Parametrisches Integer-Solving[1]

Aless Lasaruk[2]

14. Juni 2005

[1]Eingereicht als Diplomarbeit am Lehrstuhl für Mathematik, Prof. Dr. V. Weispfenning
[2]lasaruk@fmi.uni-passau.de

Zusammenfassung

Ein Quantoreneliminationsverfahren ist ein Algorithmus, der einer Formel eine äquivalente quantorenfreie Formel zuordnet. In dieser Arbeit wird ein neues effektives Quantoreneliminationsverfahren für linear quantifiziere Formeln der uniformen Presburger-Arithmetik vorgestellt. Als linear quantifiziert werden in dieser Arbeit solche Formeln verstanden, deren gebundene Variablen gewissen Liniaritätsbedingungen unterliegen. Das Quantoreneliminationsverfahren wird als Werkzeug zum Lösen von ganzzahligen parametrischen Problemen eingesetzt. Die Anwendbarkeit des Verfahrens für praktische Aufgabenstellungen wird anhand einer Auswahl von Problemen aus unterschiedlichen Bereichen der Mathematik und Informatik untersucht.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Diese Arbeit beschreibt ein neues Quantoreneliminationsverfahren für linear quantifizierte Formeln der uniformen Presburger-Arithmetik. Weiter wird eine Auswahl von Techniken beschrieben, die die Anwendbarkeit dieses Verfahrens zum Lösen solcher parametrischer Probleme verbessern, die sich als linear quantifizierte Formeln der uniformen Presburger-Arithmetik formulieren lassen. Alle Ansätze, die in dieser Arbeit beschrieben wurden, sind im Computeralgebra-System REDUCE als Teil des Pakets REDLOG implementiert. Diese Arbeit ist sowohl für theoretisch interessierte Leser als auch für die Anwender der implementierten Software geeignet. Genauere Lektüre dieser Arbeit erfordert elementare Kenntnisse der universellen Algebra und des mathematischen Beweisens. Speziell für nicht mathematisch versierte Leser gibt es zahlreiche Anmerkungen zur Umsetzung der beschriebenen Methoden. Am Ende eines jeden Kapitels werden die Ergebnisse des jeweiligen Kapitels sowie noch offene Problemstellungen zusammengefaßt und nochmals intuitiv veranschaulicht.

Presburger-Arithmetik stellt eine Theorie erster Stufe der ganzen Zahlen mit Addition und Ordnung dar. 1929 gab Presburger ein Axiomensystem für diese Theorie an und zeigte neben der Vollständigkeit und Konsistenz dieses Axiomensystems, daß es ein Entscheidungsverfahren für beliebige Ausdrücke der Presburger-Arithmetik gibt [Pre29]. Im Jahr 1974 gelang es Fischer und Rabin zu zeigen, daß jeder Entscheidungsalgorithmus für die Presburger-Arithmetik von der Komplexität $2^{2^{cn}}$ ist, wobei c eine Konstante und n die Bitlänge der Eingabeformel ist [FR74]. Neben der Existenz eines Entscheidungsverfahrens zeigte Presburger in [Pre29] die Existenz eines Quantoreneliminationsverfahrens für die Presburger-Arithmetik unter der Voraussetzung, daß man Kongruenzrelationen mit einem festen Modulus zusätzlich in die Sprache aufnimmt. Im Folgenden wird unter Presburger-Arithmetik die Struktur über der um die Kongruenzen erweiterten Sprache verstanden. 1988 gab Weispfenning ein Quantoreneliminationsverfahren für die Presburger-Arithmetik an, welches im Vergleich zum Originalverfahren von Presburger eine moderatere Komplexität hatte und zeigte, daß jede echte obere Schranke für die Worst-Case-Komplexität der Quantorenelimination für Formeln der Presburger-Arithmetik inhärent dreifach exponentiell ist [Wei88]. 1997 zeigte Weispfenning, daß das von ihm in [Wei88] angegebene Quantoreneliminationsverfahren für die Presburger-Arithmetik uniform auf den parametrischen Fall mit nichtquantifizierten skalaren multiplikativen Parametern erweitert werden kann [Wei97]. Neben dieser Tatsache zeigte Weispfenning in dieser Arbeit, daß durch die Einführung von gebundenen Quantoren die Komplexität um eine Exponentialstufe reduziert werden kann.

Eine erste Implementierung des Verfahrens für den nicht uniformen Fall erfolgte 1991 im Rahmen einer Diplomarbeit von Köppl [Koe91]. Eine weitere Implementierung in REDUCE[1] als Teil des Logik-Systems REDLOG[2] folgte dann in den Jahren 2002-2004. Diese Implementierung war im

[1] `http://www.zib.de/Symbolik/reduce/`
[2] `http://www.fmi.uni-passau.de/~redlog/`

Vergleich zur Arbeit von Köppl deutlich effizienter. Das kann durch entsprechende Testbeispiele belegt werden. So lieferte die Implementierung von Köppl für das parametrische Problem

$$\exists x \exists y \exists z (0 < x \wedge 0 < y \wedge 0 < z \wedge x + y + z < c),$$

welches als die Suche nach einem minimalen Wert der Funktion $x + y + z$ mit Nebenbedingungen $x > 0$, $y > 0$ und $z > 0$ interpretiert werden kann, bestenfalls die Lösung

$$\bigvee_{(0 \leq k_3 \leq 2)} \bigvee_{(0 \leq k_2 \leq 1)} \bigvee_{(0 \leq k_1 \leq 1)} (-c + k_1 + k_2 + k_3 < 0 \wedge k_1 > 0 \wedge k_2 > 0 \wedge k_3 > 0).$$

Die Implementierung in REDLOG lieferte bereits zum Beginn dieser Arbeit für das obige Problem das Ergebnis $c > 3$. Diese Steigerung ist unter anderem auf die Simplifikationsalgorithmen für die Presburger-Arithmetik auf der Basis des Smart-Simplifiers von REDLOG zurückzuführen [DS97a],[DS97b]. Das Verfahren in dieser Arbeit liefert das Ergebnis $c > 3$ *ohne Verwendung* des Smart-Simplifiers durch geschickte Eliminationstechnik.

Im ersten Kapitel wird die uniforme Presburger-Arithmetik formal eingeführt. Dies erscheint aufgrund des feinen Zusammenspiels zwischen Syntax und Semantik bei der Einführung von gebundenen Quantoren für das Verständnis der Arbeit als unerläßlich. Im zweiten Abschnitt des ersten Kapitels wird gezeigt, wie einige parametrische Probleme in Termini der Presburger-Arithmetik formuliert werden können. Im zweiten Kapitel werden Simplifikationsalgorithmen für uniforme Presburger-Formeln vorgestellt. Im dritten Kapitel wird ein zu [Wei88] bzw. [Wei97] ähnliches Quantoreneliminationsverfahren als eine Elimination durch virtuelle Substitution von Testpunkten dargestellt und um die Konzepte der strukturellen Eliminationsmengen, Condensing und erweiterter Quantorenelimination erweitert. Im vierten Kapitel folgt eine kurze Beschreibung der Implementierung und der Benutzerschnittstelle sowie eine praktische Untersuchung des Laufzeitverhaltens der Quantorenelimination mit Hilfe der vorgestellten Erweiterungen. Als Beispiele dienen unter anderem die im ersten Kapitel angegebenen Problemstellungen. Anschließend werden im fünften Kapitel die Ergebnisse der Arbeit zusammengefaßt.

