Theorien

Eine Signatur ist eine (endliche oder unendliche) Menge von Prädikaten- und Funktionssymbolen. Wir nehmen an, dass eine Signatur S festgelegt worden ist.

Eine Theorie ist eine Menge von Formeln T (über S) für die gilt: wenn $F_1, \ldots, F_n \in T$ und $\{F_1, \ldots, F_n\} \models G$, dann $G \in T$.

Fakt: Sei $\mathcal{A}$ eine Struktur, die zu S passt. Die Menge der Formeln F mit $\mathcal{A}(F) = 1$ bildet eine modelltheoretisch-definierte Theorie.

Fakt: Sei $\mathcal{F}$ eine entscheidbare Formelmenge (eine Axiomenmenge). Die Menge der Formeln F mit $\mathcal{F} \models F$ bildet eine axiomatisch-definierte Theorie.

Beispiele

Modelltheoretisch-definierte Theorien:

- Arithmetik: $Th(\mathbb{N}, 0, 1, +, \cdot, <)$.
- Presburger Arithmetik: $Th(\mathbb{N}, 0, 1, +, <)$.
- Lineare Arithmetik: $Th(\mathbb{Q}, 0, 1, +, c \cdot (c \in \mathbb{Q}), <)$.

Axiomatisch-definierte Theorien:

- Theorie der Gruppen, Ringe, Körper, booleschen Algebren ...
- Abstrakte Datentypen: Keller, Warteschlangen, ...

Entscheidbarkeit, Axiomatisierbarkeit

Eine Theorie T ist entscheidbar wenn es einen Algorithmus gibt, der für jede Aussage F über S entscheidet, ob $F \in T$.

Eine Theorie T ist axiomatisierbar wenn es eine entscheidbare Formelmenge $\mathcal{F} \subseteq T$ gibt, aus der alle Formeln aus T folgen.

Quantorenelimination

Eine Quantorenelimination-Prozedur (QE-Prozedur) für eine modelltheoretisch-definierte Theorie T ist eine berechenbare Funktion, die eine Formel $\exists x F$ (F quantorenfrei) auf eine quantorenfreie Formel G abbildet mit:

- $\mathcal{A}(\exists x F) = \mathcal{A}(G)$.
- Die freien Variablen von G sind auch freie Variablen von $\exists x F$.

Notation: mit $F_1 \equiv_{\mathcal{A}} F_2$ bezeichnen wir, dass $\mathcal{A}(F_1) = \mathcal{A}(F_2)$ gilt.

Satz: Wenn alle quantorenfreie Aussagen einer Theorie entscheidbar sind und die Theorie eine QE-Prozedur hat, dann ist die Theorie entscheidbar.

Beweis:

- Bringe die Formel in Pränexform.
- Eliminiere die Quantoren von innen nach außen, wobei Allquantoren mit Hilfe der Regel $\forall F \equiv \neg\exists\neg F$ in Existenzquantoren umgewandelt werden.
- Entscheide die resultierende boolesche Kombination von quantorenfreien Aussagen.

Lineare Arithmetik

Lineare Arithmetik: $Th(\mathbb{Q}, 0, 1, +, c \cdot (c \in \mathbb{Q}), <)$

Syntax:

- Terme: $t := 0 \mid 1 \mid t_1 + t_2 \mid c \cdot t$
- Atome: $A := t_1 < t_2 \mid t_1 = t_2$
- Formeln: $F := A \mid \neg F \mid F_1 \vee F_2 \mid F_1 \wedge F_2 \mid \exists F \mid \forall F$

Struktur $\mathcal{A}$:

- Universum: $\mathbb{Q}$
- Interpretation von 0, 1, $+$, $<$ ist klar.
- $\mathcal{A}(c \cdot t) = c \cdot \mathcal{A}(t)$.

Mächtigkeit

Einige Aussagen, die sich in der linearen Arithmetik ausdrücken lassen:

- Das Ungleichungssystem $Ax \leq b$ hat keine Lösung.
- Jede Lösung von $A_1x \leq b_1$ ist auch eine Lösung von $A_2x \leq b_2$.
- Für jede Lösung x_1 von $A_1x \leq b_1$ gibt es Lösungen x_2 bzw. x_3 von $A_2x \leq b_2$ bzw. $A_3x \leq b_3$ mit $x_1 = x_2 + x_3$.
- Die kleinste Lösung von $A_1x \leq b_1$ ist größer als die größte Lösung von $A_2x \leq b_2$.

Fourier-Motzkin Elimination

(Folgende Folien aus Notizen von Prof. Nipkow.)

Eine QE-Prozedur für die lineare Arithmetik.

