Inductive theorem proving using tree grammars

Kandidat: Gabriel Ebner
Prüfer: Gernot Tragler, Stefan Hetzl, Nicolas Peltier

Der Satz von Herbrand [8], ein grundlegendes Ergebnis der Logik und Beweistheorie, charakterisiert die Gültigkeit von quantifizierten Formeln der klassischen Logik erster Ordnung durch die Existenz einer tautologischen endlichen Menge von quantorenfreien Grundinstanzen. Im einfachsten Fall entspricht einer gültigen, rein existenziellen Formel $\exists x\, \varphi(x)$ eine tautologische Disjunktion $\varphi(t_1) \lor \cdots \lor \varphi(t_n)$, eine sogenannte Herbrand-Disjunktion.

Schnittfreie Beweise enthalten die zur Bildung einer solchen Herbrand-Disjunktion notwendigen Terme [4] unmittelbar in ihren Quantorenschlüssen. Die fundamentalste Operation der Beweistheorie, die Gentzensche Schnittelimination [7], enthält einen Algorithmus, der aus Beweisen die Schlussfigur des Schnitts eliminiert, und somit aus einem Beweis einer rein existenziellen Formel eine Herbrand-Disjunktion berechnet.

Ein moderner Ansatz, um diese Schnittelimination auf Ebene der Quantorenschlüsse mittels formalsprachlichen Methoden zu verstehen, wurde von Hetzl vorgestellt [9]: jedem Beweis π in einer geeigneten Klasse wird eine Grammatik $G(\pi)$ auf solche Art zugeordnet, dass die Grammatik unter der Schnitteliminationsoperation erhalten bleibt. Die von der Grammatik erzeugte Sprache $L(G(\pi))$ ist dann isomorph zu einer Herbrand-Disjunktion; um die Schnittelimination zu verstehen, reicht es folglich, diese Grammatiken zu verstehen.

Die erste Instanz dieses Homomorphismus in [9] verbindet Beweise von schwach quantifizierten Pränexsequenten, deren Schnittformeln aus Pränexformeln ohne Quantorenalternation bestehen, mit vektoriellen totalrigiden Baumgrammatiken (VTRATGs). Darauffolgende Verallgemeinerungen auf umfangreichere und kompliziertere Beweisklassen verlangen dementsprechend ausdrucksstärkere Grammatikklassen [5, 10, 2].

Für Beweise mit Induktion ist es sogar erforderlich, den Begriff der Herbrand-Disjunktion selbst zu verallgemeinern. Ein Beweis π von $\forall x\, \varphi(x)$ in einer geeigneten Klasse von Beweisen mit Induktion induziert auf natürliche Weise für jeden Konstruktorterm t einen induktionsfreien Instanzbeweis π_t von $\varphi(t)$. Somit entsteht eine durch Konstruktorterme indizierte Familie von Herbrand-Disjunktionen $(L(\pi_t))_t$, die jeweils durch die von der Grammatik des Induktionsbeweises instanzierten Grammatik $I(G(\pi), t) \supseteq L(\pi_t)$ überdeckt werden.

Die Umkehrung der Schnittelimination ist vielleicht die größte Herausforderung der Beweistheorie überhaupt. Beweise mit Schnitt können im allgemeinen nicht-elementar kleiner sein als schnittfreie Beweise; eine Umkehrung der Schnittelimination, in anderen Worten eine *Schnitteinführung*, ermöglicht daher direkt eine enorme Beweiskompression. Weiters fasst die Schlussfigur des Schnitts auf formale Weise den Begriff des mathematischen Satzes; die Schnitteinführung führt somit auch Hilfssätze in Beweise ein, und entdeckt folglich in einem gewissen Sinne sogar neue mathematischen Konzepte.

Auf Ebene der Gentzenschen Schnittreduktionsrelation ist die Umkehrung der Schnittelimination jedoch völlig aussichtslos: allein um nur einen einzigen Schritt umzukehren, gibt es schon unendliche viele Möglichkeiten (man denke nur an den Fall der Verdünnungsreduktion). Unter dem Bild des grammatikalischen Homomorphismus hingegen betrachtet, entbart sich die Schnitteinführung nicht nur als praktisch erfolgreich durchführbar [6], sondern sie zerfällt sogar in zwei klar abgegrenzte Teilprobleme. Angenommen wir beginnen mit einem schnittfreien Beweis π. Dann bildet ersteres Teilproblem das direkte formalsprachliche Gegenstück zur Umkehrung der Schnittelimination, und besteht aus der Lösung des Überdeckungsproblem, also eine Grammatik G zu finden, sodass $L(G) \supseteq L(\pi)$. Das zweite Teilproblem übersetzt die Grammatik aus der formalsprachlichen Welt zurück zu einem Beweis π, sodass $G(\pi) = G$.

Derselbe grammatikalische Ansatz glückt auch für die Einführung von Induktionsschlussfiguren, wobei im Unterschied zur Schnitteinführung die Nichtanalytizität der Induktionsformeln—dass sie also nicht bereits wörtlich im zu beweisenden Sequent vorkommen—von essenziellem und unerlässlichem Charakter ist. Eberhard und Hetzl [5] schlagen diesen Ansatz als zukunftsweisendes Paradigma für das induktive Beweisen vor: schnittfreie Instanzbeweise sind durch automatische Theorembeweiser mühelos zu erzeugen, und können dann zu einem Beweis mit Induktion verallgemeinert werden.