1.1 Uniforme Presburger-Arithmetik

In diesem Abschnitt wird die uniforme Presburger-Arithmetik formal eingeführt. Im Folgenden wird unter Presburger-Arithmetik, falls nichts anderes explizit angegeben wird, stets die uniforme Presburger-Arithmetik verstanden. Man unterscheidet grundsätzlich zwei Betrachtungsweisen: die *Syntax* und die *Semantik*. Die syntaktische Basis für die folgenden Ausführungen bilden Presburger- *Sprache*, *Terme* und *Formeln*. Diese werden auf der semantischen Seite in der *Struktur* der Presburger-Arithmetik interpretiert.

1.1.1 Syntaktische Vereinbarungen

Im Gegensatz zum Ansatz in [Wei97] wird die uniforme Presburger Arithmetik in dieser Arbeit durch die Sprache der Ringe mit Ordnung, erweitert um die Symbole für Kongruenzen und Inkongruenzen, dargestellt. Die *Sprache* der Presburger-Arithmetik besteht somit aus den endlichen Mengen von *Funktionszeichen* $\{0^{(0)}, 1^{(1)}, -^{(1)}, +^{(2)}, \cdot^{(2)}\}$ und *Relationszeichen* $\{\neq^{(2)}, <^{(2)}, >^{(2)}, \leq^{(2)}, \geq^{(2)}, \cong^{(3)}, \ncong^{(3)}\}$. Die Zahl in Klammern gibt dabei die Stelligkeit an. Weiterhin sei

$$\mathcal{X} = \{=, (,), \exists, \forall, \wedge, \vee, \neg, \bigwedge, \bigvee, \text{true}, \text{false}\}$$

die Menge von *Sonderzeichen* und $\mathcal{V}$ eine unendliche Menge von *Variablen*, sodaß alle obigen Mengen paarweise disjunkt sind. Mit Hilfe des somit erhaltenen Alphabets

$$\mathcal{Z} = \mathcal{X} \cup \mathcal{V} \cup \{0, 1, -, +, \cdot, \neq, <, >, \leq, \geq, \cong, \ncong\}$$

werden *Terme*, *atomare Formeln* und *Formeln* der Presburger-Arithmetik als Elemente von $\mathcal{Z}^*$ wie folgt definiert.

Definition 1.1.1 (Presburger-Terme) Die Menge der *Presburger-Terme* $\mathcal{T}$ ist die kleinste Teilmenge von $\mathcal{Z}^*$ mit den Eigenschaften

(i) $\mathcal{V} \subseteq \mathcal{T}$, $\{1,0\} \subseteq \mathcal{T}$,

(ii) falls $t \in \mathcal{T}$, so ist auch $-(t) \in \mathcal{T}$ und

(iii) falls $t_1, t_2 \in \mathcal{T}$, so ist auch $+(t_1,t_2), \cdot(t_1,t_2) \in \mathcal{T}$.

Um bessere Lesbarkeit zu erzielen, werden bestimmte Terme abgekürzt geschrieben. Neben der üblichen infixen Schreibweise für zweistellige Terme der Form $+(t_1,t_2)$ und $\cdot\,(t_1,t_2)$ werden Terme der Form

$$\underbrace{1+(1+\ldots(1+1))}_{z\text{-mal}} \text{ falls } z>0 \text{ und } \quad \underbrace{-1+(-1+\ldots(-1+(-1)))}_{-z\text{-mal}} \text{ falls } z<0$$

für $z \in \mathbb{Z}\setminus\{0,1\}$ mit z abgekürzt. Da sich somit jeder Zahl $z \in \mathbb{Z}$ ein Presburger-Term t zuordnen läßt, werden die Zahl z und der entsprechende Term t identifiziert. Neben dem Weglassen des „$\cdot$" Zeichens werden, wie üblich, die Ausdrücke der Form $\cdot(t,\cdot(t,\ldots\cdot(t,t)))$ mit $t \in \mathcal{T}$ als t^k für ein geeignetes $k \in \mathbb{N}$ geschrieben und als Potenzen von t bezeichnet. Bei der infixen Darstellung der Terme werden übliche Präzedenzregeln verwendet. So wird zum Beispiel der Term $+(y,+(x,x))$ als $y+x+x$ bzw. als $y+2x$ geschrieben.

Für jedes $t \in \mathcal{T}$ bezeichnet $\mathcal{V}(t)$ die Menge der in t vorkommenden Variablen. Das Tupel der Form $(t,(x_1,\ldots,x_n))$ mit paarweise verschiedenen Variablen x_i mit der Eigenschaft $\mathcal{V}(t) \subseteq \{x_1,\ldots,x_n\}$ bezeichnet man als *erweiterten Term*. Im Folgenden wird ein erweiterter Term $(t,(x_1,...,x_n))$ auch mit $t(x_1,\ldots,x_n)$ abgekürzt. Weiterhin soll bei der Angabe von Variablen durch *verschiedene* Metasymbole stets gemeint sein, daß auch die Variablen, die durch diese Symbole repräsentiert werden, verschieden sind.

Definition 1.1.2 (Atomare Presburger-Formeln) Seien $t_1,t_2,m \in \mathcal{T}$ Presburger-Terme. Eine *atomare Presburger-Formel* ist entweder eine *Gleichung* $=(t_1,t_2)$ oder ein *Prädikat* von der Form $\rho(t_1,t_2)$ für $\rho \in \{\neq,<,>,\leq,\geq\}$ bzw. $\rho(t_1,t_2,m)$ für $\rho \in \{\cong,\ncong\}$. Die Menge aller atomaren Formeln wird mit $\mathcal{A}$ bezeichnet.

Man nennt eine atomare Formel $\varphi=\rho(t_1,t_2)$ eine *Disgleichung*, falls ρ das Symbol „$\neq$" ist. Falls $\rho \in \{<,>,\leq,\geq\}$, so bezeichnet man die atomare Formel φ als eine *Ungleichung*. Falls ρ für eine atomare Formel $\varphi=\rho(t_1,t_2,m)$ das Symbol „ $\cong''$ bzw. „ $\ncong''$ ist, so bezeichnet man φ als eine *Kongruenz* bzw. *Inkongruenz*. Gleichungen und atomare Formeln mit zweistelligen Relationen $\rho(t_1,t_2)$ werden durch $t_1\ \rho\ t_2$ und (In-)Kongruenzen der Form $\rho(t_1,t_2,m)$ durch $t_1\ \rho_m\ t_2$ abgekürzt. Der Term m heißt dann der *Modulus* der (In-)Kongruenz.

