for first-order logic on* finite *structures (domains) is undecidable.*

Definition 3.7 (Finite model property)**.** A logic has the finite model property if each non-theorem is falsified by some finite model.

Theorem 3.8 (Compactness, Gödel 1930, Maltsev 1936)**.** *If every finite subset of a set of formulas S has a model then S has a model.*

Theorem 3.9 (Löwenheim Skolem, 1915, 1920)**.** *If a set of formulas S has a model then S has a countable model.*

Theorem 3.10 (Herbrand, 1930)**.** *Let S be a set of clauses without equality. Then the following statements are equivalent.*

- *S is satisfiable.*
- *S has a Herbrand model.*
- *Every finite subset of all ground instances of S has a Herbrand model.*

Corollary 3.11. *Let S be a set of clauses without equality. Then S is unsatisfiable if and only if there exists an unsatisfiable finite set of ground instances of S.*

Lemma 3.12. *Skolemization preserves satisfiability.*

Lemma 3.13. *Tseytin's transformation preserves satisfiability.*

Lemma 3.14. *Herbrandization preserves validity.*

Lemma 3.15. *With Skolemization and Tseytin transformation we can effectively transform a arbitrary first-order formula into an equisatisfiable set of clauses.*

3.2 Decidable Fragments of First Order Logic

Validity and satisfiability in general are undecidable in First Order Logic. However this section presents purely syntactical defined (tiny) fragments of first-order logic where satisfiability is decidable.

Definition 3.16 ([5])**.** We define classes, i.e. sets of first-order formulae, with triples

$$[\,\Pi, (p_1, p_2, \ldots, p_j), (f_1, f_2, \ldots, f_k)\,]_{(\approx)} \subseteq [\,all, all, all\,]_{\approx}$$

where $\Pi = \exists\forall_1 \ldots \exists\forall_n$, $\exists\forall_i \in \{\forall, \exists\}$ describes the structure of the quantifier prefix (without variables) of the formulae in `PNF` and *all* in the first position denotes arbitrary prefixes of any length. The value p_i represents the maximal number of predicate symbols with arity i and $p_\ell = 0$ for $\ell > j$. Similar the value f_i represents the maximal number of function symbols with arity i and $f_\ell = 0$ for $\ell > k$. In both cases *all* denotes an arbitrary number of symbols with any arity. The equality symbol is not counted as binary predicate symbol. Instead, the absence or presence of equality in the formulae is indicated by the absence or presence of a subscript $\approx$.

Example 3.17. The monadic predicate calculus includes formulae with arbitrary quantifier prefixes, arbitrary many unary predicate symbols, the equality symbol, but no function symbols.

$$[\,all, (\omega), (1)\,]_{\approx} \quad \supsetneq \quad [\,all, (\omega), (0)\,]_{\approx} \qquad \text{(Löwenheim 1925, Kalmár 1929)}$$

Example 3.18. The Ackermann prefix class contains formulae with arbitrary many existential quantifiers, but just one universal quantifier. It contains arbitrary many predicate symbols with arbitrary arities, the equality symbol, but no function symbols.

$$[\,\exists^*\forall\exists^*, all, (1)\,]_{\approx} \quad \supsetneq \quad [\,\exists^*\forall\exists^*, all, (0)\,]_{\approx} \qquad \text{(Ackermann 1928)}$$

Remark. One unary function symbol can be added to these fragments of first order logic above without loosing decidability (see Table 3.2).

$[\,\exists^*\forall^*, all, (0)\,]_{\approx}$	(Bernays, Schönfinkel 1928, Ramsey 1932)
$[\,\exists^*\forall^2\exists^*, all, (0)\,]$	(Gödel 1932, Kalmár 1933, Schütte 1934)
$[\,all, (\omega), (\omega)\,]$	(Löb 1967, Gurevich 1969)
$[\,\exists^*\forall\exists^*, all, all\,]$	(Gurevich 1973)
$[\,\exists^*, all, all\,]_{\approx}$	(Gurevich 1976)

Table 3.1: Decidable prefix classes with finite model property

$[\,all, (\omega), (1)\,]_{mEQ}$	(Rabin 1969)
$[\,\exists^*\forall\exists^*, all, (1)\,]_{\approx}$	(Shelah 1977)

Table 3.2: Decidable prefix classes with infinity axioms.