Diese Dissertation nimmt sich das Ziel, eine praktisch einsetzbare Implementierung dieses neuen Paradigmas umzusetzen. Diesem Ziel vorangehend, erkunden wir zunächst ein reichhaltiges Spektrum an theoretischen Fragestellungen, die unser Verständnis für die zugrundelegenden Strukturen vertiefen.

Eine grundlegende Frage, die sich beim Finden von überdeckenden Grammatiken stellt, ist eine komplexitätstheoretische: wie schwierig ist es, eine kleinstmögliche überdeckende Grammatik zu finden? Wie schwierig ist es überhaupt, festzustellen, ob eine Grammatik eine gegebene Termmenge überdeckt? Oder ob zwei Grammatiken dieselbe Sprache erzeugen? Fragen dieser Art werden wir in Kapitel 3 für unterschiedliche Klassen von Grammatiken untersuchen und auch mit anderen formalsprachlichen Modellen in Beziehung setzen.

Darauffolgend ergibt sich natürlich die Frage, wie wir praktisch überdeckende Grammatiken finden können. Dazu stellen wir in Kapitel 4 drei unterschiedliche Algorithmen vor.

Das Erzeugen einer überdeckenden Grammatik ist jedoch nur der erste Zwischenschritt: danach gilt es, einen Beweis zu erzeugen, dem diese Grammatik zugeordnet ist. Konkret sind die Matrizen der Schnitt- und Induktionsformeln zu finden. Die Bedingungen, denen diese Matrizen zu genügen haben, bilden eine sogenannte Formelgleichung, welche in engem Zusammenhang mit dem Lemma von Ackermann [1] auf der theoretischen Seite, und mit bedingten Hornklauseln [3] auf der praktischen Seite stehen. In Kapitel 5 beschreiben wir diesen Zusammenhang, untersuchen Fragestellungen zur Lösbarkeit von den relevanten Formelgleichungen, und stellen zwei Lösungsalgorithmen vor.

Um Herbrand-Disjunktionen von automatischen Theorembeweisern zu erhalten, stellen wir in Kapitel 6 einen neuen Algorithmus vor. Dieser wandelt Resolutionsbeweise in Expansionsbäume—eine Verallgemeinerung von Herbrand-Disjunktionen—um, ohne als Zwischenschritt einen schnittfreien Beweis zu konstruieren.

Zum Schluss, in Kapitel 7, fügen wir die Puzzlesteine aus den vorangehenden Kapiteln zu einer vollständigen Implementierung zusammen, und evaluieren diese. Die Testergebnisse erschließen einen tiefliegenden Unterschied zwischen automatisch erzeugten Beweisen und Beweisen, die aus der Induktionselimination gewonnen sind. Diese Andersartigkeit ist selbst auf Ebene der Herbrand-Disjunktionen eindeutig erkennbar.

Literatur

[1] Wilhelm Ackermann. "Untersuchungen über das Eliminationsproblem der mathematischen Logik". In: *Mathematische Annalen* 110.1 (1935), pp. 390–413.

[2] Bahareh Afshari, Stefan Hetzl, and Graham E. Leigh. "Herbrand's theorem as higher order recursion". In: *Annals of Pure and Applied Logic* 171.6 (2020), p. 102792.

[3] Nikolaj Bjørner, Arie Gurfinkel, Ken McMillan, and Andrey Rybalchenko. "Horn clause solvers for program verification". In: *Fields of Logic and Computation II*. Springer, 2015, pp. 24–51.

[4] Samuel R. Buss. "On Herbrand's Theorem". In: *Logic and Computational Complexity*. Vol. 960. Lecture Notes in Computer Science. Springer, 1995, pp. 195–209.

[5] Sebastian Eberhard and Stefan Hetzl. "Inductive theorem proving based on tree grammars". In: *Annals of Pure and Applied Logic* 166.6 (2015), pp. 665–700.

[6] Gabriel Ebner, Stefan Hetzl, Alexander Leitsch, Giselle Reis, and Daniel Weller. "On the generation of quantified lemmas". In: *Journal of Automated Reasoning* (2018), pp. 1–32.

[7] Gerhard Gentzen. "Untersuchungen über das logische Schließen I". In: *Mathematische Zeitschrift* 39.1 (1935), pp. 176–210.

[8] Jacques Herbrand. "Recherches sur la théorie de la démonstration". PhD thesis. Université de Paris, 1930.

[9] Stefan Hetzl. "Applying Tree Languages in Proof Theory". In: *Language and Automata Theory and Applications*. Ed. by Adrian-Horia Dediu and Carlos Martín-Vide. Vol. 7183. Lecture Notes in Computer Science. Springer, 2012, pp. 301–312.

[10] Stefan Hetzl and Sebastian Zivota. "Tree Grammars for the Elimination of Non-prenex Cuts". In: *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*. Ed. by Stephan Kreutzer. Vol. 41. LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015, pp. 110–127.