Ausgehend von der Definition von $\mathcal{V}(t)$ für einen Term $t \in \mathcal{T}$ definiert man die *Menge der Variablen* $\mathcal{V}(\varphi)$ einer atomaren Formel $\varphi \in \mathcal{A}$. Dabei gilt für eine atomare Formel φ der Form $t_1\ \rho\ t_2$ bzw. $t_1=t_2$ für die Menge der vorkommenden Variablen $\mathcal{V}(\varphi)=\mathcal{V}(t_1)\cup\mathcal{V}(t_2)$. Für (In-)Kongruenzen gilt $\mathcal{V}(\rho(t_1,t_2,m))=\mathcal{V}(t_1)\cup\mathcal{V}(t_2)\cup\mathcal{V}(m)$. Analog zur Definition von erweiterten Termen erhält man für eine atomare Formel φ eine *erweiterte atomare Formel* $\varphi(x_1,\ldots,x_n)$.

Definition 1.1.3 (Presburger-Formeln) Die Menge der *Presburger-Formeln* $\mathcal{F}$ wird definiert als die kleinste Teilmenge von $\mathcal{Z}^*$ mit folgenden Eigenschaften.

(i) true $\in \mathcal{F}$, false $\in \mathcal{F}$ und $\mathcal{A} \subseteq \mathcal{F}$.

(ii) Falls $\varphi \in \mathcal{F}$, so ist auch $\neg\varphi \in \mathcal{F}$.

(iii) Falls φ_1, $\varphi_2 \in \mathcal{F}$, so sind auch $(\varphi_1 \vee \varphi_2) \in \mathcal{F}$ und $(\varphi_1 \wedge \varphi_2) \in \mathcal{F}$.

(iv) Falls $\varphi \in \mathcal{F}$ und $x \in \mathcal{V}$, so sind auch $(\exists x\varphi) \in \mathcal{F}$ und $(\forall x\varphi) \in \mathcal{F}$. Die Formel φ heißt dann der *Bereich* von $(\exists x\varphi)$ bzw. $(\forall x\varphi)$.

Eine Formel, in der die Symbole $\exists$ bzw. $\forall$ nicht vorkommen, heißt *schwach quantorenfrei*. Die Menge aller atomaren Formeln, die in einer Formel φ vorkommen, wird mit $\mathrm{at}(\varphi)$ bezeichnet.

Es soll weiterhin gelten, daß das Symbol „$\neg$" am stärksten und das Symbol „$\vee$" am schwächsten unter den Symbolen „$\neg$", „$\vee$" und „$\wedge$" bindet. Dadurch lassen sich redundante Klammerungen, die zum Beispiel bei verschachtelten Anwendungen ein und der selben Regel entstehen, bei der Darstellung der Formel vermeiden. Die äußeren Klammern einer Formel werden ebenfalls nach Möglichkeit bei der Darstellung dieser vermieden. So schreibt man zum Beispiel für $(\varphi_1 \wedge (\varphi_2 \wedge (\varphi_3 \vee \varphi_4)))$ einfach $\varphi_1 \wedge \varphi_2 \wedge (\varphi_3 \vee \varphi_4)$. Die Abkürzungen $\varphi_1 \longrightarrow \varphi_2$, $\varphi_1 \longleftarrow \varphi_2$ und $\varphi_1 \longleftrightarrow \varphi_2$ stehen für $\neg\varphi_1 \vee \varphi_2$, $\varphi_1 \vee \neg\varphi_2$ bzw. $(\neg\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_2)$. Konjunktionen bzw. Disjunktionen der Form

$$\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n \quad \text{bzw.} \quad \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n$$

werden im Folgenden, wie üblich, durch $\bigwedge_{i\in I} \varphi_i$ bzw. durch $\bigvee_{i\in I} \varphi_i$ mit geeigneter Indexmenge I abgekürzt. Dabei ist zu beachten, daß die Abkürzungen metasprachlich sind und *nicht* in die Menge der Formeln aufgenommen werden. Speziell kommt das Zeichen „$\bigwedge$" bzw. „$\bigvee$" streng formal in obigen Zeichenketten *nicht* vor.

Jede Formel, die während der Bildung von $\varphi \in \mathcal{F}$ durch die Anwendung der Regeln (i)-(iv) entsteht, heißt eine *Teilformel* von φ. Die maximale Anzahl von *verschachtelten* Anwendungen der Regeln (ii)-(iv) zur Bildung von φ wird als *maximale Tiefe* von φ bezeichnet. So ist die Formel

$$\varphi = \exists x(x + a^2 < 0 \vee (x > 0 \wedge 2x \cong_a 0))$$

von maximaler Tiefe 3. Mit $\mathcal{V}(\varphi)$ wird die Menge aller in $\varphi \in \mathcal{F}$ vorkommenden Variablen bezeichnet. Ein Vorkommen von x im Bereich einer Formel φ der Form $\exists x\psi$ bzw. $\forall x\psi$ heißt ein *gebundenes Vorkommen* von x in φ und die Variable x gebunden in φ. Ein Vorkommen von x in einer Formel φ nicht im Bereich von $\exists x\psi$ bzw. $\forall x\psi$ heißt ein *freies Vorkommen*[3]. Mit $\mathcal{V}_g(\varphi)$ bzw. $\mathcal{V}_f(\varphi)$ wird für eine Formel $\varphi \in \mathcal{F}$ die Menge aller Variablen von φ bezeichnet, die ein gebundenes bzw. ein freies Vorkommen in φ haben. So gilt zum Beispiel $\mathcal{V}(\varphi) = \{a, b, c, m, y, z\}$, $\mathcal{V}_f(\varphi) = \{a, b, c, m, z\}$ und $\mathcal{V}_g(\varphi) = \{y, z\}$ für

$$\varphi = a^2 + (z + 1)c \cong_m 0 \vee \exists y(a^3 < y \wedge \exists z(b^3 > z)).$$

Eine *erweiterte Formel* ist ein Tupel $(\varphi, (x_1, \ldots, x_n))$, wobei $\varphi \in \mathcal{F}$, x_i paarweise verschieden sind und $\mathcal{V}_f(\varphi) \subseteq \{x_1, \ldots, x_n\}$. Es wird im Folgenden stillschweigend vorausgesetzt, daß $\mathcal{V}_f(\varphi) \cap \mathcal{V}_g(\varphi) = \emptyset$ für jede Formel $\varphi \in \mathcal{F}$ gilt. In der Praxis wird dies stets mit Hilfe einer geeigneten Umbenennung der gebundenen Variablen erreicht, sodaß weder die semantische Interpretation der Formel noch die Erweiterung verändert wird.

Definition 1.1.4 (Substitution) Sei $\varphi \in \mathcal{F}$, $v \in \mathcal{V}$ und $t \in \mathcal{T}$. Dann wird die Formel, die aus φ dadurch entsteht, daß man jedes freie Vorkommen der Variable v durch t ersetzt, mit

$$\varphi[t/v]$$

abgekürzt. Man sagt dabei, daß $\varphi[t/v]$ aus φ durch *Substitution* von v durch t entsteht.