Lemma 3.19 ([5])**.** *Satisfiability is decidable in all prefix classes from Tables 3.1 and 3.2. Each of theses classes is closed under conjunction with respect to satisfiability.*

3.3 Theories in First Order Logic

We follow the definitions and examples of first order theories by A. Middeldorp in [19].

Definition 3.20 (Theory)**.** A *first-order theory* is a pair of a first-order signature and the possible infinite conjunction $\bigwedge_i A_i$ of first-order formulae, i.e. the axioms, over the theory's signature. A theory is *consistent* if the contradiction is not derivable. A theory

is satisfiable if there exists a model for its axioms. A *theorem* is a sentence over the theory's signature, i.e. a closed formula, that holds in any model for the theory's axioms.

$$\bigwedge_i A_i \models \mathsf{theorem} \quad \text{or} \quad \bigwedge_i A_i \rightarrow \mathsf{theorem}$$

A theory is *decidable* if there is a decision procedure whether an arbitrary sentence holds in the theory, i.e. if the sentence is a consequence of the axioms.

Example 3.21. A theory with axioms $\forall x\, \mathsf{P}(x)$ and $\exists x\, \neg\mathsf{P}(x)$ is neither consistent nor satisfiable.

Lemma 3.22. *A first order theory is consistent if and only if it is satisfiable.*

Even if we can decide that a sentence is not a theorem in a theory, e.g. if we find a model for the axioms where the sentence does not hold, we still cannot conclude that the negation of the sentence is a theorem.

Example 3.23. The sentence $F = \forall x \forall y\, (x \approx y)$ is definitly not a theorem in many theories, but its negation $\neg F \equiv \exists x \exists y\, (x \not\approx y)$ is not a theorem in any theory that allows models with a universe of cardinality one.

Definition 3.24. A (consistent) theory is *complete* iff for any sentence in the theory (either) the sentence or its negation is a theorem in the theory.

Remark. In refutational theorem proving we show the unsatisfiability of a negated sentence, i.e. a *conjecture*, in conjunction with the axioms to conclude that the conjecture is indeed a theorem.

$$\neg \left(\bigwedge_i A_i \rightarrow \mathsf{conj} \right) \equiv \neg \left(\neg \bigwedge_i A_i \vee \mathsf{conj} \right) \equiv \bigwedge_i A_i \wedge \neg\mathsf{conj}$$

3.3.1 Theory of equality

The following equivalence and congruence axioms form the theory of equality over a first order signature.

Definition 3.25 (Equivalence)**.** A binary relation $\approx$ over a domain is an equivalence relation if and only if the following axioms hold over the given domain.

$$\begin{aligned} \forall x\ (x \approx x) && \text{reflexivity} \\ \forall x \forall y\ (x \approx y \rightarrow y \approx x) && \text{symmetry} \\ \forall x \forall y \forall z\ (x \approx y \wedge y \approx z \rightarrow x \approx z) && \text{transitivity} \end{aligned}$$

Remark. The equivalence axioms are expressible within the decidable first-order fragments $[\,\forall^3, (0), (0)\,]_\approx \subsetneq [\,all, (\omega), (1)\,]_\approx$.

Definition 3.26 ($\vec{x}$-Notation)**.** Occasionally we may abbreviate a sequence of n variables by $\vec{x}$. Then we write $f(\vec{x})$ for first-order expression $f(x_1, \ldots, x_n)$ with n-ary function or predicate symbol f, a single equation $\vec{x} \approx \vec{y}$ for the conjunction of n equations $x_1 \approx y_1 \wedge \ldots \wedge x_n \approx y_n$, and $\forall\vec{x}$ for the sequence of quantified variables $\forall x_1 \ldots \forall x_n$.