Gegeben: Formel $\exists x F$ mit F quantorenfrei.
Ziel: Quantorenfreie Formel G mit $G \equiv_{\mathcal{A}} \exists x F$.

Zwei Phasen:

- Phase I: Vereinfachung des Problems durch logische Manipulationen.
- Phase II: QE für den vereinfachten Fall.

Phase I

Schritt 1: Bringe die Negationen nach innen und eliminiere sie mit

$$\begin{aligned} \neg(t_1 = t_2) &\equiv_{\mathcal{A}} (t_2 < t_1) \vee (t_1 < t_2) \\ \neg(t_1 < t_2) &\equiv_{\mathcal{A}} (t_2 < t_1) \vee (t_2 = t_1) \end{aligned}$$

Schritt 2: Wandle in DNF um und schiebe $\exists x$ durch $\vee$ nach innen mit

$$\exists x(F_1 \vee F_2) \equiv \exists x F_1 \vee \exists x F_2$$

Das Ergebnis hat die Form $\bigvee_{i=1}^{n} \exists x(\bigwedge_{j=1}^{m_i} A_{ij})$. Damit können wir uns o.B.d.A. auf den Fall

$$F = A_1 \wedge \ldots \wedge A_n$$

beschränken.

Phase I (Fort.)

Schritt 3: Miniscoping. Betrachte nur die A_i, die x enthalten. Mit der Regel

$$\exists x(A_1 \wedge A_2) \equiv (\exists x A_1) \wedge A_2 \quad \text{wenn } x \text{ in } A_2 \text{ nicht frei vorkommt.}$$

können wir uns o.B.d.A. auf den Fall

$$F = A_1 \wedge \ldots \wedge A_n \quad \text{und } x \text{ kommt in jedem } A_i \text{ frei vor}$$

beschränken.

Phase I (Fort.)

Schritt 4: Isoliere x in A_i.

Definiere x-Atome: $A^x := \; x = t \mid x < t \mid t < x.$

Fakt: Für jedes $i \in [1..n]$ gibt es ein x-Atom A_i^x mit $A_i^x \equiv_{\mathcal{A}} A_i$.
(Linearität notwendig!!)

Beispiel:

$$\begin{array}{rrcl} \text{Wenn} & A_i & = & 3 \cdot x + 5 \cdot y < 7 \cdot x + 3 \cdot z \\ \text{dann nehme} & A_i^x & = & \frac{5}{4} \cdot y + \left(-\frac{3}{4}\right) \cdot z < x \end{array}$$

O.B.d.A. können wir uns auf den Fall

$$F = A_1^x \wedge \ldots \wedge A_n^x$$

beschränken.

Phase II

Fall 1. Es gibt ein $k \in [1..n]$ mit $A_k^x = (x = t_k)$.

Dann gilt: $\exists x F \equiv_{\mathcal{A}} F[x/t_k]$.

Setze $G := F[x/t_k] = A_1^x[x/t_k] \wedge \ldots \wedge A_n^x[x/t_k]$.

Fall 2. Für alle $k \in [1..n]$ gilt $A_k^x = (x < t_k)$ oder $A_k^x = (t_k < x)$.

Klassifiziere die A_i^x:

$$F = \bigwedge_{i=1}^{l} L_i \wedge \bigwedge_{j=1}^{u} U_j \quad \text{mit } L_i = (l_i < x) \text{ und } U_j = (x < u_j)$$

D.h., l_i ist eine untere Schranke (lower bound) und u_j eine obere Schranke (upper bound) für x.

Phase II (Fort.)

Fall 2a: $l = 0$ oder $u = 0$. (Nur obere oder untere Schranken.)

Dann gilt: $\exists x F \equiv_{\mathcal{A}} 1$.

Setze $G := 1$

Fall 2b: $l > 0$ und $u > 0$. (Sowohl obere wie auch untere Schranken.)

Dann gilt: $\exists x F \equiv_{\mathcal{A}} \bigwedge_{i=1}^{l} \bigwedge_{j=1}^{u} (l_i < u_j)$.
($\exists X F$ gilt gdw. alle untere Schranken kleiner als alle obere Schranken. Nur wahr weil $\mathbb{Q}$ dicht geordnet!)

Setze $G = \bigwedge_{i=1}^{l} \bigwedge_{j=1}^{u} (l_i < u_j)$.

Komplexität

Dominiert durch den Fall 2b.

Wenn $|F| = O(n)$ dann gilt $|G| = O(n^2)$.

Die Prozedur braucht damit $O(n^{2^m})$ Zeit für eine Formel $\exists x_1 \ldots \exists x_m F$ der Länge n.
(Angenommen F ist in DNF.)