An dieser Stelle sollte man sich daran erinnern, daß jedes $z \in \mathbb{Z}$ mit einem Presburger-Term $1+1+\ldots+1$, $(-1)+(-1)+\ldots+(-1)$ oder 0 identifiziert ist. Somit wird im Folgenden für $z \in \mathbb{Z}$ eine Formel φ und eine Variable v auch der Ausdruck $\varphi[z/v]$ als gültig betrachtet. Analog ist die Substitution $t[t'/v]$ von t' durch v in einen Term t definiert. In dieser Arbeit wird die Schreibweise

$$\varphi[t_1/v_1, ..., t_n/v_n]$$

nur für $t_i \in \mathbb{Z} \cup (\mathcal{V} \setminus \{v_1, ..., v_n\})$ für jedes $1 \leq i \leq n$ verwendet.

[3]Die Zeichenkette $\exists x$ bzw. $\forall x$ wird *nicht* als ein Vorkommen von x betrachtet.

1.1.2 Semantik der Presburger-Arithmetik

Die im letzten Abschnitt vorgestellten Objekte besitzen einen rein syntaktischen Charakter. Durch die Einführung der *Struktur* der Presburger-Arithmetik $\mathbf{P}$ erhält man semantische Interpretation dieser Objekte. Jedem Funktionszeichen f mit der Stelligkeit k wird dabei eine Funktion $f^{\mathbf{P}} : \mathbb{Z}^k \to \mathbb{Z}$ und jeder Relation $\rho^{(k)}$ eine Funktion $\rho^{\mathbf{P}} : \mathbb{Z}^k \to \{\bot, \top\}$ zugeordnet. Das Symbol „$\bot$" steht für *falsch* und „$\top$" für *richtig*. Die Symbole „$\bot$" und „$\top$" sind *keine* syntaktischen Objekte und dürfen nicht mit den Formeln true und false verwechselt werden.

Definition 1.1.5 (Presburger-Arithmetik) Die Struktur der Presburger-Arithmetik, kurz die *Presburger-Arithmetik*, $\mathbf{P}$ ordnet den Funktions- und Relationszeichen aus $\{0, 1, -, +, \cdot, \neq, <, >, \leq, \geq\}$ die gewöhnlichen Interpretationen von 0, 1, $-$, $+$, $\cdot$, $\neq$, $<$, $>$, $\leq$ und $\geq$ auf der Menge der ganzen Zahlen zu. Für die Relationszeichen aus $\{\cong, \ncong\}$ und für $a, b, m \in \mathbb{Z}$ gilt $\cong^{\mathbf{P}}(a, b, m) = \top$ genau dann, wenn für ein $k \in \mathbb{Z}$ gilt $a + km = b$ bzw. $\ncong^{\mathbf{P}}(a, b, m) = \top$ genau dann, wenn für kein $k \in \mathbb{Z}$ gilt $a + km = b$.

Obige Definition verallgemeinert den gewöhnlichen Begriff einer „Kongruenz".

Lemma 1.1.6 (Eigenschaften der Kongruenzrelationen) Seien $a, b, m \in \mathbb{Z}$.

(i) $\cong^{\mathbf{P}}(a, b, 0) = \top$ bzw. $\ncong^{\mathbf{P}}(a, b, 0) = \top$ genau dann, wenn $a = b$ bzw. $a \neq b$.

(ii) $\cong^{\mathbf{P}}(a, b, m) = \top$ bzw. $\ncong^{\mathbf{P}}(a, b, m) = \top$ genau dann, wenn

$$\cong^{\mathbf{P}}(a, b, -m) = \top \quad \text{bzw.} \quad \ncong^{\mathbf{P}}(a, b, -m) = \top.$$

(iii) $\cong^{\mathbf{P}}(a, b, m) = \top$ bzw. $\ncong^{\mathbf{P}}(a, b, m) = \top$ genau dann, wenn $m \mid a - b$ bzw. $m \nmid a - b$.

Die Interpretation der Funktionszeichen läßt sich auf die Menge der Terme erweitern. Jedem erweiterten Term $t(x_1, \ldots, x_n)$ wird entsprechend der Definition von Termen induktiv eine Termfunktion $t^{\mathbf{P}} : \mathbb{Z}^n \to \mathbb{Z}$ zugeordnet, welche jeder *Belegung* $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{Z}^n$ des Variablentupels $(x_1, \ldots, x_n)$ mit Werten aus $\mathbb{Z}$ die Auswertung bzw. Interpretation des Terms in $\mathbb{Z}$ zuordnet. Die somit erhaltene Interpretation der Presburger-Terme ist mit der der multivariaten ganzzahligen Polynomausdrücke identisch.

Definition 1.1.7 (Interpretation erweiterter atomarer Formeln) Sei eine erweiterte atomare Presburger-Formel $\varphi(x_1, \ldots, x_n)$ gegeben. Dann ist die *Interpretation* von $\varphi(x_1, \ldots, x_n)$ für $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{Z}^n$ durch die Funktion $\varphi^{\mathbf{P}} : \mathbb{Z}^n \to \{\bot, \top\}$ definiert mit folgenden Eigenschaften.

(i) Falls φ eine Gleichung $t_1 = t_2$ ist, so gilt $\varphi^{\mathbf{P}}(\mathbf{z}) = \top$ genau dann, wenn $t_1^{\mathbf{P}}(\mathbf{z}) = t_2^{\mathbf{P}}(\mathbf{z})$.

(ii) Falls φ ein Prädikat $\rho(t_1, t_2)$ bzw. $\rho(t_1, t_2, m)$ ist, so gilt $\varphi^{\mathbf{P}}(\mathbf{z}) = \top$ genau dann, wenn

$$\rho^{\mathbf{P}}(t_1^{\mathbf{P}}(\mathbf{z}), t_2^{\mathbf{P}}(\mathbf{z})) = \top \quad \text{bzw.} \quad \rho^{\mathbf{P}}(t_1^{\mathbf{P}}(\mathbf{z}), t_2^{\mathbf{P}}(\mathbf{z}), m^{\mathbf{P}}(\mathbf{z})) = \top.$$

Darauf aufbauend erhält man die Interpretation von erweiterten Formeln. Betrachte dafür die Ordnung $\preceq$ auf der Menge $\{\bot, \top\}$, mit $\bot \preceq \top$. Ferner sei die Funktion not : $\{\bot, \top\} \to \{\bot, \top\}$ definiert durch

$$\text{not}(x) = \begin{cases} \top & \text{falls } x = \bot \text{ und} \\ \bot & \text{sonst.} \end{cases}$$

Dann ergibt sich die Interpretation einer erweiterten Formel $\varphi(x_1, \ldots, x_n)$ durch die Funktion $\varphi^{\mathbf{P}} : \mathbb{Z}^n \to \{\bot, \top\}$, wie die folgende Definition zeigt.

Definition 1.1.8 (Interpretation erweiterter Presburger-Formeln) Sei $\varphi(x_1, \ldots, x_n)$ eine erweiterte Presburger-Formel. Dann ist die *Interpretation* von $\varphi(x_1, \ldots, x_n)$ für $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{Z}^n$ durch die Funktion $\varphi^{\mathbf{P}} : \mathbb{Z}^n \to \{\bot, \top\}$ definiert mit folgenden Eigenschaften.