Definition 3.27 (Congruence schemata)**.** An equivalence relation $\approx$ is a congruence relation if and only if the following formulae hold

$$\forall\vec{x}\forall\vec{y}\,(\vec{x} \approx \vec{y} \to \mathsf{f}(\vec{x}) \approx \mathsf{f}(\vec{y})) \qquad \text{for all } \mathsf{f} \in \mathcal{F}_{\mathsf{f}}^{(n)}$$
$$\forall\vec{x}\forall\vec{y}\,(\vec{x} \approx \vec{y} \to (\mathsf{P}(\vec{x}) \to \mathsf{P}(\vec{y}))) \qquad \text{for all } \mathsf{P} \in \mathcal{F}_{\mathsf{P}}^{(n)}$$

Lemma 3.28. *The equivalence and congruence axioms of equality are provable with natural deduction (Definition 2.26 on page 7, Table 2.1 on page 7, Table 2.2 on page 8, and Table 2.3 on page 8).*

Proof. For brevity we skip the quantifier introductions (and handle variables like constants) for symmetry, transitivity, and congruence. Additionally we just show congruence for a unary function and a unary predicate symbol.

1	$\mathsf{c}_0 \quad \mathsf{c}_0 = \mathsf{c}_0$	$=i$
2	$\forall x\,(x = x)$	$\forall i, 1, \{x \mapsto \mathsf{c}_0\}$

1	$y = y$	$=i$
2	$x = y$	assume
3	$y \neq x$	assume
4	$y \neq y$	$=e, 2, 3$
5	$\bot$	$\neg e, 1, 4$
6	$y = x$	PBC, 3–5
7	$x = y \to y = x$	$\to i, 1$–5

1	$\mathsf{f}(y) = \mathsf{f}(y)$	$=i$
2	$x = y$	assume
3	$\mathsf{f}(x) \neq \mathsf{f}(y)$	assume
4	$\mathsf{f}(y) \neq \mathsf{f}(y)$	$=e, 2, 3$
5	$\bot$	$\neg e, 1, 4$
6	$\mathsf{f}(x) = \mathsf{f}(y)$	PBC, 3–5
7	$x = y \to \mathsf{f}(x) = \mathsf{f}(y)$	$\to i, 1$–5

1	$x = y \wedge y = z$	assume
2	$y = z$	$\wedge e_2$
3	$x = y$	$\wedge e_1, 2$
4	$x = z$	$=e, 2, 3$
5	$x = y \wedge y = z \to x = z$	$\to i, 2$–4

1	$x = y$	assume
2	$\mathsf{P}(x)$	assume
3	$\mathsf{P}(y)$	$=e, 1, 2$
4	$\mathsf{P}(x) \to \mathsf{P}(y)$	$\to i, 2$–3
5	$x = y \to (\mathsf{P}(x) \to \mathsf{P}(y))$	$\to i, 1$–4

□

3.3.2 Natural numbers

The following axioms characterize natural numbers, addition, and multiplication.

Definition 3.29 (Natural Numbers)**.** We introduce a constant symbol 0, a unary successor symbol $\mathsf{s} \in \mathcal{F}^{(1)}$ — congruence must hold — and restrict the possible models

with two axioms.

$$\begin{aligned}
\forall x\,(\mathsf{s}(x) \not\approx 0) &\qquad \text{zero is smallest}\\
\forall x \forall y\,(\mathsf{s}(x) \approx \mathsf{s}(y) \to x \approx y) &\qquad \text{injectivity of } \mathsf{s}\\
\forall x \forall y\,(x \approx y \to \mathsf{s}(x) \approx \mathsf{s}(y)) &\qquad \text{congruence of } \mathsf{s}\\
\underbrace{G(0)}_{\text{base}} \wedge \forall x'\ \underbrace{(G(x') \to G(\mathsf{s}(x')))}_{\text{step case}} \to \forall x\, G(x) &\qquad \text{induction schema}
\end{aligned}$$

Remark. The axioms of natural numbers are expressible within the decidable first-order fragments $[\,\exists\forall^2, (0), (1)\,]_{\approx} \subsetneq [\,all, (\omega), (1)\,]_{\approx}$.