(i) Falls $\varphi = \text{false}$, so ist $\varphi^{\mathbf{P}}(\mathbf{z}) = \bot$. Falls $\varphi = \text{true}$, so ist $\varphi^{\mathbf{P}}(\mathbf{z}) = \top$.

(ii) Falls φ atomar ist, betrachte die Definition 1.1.7.

(iii) Falls φ von der Form $\neg\varphi_1$ ist, so ist $\varphi^{\mathbf{P}}(\mathbf{z}) = \text{not}(\varphi_1^{\mathbf{P}}(\mathbf{z}))$.

(iv) Falls φ von der Form $(\varphi_1 \vee \varphi_2)$ bzw. $(\varphi_1 \wedge \varphi_2)$ ist, so ist

$$\varphi^{\mathbf{P}}(\mathbf{z}) = \max\{\varphi_1^{\mathbf{P}}(\mathbf{z}), \varphi_2^{\mathbf{P}}(\mathbf{z})\} \quad \text{bzw.} \quad \varphi^{\mathbf{P}}(\mathbf{z}) = \min\{\varphi_1^{\mathbf{P}}(\mathbf{z}), \varphi_2^{\mathbf{P}}(\mathbf{z})\}.$$

(v) Falls φ von der Form $(\exists x\psi)$ bzw. $(\forall x\psi)$ ist, so ist für die erweiterte Formel $\psi(x_1, \ldots, x_n, x)$

$$\varphi^{\mathbf{P}}(\mathbf{z}) = \max\{\psi^{\mathbf{P}}(\mathbf{z}, z) \mid z \in \mathbb{Z}\} \quad \text{bzw.} \quad \varphi^{\mathbf{P}}(\mathbf{z}) = \min\{\psi^{\mathbf{P}}(\mathbf{z}, z) \mid z \in \mathbb{Z}\}.$$

Ist für eine erweiterte Formel $\varphi(x_1, \ldots, x_n)$ und für eine Stelle $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{Z}^n$ die Gleichheit $\varphi^{\mathbf{P}}(z_1, \ldots, z_n) = \top$ erfüllt, so sagt man, daß φ an der Stelle $\mathbf{z}$ gilt. Dies wird mit $\mathbf{P} \models \varphi(\mathbf{z})$ abgekürzt. Die Menge $S(\varphi(x_1, \ldots, x_n))$ aller Stellen, an denen die erweiterte Formel $\varphi(x_1, \ldots, x_n)$ gilt

$$S(\varphi(x_1, \ldots, x_n)) = \{\mathbf{z} \in \mathbb{Z}^n \mid \mathbf{P} \models \varphi(\mathbf{z})\}$$

bezeichnet man als *Erfüllungsmenge* von $\varphi(x_1, \ldots, x_n)$. Analog dazu definiert man für $k \leq n$ die Erfüllungsmenge $S^{\mathbf{x}}_{\mathbf{z}}(\varphi(x_1, \ldots, x_n))$ einer erweiterten Formel $\varphi(x_1, \ldots, x_n)$ bezüglich einer festen Belegung $\mathbf{z} = (z_1, \ldots, z_k)$ des Variablentupels $\mathbf{x} = (x_1, \ldots, x_k)$ als

$$S^{\mathbf{x}}_{\mathbf{z}}(\varphi(x_1, \ldots, x_n)) = \{(z_{k+1}, \ldots, z_n) \in \mathbb{Z}^{n-k} \mid \mathbf{P} \models \varphi(z_1, \ldots, z_k, z_{k+1}, \ldots, z_n)\}.$$

Formeln, die für jede Belegung der vorkommenden Variablen den gleichen Wahrheitsgehalt tragen, werden als *semantisch äquivalent* bezeichnet.

Definition 1.1.9 (Semantische Äquivalenz) Zwei Presburger-Formeln φ und ψ heißen dann *semantisch äquivalent*, wenn gilt

$$\mathbf{P} \models \varphi \longleftrightarrow \psi.$$

Semantische Äquivalenz induziert offenbar eine Äquivalenzrelation auf der Menge der Presburger-Formeln, welche im Folgenden mit $\sim$ bezeichnet wird. Die Eigenschaft zweier Formeln semantisch Äquivalent zu sein hängt außerdem nicht von der Wahl der Erweiterung ab.

1.1.3 Gebundene Quantoren

Eine existenziell quantifizierte Formel der Form $\exists x\varphi$ läßt sich intuitiv als eine unendliche Disjunktion und eine universell quantifizierte Formel der Form $\forall x\varphi$ als eine unendliche Konjunktion von Formeln $\varphi[z/x]$ für jedes $z \in \mathbb{Z}$ auffassen. Ist die Menge der Werte, die substituiert werden müssen, aufgrund der Struktur von φ endlich, so nennt man den Quantor $\exists x$ bzw. $\forall x$ einen *gebundenen Quantor*. Ein Beispiel dafür ist

$$\exists x((0 \leq x \leq 100) \wedge \psi) \quad \text{bzw.} \quad \forall x((0 \leq x \leq 100) \longrightarrow \psi).$$

Gebundene Quantoren werden aufgrund ihrer Bedeutung für die Quantorenelimination als zusätzliche syntaktische Objekte eingeführt. Dabei soll die Auffassung eines gebundenen Quantors als eine endliche Konjunktion bzw. Disjunktion von Formeln dominieren. Eine schwach quantorenfreie Formel φ, in der die Symbole „$\bigvee$" und „$\bigwedge$" nicht vorkommen, heißt *stark quantorenfrei*. Alle bisher definierbaren schwach quantorenfreien Formeln sind stark quantorenfrei.

Definition 1.1.10 (Gebundene Quantoren) Die Definition einer Presburger-Formel wird um den folgenden Fall vervollständigt (vergleiche Definition 1.1.3).

(v) Falls $\varphi \in \mathcal{F}$ und $\psi \in \mathcal{F}$ eine stark quantorenfreie Formel mit etwa $\mathcal{V}(\psi) = \{x_1, \ldots, x_n, v\}$, sodaß die Erfüllungsmenge

$$S_{\mathbf{z}}^{(x_1,\ldots,x_n)}(\psi(x_1, \ldots, x_n, v))$$

für jede Wahl von $\mathbf{z} \in \mathbb{Z}^n$ endlich ist, so sind

$$((\bigvee v\psi)\varphi) \in \mathcal{F} \quad \text{und} \quad ((\bigwedge v\psi)\varphi) \in \mathcal{F}.$$

Eine Zeichenkette der Form $(\bigvee v\psi)$ bzw. $(\bigwedge v\psi)$ heißt ein *gebundener Quantor*. Für eine Formel $\gamma = ((\bigvee v\psi)\varphi)$ bzw. $\gamma = ((\bigwedge v\psi)\varphi)$ heißt die Formel ψ der *Bound* von γ und die Variable v die *Bound-Variable*. Die Variablen $x_1, \ldots, x_n$ heißen die *Bound-Parameter*. Die Formel φ heißt der *Bereich* von γ. Ist φ in obiger Definition schwach quantorenfrei, so ist auch γ eine schwach quantorenfreie Formel.

Obige Notation $((\bigvee v\psi)\varphi)$ bzw. $((\bigwedge v\psi)\varphi)$ erfüllt offenbar die Kriterien *eindeutiger Lesbarkeit* und wird im Folgenden mit

$$\bigvee_{\psi(v)} \varphi \quad \text{bzw.} \quad \bigwedge_{\psi(v)} \varphi$$

abgekürzt. Dabei sind Abkürzungen für verschachtelte Konjunktionen bzw. Disjunktionen von gebundenen Quantoren zu unterscheiden, da erstere streng formal keine Elemente von $\mathcal{F}$ sind. Im Folgenden wird, der sich eingebürgerten Sprechweise folgend, auch $\bigvee_{\psi(v)} \varphi$ bzw. $\bigwedge_{\psi(v)} \varphi$ als gebundener Quantor bezeichnet.

Beispiel 1.1.11 Seien $t_i, l_i \in \mathcal{T}$ für $i \in I$ mit einer endlichen Indexmenge I, sodaß v in t_i und l_i für alle $i \in I$ nicht vorkommt. Dann ist für

$$\psi = \bigvee_{i \in I} l_i \leq v \wedge v \leq t_i$$

und eine Formel $\varphi \in \mathcal{F}$ die Formel $\bigvee_{\psi(v)} \varphi$ bzw. $\bigwedge_{\psi(v)} \varphi$ ein gebundener Quantor. Man beachte dabei, daß die Definition von ψ nicht der Definition eines gebundenen Quantors widerspricht, denn das Symbol „$\bigvee$" kommt in ψ streng formal nicht vor. Formeln der Form $a \leq x \wedge x \leq b$ werden im Folgenden durch $a \leq x \leq b$ abgekürzt.

Die Definition der Menge der *freien Variablen* $\mathcal{V}_f(\varphi)$ bzw. der *gebundenen Variablen* $\mathcal{V}_g(\varphi)$ einer Formel φ kann auf Formeln mit gebundenen Quantoren fortgesetzt werden. Ein Vorkommen einer Variablen x in einer Formel φ im Bereich oder im Bound eines gebundenen Quantors $\bigvee_{\psi(x)}$ bzw. $\bigwedge_{\psi(x)}$ heißt ebenfalls ein gebundenes Vorkommen von x. Analog dazu heißt ein Vorkommen von x frei, wenn x in φ weder durch einen gebundenen noch durch einen normalen Quantor gebunden ist[4]. Ein Tupel von der Form $(\bigvee_{\psi(v)} \varphi, (x_1, \ldots, x_n))$ bzw. $(\bigwedge_{\psi(v)} \varphi, (x_1, \ldots, x_n))$, sodaß $\mathcal{V}_f(\bigvee_{\psi(v)} \varphi) \subseteq \{x_1, \ldots, x_n\}$ bzw. $\mathcal{V}_f(\bigwedge_{\psi(v)} \varphi) \subseteq \{x_1, \ldots, x_n\}$ und die Variablen x_i paarweise verschiedenen sind, heißt ein erweiterter gebundener Quantor.

Definition 1.1.12 (Interpretation gebundener Quantoren) Die Interpretation einer erweiterten Formel (vergleiche Definition 1.1.8) wird vervollständigt durch die folgenden Fälle.

(vi) Falls φ ein gebundener Quantor $((\bigvee v\psi)\gamma)$ ist, so ist für $\gamma(x_1, \ldots, x_n, v)$ und $\psi(x_1, \ldots, x_k, v)$

$$\varphi^{\mathbf{P}}(z_1, \ldots, z_n) = \max\{\gamma[z/v]^{\mathbf{P}}(z_1, \ldots, z_n) \mid z \in S_{(z_1,\ldots,z_k)}^{(x_1,\ldots,x_k)}(\psi(x_1, \ldots, x_k, v))\}.$$

(vii) Falls φ ein gebundener Quantor $((\bigwedge v\psi)\gamma)$ ist, so ist für $\gamma(x_1, \ldots, x_n, v)$ und $\psi(x_1, \ldots, x_k, v)$

$$\varphi^{\mathbf{P}}(z_1, \ldots, z_n) = \min\{\gamma[z/v]^{\mathbf{P}}(z_1, \ldots, z_n) \mid z \in S_{(z_1,\ldots,z_k)}^{(x_1,\ldots,x_k)}(\psi(x_1, \ldots, x_k, v))\}.$$

[4]Die Zeichenkette $\bigvee x$ bzw. $\bigwedge x$ wird *nicht* als ein Vorkommen von x betrachtet

An dieser Stelle wird nochmal daran erinnert, daß die Substitution von Werten $z \in \mathbb{Z}$ in obiger Definition als gültig erklärt wurde. Nach eingeführter Konvention können im Bound eines gebundenen Quantors Bound-Variablen anderer gebundener Quantoren vorkommen.

Beispiel 1.1.13 Man betrachte die Formel

$$\bigwedge_{(0\leq k\leq 10)(k)} \bigwedge_{(0\leq l\leq k)(l)} x > l.$$

Aus Beispiel 1.1.13 folgt nach Interpretation gebundener Quantoren 1.1.12, daß die Reihenfolge der Verschachtelung gleichartiger gebundener Quantoren im Allgemeinen *nicht* verändert werden darf.

Lemma 1.1.14 (Eigenschaften gebundener Quantoren) Es gelten folgende Äquivalenzen.

(i) $\bigvee_{\psi(v)} \varphi \sim \exists v(\psi \wedge \varphi)$.

(ii) $\bigwedge_{\psi(v)} \varphi \sim \forall v(\psi \longrightarrow \varphi)$.

(iii) $\neg \bigvee_{\psi(v)} \varphi \sim \bigwedge_{\psi(v)} \neg\varphi$.

(iv) $\neg \bigwedge_{\psi(v)} \varphi \sim \bigvee_{\psi(v)} \neg\varphi$.

Beweis: (i)-(ii) Die Behauptungen folgen direkt aus dem Vergleich der Interpretationen in Definitionen 1.1.8 und 1.1.12. (iii)-(iv) Die Behauptungen folgen aus logischen Umformungen mit Hilfe von (i) bzw. (ii). □

Für einen Bound ψ eines gebundenen Quantors φ mit $|\mathcal{V}(\psi)| > 1$ hängt die Erfüllungsmenge von ψ von den Bound-Parametern von ψ ab. Somit kann die entsprechende Disjunktion bzw. Konjunktion im Allgemeinen nicht explizit angegeben werden. Für eine feste Wahl der Bound-Parameter, die man zum Beispiel durch Substitution von ganzen Zahlen für die Bound-Parameter erreichen kann sowie für nichtparametrische Bounds kann ein gebundener Quantor jedoch als eine Konjunktion bzw. Disjunktion dargestellt werden.

Definition 1.1.15 (Expansion gebundener Quantoren) Für einen gebundenen Quantor der Form $\bigvee_{\psi(v)} \varphi$ bzw. $\bigwedge_{\psi(v)} \varphi$ mit $\mathcal{V}(\psi) \subseteq \{v\}$ heißt die äquivalente Formel

$$\bigvee_{z\in S(\psi(v))} \varphi[z/v] \quad \text{bzw.} \quad \bigwedge_{z\in S(\psi(v))} \varphi[z/v]$$

eine *Expansion* von $\bigvee_{\psi(v)} \varphi$ bzw. $\bigwedge_{\psi(v)} \varphi$.