Example 3.30. We may prove $\forall x\,(\mathsf{s}(x) \not\approx x)$ with $G(x) = \mathsf{s}(x) \not\approx x$ by induction.

$$\underbrace{\mathsf{s}(0) \not\approx 0}_{\text{base}} \wedge \forall x'\ \underbrace{(\mathsf{s}(x') \not\approx x' \to \mathsf{s}(\mathsf{s}(x')) \not\approx \mathsf{s}(x'))}_{\text{step case}} \to \forall x\, \mathsf{s}(x) \not\approx x$$

Definition 3.31 (Addition)**.** We introduce the binary addition symbol $+ \in \mathcal{F}_{\mathsf{f}}^{(2)}$ — congruence must hold — with two axioms about defining equalities of sums.

$$\begin{aligned}
\forall x\,(x + 0 \approx x) &\qquad \text{addition of zero}\\
\forall x \forall y\,(x + \mathsf{s}(y)) \approx \mathsf{s}(x + y) &\qquad \text{addition of non-zero}\\
\forall x_1 \forall x_2 \forall y_1 \forall y_2\,(x_1 \approx y_1 \wedge x_2 \approx y_2 \to x_1 + y_1 \approx x_2 + y_2) &\qquad \text{congruence of } +
\end{aligned}$$

Example 3.32.

$$\mathsf{s}(\mathsf{s}(\mathsf{s}(0))) + \mathsf{s}(\mathsf{s}(0)) \approx \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))) + \mathsf{s}(0)) \approx \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))) + 0)) \approx \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))))$$

Remark. Already the axioms of addition (without function congruence) are only contained in a non-decidable first order fragment $[\,\forall^2, (0), (1, 1)\,]_{\approx}$. Still they are part of a decidable first-order theory.

Theorem 3.33. *Presburger arithmetic (Mojżesz Presburger, 1929), i.e. the first-order theory that includes the axioms for equality, natural numbers, induction schemata, and addition, is consistent, complete and decidable. The computational complexity of the decision problem is at least doubly exponential $2^{2^{cn}}$ (Fischer and Rabin, 1974), but less than triple exponential (Oppen, 1978. Berman, 1980).*

Remark. Sentences in Presburger arithmetic are contained in $[\,all, (0), (1, 1)\,]_{\approx}$.

Definition 3.34 (Multiplication)**.** We introduce the binary multiplication symbol $\times \in \mathcal{F}_{\mathsf{f}}^{(2)}$ — congruence must hold — with two axioms about defining equalities of products.

$$\begin{aligned}
\forall x\,(x \times 0 \approx 0) &\qquad \text{multiplication by zero}\\
\forall x \forall y\,(x \times \mathsf{s}(y) \approx (x \times y) + x) &\qquad \text{multiplication by non-zero}\\
\forall x_1 \forall x_2 \forall y_1 \forall y_2\,(x_1 \approx y_1 \wedge x_2 \approx y_2 \to x_1 \times y_1 \approx x_2 \times y_2) &\qquad \text{congruence of } \times
\end{aligned}$$

Theorem 3.35. *Peano Arithmetic (Giuseppe Peano, 1889), i.e. the first-order theory that extends Presburger Arithmetic with multiplication, is incomplete (Gödel's second incompleteness theorem in 1932) and undecidable.*

Remark. Sentences in Peano arithmetic are contained in $[\,all, (0), (1,2)\,]_{\approx}$.

Theorem 3.36. *The axioms of Peano Arithmetic appear consistent (Gentzen, 1936).*

Lemma 3.37 (ACN)**.** *Addition and Multiplication on natural numbers are associative, commutative, and determine neutral elements.*

$$\forall x \forall y \forall z\, (x \circ (y \circ z) \approx (x \circ y) \circ z) \qquad \text{associativity of } \circ \in \{+, \times\}$$
$$\forall x \forall y\, (x \circ y \approx y \circ x) \qquad \text{commutativity of } \circ \in \{+, \times\}$$
$$\forall x\, (x + 0 \approx x \wedge 0 + x \approx x) \qquad \text{neutral element for } +$$
$$\forall x\, (x \times \mathsf{s}(0) \approx x \wedge \mathsf{s}(0) \times x \approx x) \qquad \text{neutral element for } \times$$

List of Figures

List of Tables

Bibliography

[1] L. Albert, R. Casas, and F. Fages. Average-case analysis of unification algorithms. *Theoretical Computer Science*, 113(1):3 – 34, 1993.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, New York, NY, USA, 1998.

[3] D. W. Bennett. An elementary completeness proof for a system of natural deduction. *Notre Dame J. Formal Logic*, 14(3):430–432, 07 1973.

[4] A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.

[5] E. Börger, E. Grädel, and Y. Gurevich. *The classical decision problem*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1997. With an appendix by Cyril Allauzen and Bruno Durand.

[6] A. Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41, 1936.

[7] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, July 1962.

[8] M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960.

[9] H. Ganzinger and K. Korovin. Integrating Equational Reasoning into Instantiation-Based Theorem Proving. In *18th CSL 2004. Proceedings*, volume 3210 of *LNCS*, pages 71–84, 2004.

[10] P. C. Gilmore. A proof method for quantification theory: Its justification and realization. *IBM Journal of Research and Development*, 4(1):28–35, Jan 1960.

[11] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37(1):349–360, Dec 1930.

[12] P. Graf and D. Fehrer. *Term Indexing*, pages 125–147. Springer Netherlands, Dordrecht, 1998.

[13] J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[14] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning About Systems*. Cambridge University Press, New York, NY, USA, 2004.

[15] K. Korovin. Inst-Gen – a Modular Approch. In *IJCAR 2008. Proceedings*, pages 292–298.

[16] D. Kroening and O. Strichman. *Decision Procedures: An Algorithmic Point of View*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[17] T. A. M. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265.

[18] A. Middeldorp. Lecture Notes – Term Rewriting, 2015.

[19] A. Middeldorp. Lecture Notes – Logic, 2016.

[20] G. Moser. Lecture Notes – Module Automated Reasoning, 2013.

[21] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 371–443. Elsevier, 2001.

[22] N. Olivetti and A. Tiwari, editors. *Automated Reasoning - 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27 - July 2, 2016, Proceedings*, volume 9706 of *Lecture Notes in Computer Science*. Springer, 2016.

[23] D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2(3):293 – 304, 1986.

[24] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, Jan. 1965.

[25] S. Schulz and M. Möhrmann. Performance of clause selection heuristics for saturation-based theorem proving. In N. Olivetti and A. Tiwari, editors, *Automated Reasoning - 8th International Joint Conference, IJCAR 2016, Coimbra, Portugal, June 27 - July 2, 2016, Proceedings*, volume 9706 of *Lecture Notes in Computer Science*, pages 330–345. Springer, 2016.

[26] C. Sticksel. *Efficient Equational Reasoning for the Inst-Gen framework*. PhD thesis, School of Computer Science, University of Manchester, 2011.

[27] G. S. Tseitin. On the complexity of derivations in the propositional calculus. *Studies in Constructive Mathematics and Mathematical Logic*, Part II:115–125, 1970.

[28] L. Wos, G. A. Robinson, and D. F. Carson. Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM*, 12(4):536–541, Oct. 1965.