Es gilt zum Beispiel die folgende Äquivalenz

$$\bigwedge_{(k=0\vee(3\leq k\leq 5))(k)} k + x = 3 \sim (0 + x = 3 \wedge 3 + x = 3 \wedge 4 + x = 3 \wedge 5 + x = 3).$$

Offenbar ist die Expansion eines gebundenen Quantors mit stark quantorenfreiem Bereich stark quantorenfrei. Es gibt allerdings schwach quantorenfreie Formeln, die keine äquivalenten stark quantorenfreien Form besitzen.

Beispiel 1.1.16 Betrachte die Formel

$$\varphi = \bigvee_{(0\leq 2k\wedge 2k\leq a)(k)} x \cong_a k.$$

Es gilt nach Lemma 1.1.14(i)

$$\varphi \sim \exists k(0 \leq 2k \wedge 2k \leq a \wedge x \cong_a k).$$

Die rechte Seite der letzten Äquivalenz besitzt nach Theorem 3.2 in [Wei97] keine äquivalenten stark quantorenfreien Formel.

1.1.4 Linear quantifizierte Formeln und Quantorenelimination

Der Wahrheitsgehalt einer erweiterten schwach quantorenfreien Presburger-Formel $\varphi(x_1, \ldots, x_n)$ kann für eine vorgegebene Stelle $\mathbf{z} \in \mathbb{Z}^n$ stets in endlicher Zeit berechnet werden. Für nicht quantorenfreie Formeln ist das entsprechende Problem im Allgemeinen lediglich positiv semientscheidbar. Man ist daher für konkrete Problemstellungen daran interessiert, aus gegebenen Presburger-Formeln semantisch äquivalente schwach quantorenfreie Formeln zu errechnen.

Definition 1.1.17 Ein *Quantoreneliminationsverfahren* für eine Menge von Formeln Φ ist ein Algorithmus, der für jede Formel $\varphi \in \Phi$ als Eingabe eine schwach quantorenfreie Formel ψ ausgibt mit

$$\varphi \sim \psi.$$

Für $\mathbf{P}$ existiert *kein* Quantoreneliminationsverfahren, denn mit Hilfe eines solchen könnte man variablenfreie Formeln (Sätze) in den natürlichen Zahlen mit Multiplikation entscheiden. Dies widerspricht allerdings dem Gödelschen Unvollständigkeitssatz. Für eine Auswahl von Formeln, in denen quantifizierte Variablen „linear" vorkommen, läßt sich jedoch ein Quantoreneliminationsverfahren angeben.

Presburger-Terme können sowohl semantisch als auch syntaktisch mit den entsprechenden multivariaten ganzzahligen Polynomausdrücken identifiziert werden. Somit lassen sich Presburger-Terme nach üblichen Regeln für Polynomausdrücke umformen. Ein Presburger-Term heißt linear in einer Menge von Variablen $V = \{x_1, \ldots, x_n\}$, falls t sich durch Umformungen schreiben läßt als

$$a_0 + \sum_{i=1}^{n} a_i x_i,$$

sodaß $a_0, a_i \in \mathcal{T}$ und $\mathcal{V}(a_i) \cap V = \emptyset$ für alle $0 \leq i \leq n$. Eine atomare Formel $\varphi = t_1 \; \rho \; t_2$ heißt linear in V, falls sowohl t_1 als auch t_2 linear in V sind. Eine (In-)Kongruenz $\rho(t_1, t_2, m)$ heißt linear in V, falls t_1 und t_2 linear in V sind und $V \cap \mathcal{V}(m) = \emptyset$. Eine Presburger-Formel φ heißt linear in V, falls jede atomare Formel $\psi \in \mathrm{at}(\varphi)$ linear in V ist. Speziell sind auch atomare Formeln in den Bounds einer in V linearen Presburger-Formel φ linear in V.

Definition 1.1.18 (Linear quantifizierte Formeln) Die Menge der *linear quantifizierten Formeln* $\mathcal{F}_L \subseteq \mathcal{F}$ ist definiert durch

$$\mathcal{F}_L = \{\varphi \in \mathcal{F} \mid \varphi \text{ ist linear in } \mathcal{V}_g(\varphi)\}.$$

Speziell ist auch eine schwach quantorenfreie Formel φ in der Menge der Bound-Variablen von gebundenen Quantoren, die in φ vorkommen, linear. Es folgen nun einige Beispiele für linear quantifizierte bzw. nicht linear quantifizierte Presburger-Formeln.

Beispiele 1.1.19 Folgende Formeln sind zum Beispiel linear quantifiziert.

(i) $\exists x \forall y (a^2 x + b^2 y = r^2)$

(ii) $\exists x \bigvee_{(0 \leq k \leq a)(k)} ax + ky^2 \cong_m 0$

Folgende Formeln sind hingegen *nicht* linear quantifiziert.

(iii) $\exists a \forall b (a^2 x + b^2 y = r^2)$

(iv) $\exists x \bigvee_{(0 \leq k \leq a)(k)} kx + by^2 \cong_m 0$

Für die Menge der linear quantifizierten Formeln kann ein Quantoreneliminationsverfahren angegeben werden. Der Beweis dafür wird im Kapitel 3 angegeben.

1.2 Ganzzahlige parametrische Probleme

In diesem Abschnitt werden Formalisierungen von Problemen im Rahmen des eingeführten Kalküls vorgestellt, die mit dem in dieser Arbeit vorgestellten Quantoreneliminationsverfahren behandelt werden können. Die bei der Modellierung entstehende Formel wird mit Hilfe der Quantorenelimination auf eine semantisch äquivalente quantorenfreie Formel abgebildet, welche je nach Problemstellung ausgewertet werden kann um Lösungen in Abhängigkeit von den Parametern zu erhalten.

1.2.1 Parametrische Gleichungsysteme

Eine in der linearen Algebra oft anzutreffende Fragestellung ist die nach der Lösbarkeit eines linearen Gleichungssystems

$$A\mathbf{x} = \mathbf{b}$$

für $\mathbf{x} \in \mathbb{K}^k$, $A \in \mathbb{K}^{k \times n}$ und $\mathbf{b} \in \mathbb{K}^n$ oder in Abhängigkeit von Parametern über einem Körper $\mathbb{K}$. Das entsprechende Problem über den ganzen Zahlen kann mit Hilfe der Presburger-Arithmetik formuliert werden. Ein System der Form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k &= b_2 \\ \vdots \qquad\qquad & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nk}x_k &= b_n \end{aligned}$$

mit $a_{ij}, b_i \in \mathcal{T}$, sodaß x_j für $1 \leq j \leq k$ in a_{uv} und b_u mit $1 \leq u \leq n$ und $1 \leq v \leq k$ nicht vorkommt wird als ein *parametrisches Gleichungssystem* bezeichnet. Die Frage nach der Lösbarkeit dieses Systems in den ganzen Zahlen kann als eine Presburger-Formel φ formuliert werden mit

$$\begin{aligned} \varphi \quad = \quad & \exists x_1 \ldots \exists x_n (a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k = b_1 \\ \wedge \quad & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k = b_2 \\ & \vdots \\ \wedge \quad & a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nk}x_k = b_n) \end{aligned}$$

Offenbar ist die Formel φ linear quantifiziert. Die schwach quantorenfreie zu φ äquivalente Formel gibt dann in Abhängigkeit von den Belegungen der Variablen in a_{ij} die Bedingungen für die Lösbarkeit des Systems an. Neben der obigen Fragestellung können mit Hilfe der Presburger-Formeln parametrische Gleichungssysteme mit *Nebenbedingungen* formuliert werden.

Beispiel 1.2.1 Ermittle die Lösbarkeit des folgenden Gleichungssystems in den ganzen Zahlen in Abhängigkeit von a und b unter der Voraussetzung, daß $a < 0$ und $3 \mid b$.

$$\begin{aligned} ax + by &= 12 \\ bx + ay &= 15 \end{aligned}$$

Das Problem kann als Presburger-Formel φ formuliert werden mit

$$\varphi = \exists x \exists y (ax + by = 12 \wedge bx + ay = 15 \wedge a < 0 \wedge b \cong_3 0).$$

1.2.2 Kombinatorische Suchprobleme

In der Kombinatorik taucht immer wieder die Fragestellung auf, ob in einer endlichen Menge von Auswahlmöglichkeiten mindestens eine existiert, die eine bestimmte Voraussetzung erfüllt. In dieser Arbeit wird stellvertretend für kombinatorische Suchprobleme das bekannte n-Damen Problem untersucht.

Beispiel 1.2.2 (n-Damen Problem) Wann gibt es eine Aufstellung von n Damen auf einem $k \times k$ Schachbrett, sodaß keine eine andere bedroht.

Es ist offensichtlich, daß die Fragestellung nur für $n \leq k$ einen Sinn macht, da jede Dame eine Spalte bzw. Zeile für sich beansprucht. Das n-Damen Problem läßt sich als eine Presburger-Formel formulieren.

Seien $x_i, y_i \in \mathcal{V}$ mit $1 \leq i \leq n$. Definiere die Formeln

$$\begin{aligned}
\varphi_x &= \bigwedge_{1 \leq i \leq n} (1 \leq x_i \wedge x_i \leq k), \\
\varphi_y &= \bigwedge_{1 \leq i \leq n} (1 \leq y_i \wedge y_i \leq k) \text{ und} \\
\psi &= \bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq n, j \neq i} (x_i \neq x_j \wedge y_i \neq y_j \wedge x_i - x_j \neq y_i - y_j \wedge x_i - x_j \neq y_j - y_i).
\end{aligned}$$

Man sollte dabei beachten, daß das Symbol „$\bigwedge$" hier als Abkürzung für lange Konjunktionen verwendet wird. Das n-Damen Problem läßt sich nun formulieren durch

$$\gamma = \exists x_1 \ldots \exists x_n \exists y_1 \ldots \exists y_n (\varphi_x \wedge \varphi_y \wedge \psi).$$

Für $n > k$ kann man nach obigen Überlegungen erwarten, daß γ zu false äquivalent ist. Es gibt allerdings auch Werte von n und k mit $n \leq k$, sodaß in jeder Aufstellung mindestens eine der Damen eine andere bedroht. Als Beispiel dazu betrachte ein 2×2 Brett mit 2 Damen.

1.2.3 Ganzzahlige lineare Optimierung

In diesem Abschnitt wird gezeigt, wie man lineare ganzzahlige Optimierungsprobleme mit und ohne Parameter durch Presburger-Formeln modellieren kann. Ferner wird der Zusammenhang zwischen der Erfüllungsmenge der durch die Modellierung erhaltenen Formel und den Lösungen des Problems vorgestellt.

Definition 1.2.3 (Lineares ganzzahliges Optimierungsproblem) Ein *lineares ganzzahliges Optimierungsproblem* (C, q, P) ist gegeben durch eine endliche Menge von Gleichungen und Ungleichungen $C \subseteq \mathcal{A}$, genannt *Zusicherungen* oder Constraints, eine Kostenfunktion $q \in \mathcal{T}$ und eine endliche Menge von Parametern $P \subseteq \mathcal{V}$, sodaß sowohl q als auch jedes $\psi \in C$ in allen Variablen außer der Parameter P linear ist.

Wie üblich, ist das Ziel für jede feste Wahl von Werten $\mathbf{u} = (u_1, \ldots, u_k) \in \mathbb{Z}^k$ des Parametertupels $(p_1, \ldots, p_k)$ Belegungen der verbleibenden Variablen anzugeben, die die Termfunktion $q^{\mathbf{P}}$ minimieren.

Definition 1.2.4 (Zulässige Lösung, Optimale Lösung) Sei (C, q, P) ein lineares ganzzahliges Optimierungsproblem mit $\{x_1, \ldots, x_n\} = (\mathcal{V}(q) \cup \bigcup_{\psi \in C} \mathcal{V}(\psi)) \setminus P$ und den Parametern $P = \{p_1, \ldots, p_k\}$. Definiere die erweiterte Formel φ durch

$$\varphi(p_1, \ldots, p_k, x_1, \ldots, x_n) = \left(\bigwedge_{\psi \in C} \psi \right) (p_1, \ldots, p_k, x_1, \ldots, x_n).$$

Für ein $\mathbf{u} \in \mathbb{Z}^k$ heißt die Erfüllungsmenge

$$L_{\mathbf{u}} = S_{\mathbf{u}}^{(p_1,\ldots,p_k)}(\varphi(p_1,\ldots,p_k,x_1,\ldots,x_n))$$

die Menge der *zulässigen Lösungen*. Als *Lösung* von (C,q,P) für eine Belegung $\mathbf{u} \in \mathbb{Z}^k$ der Parameter wird eine Stelle $\mathbf{a} \in L_{\mathbf{u}}$ bezeichnet mit $q^{\mathbf{P}}(\mathbf{u},\mathbf{a}) \leq q^{\mathbf{P}}(\mathbf{u},\mathbf{b})$ für alle $\mathbf{b} \in L_{\mathbf{u}}$.

Ein lineares ganzzahliges Optimierungsproblem heißt für eine Belegung der Parameter *unlösbar*, falls die entsprechende Menge der zulässigen Lösungen bezüglich der gewählten Belegung leer ist. Ein lineares ganzzahliges Optimierungsproblem heißt für eine Belegung der Parameter *unbeschränkt*, falls für die Belegung keine Lösung des Problems existiert, obwohl die entsprechende Menge der zulässigen Lösungen bezüglich der gewählten Belegung nicht leer ist. Eine Lösung eines ganzzahligen linearen Optimierungsproblems ist also zusammenfassend eine Abbildung, die jeder Belegung der Parameter entweder eine Menge von Lösungen bezüglich dieser Belegung der Parameter zuordnet, oder angibt, daß das Problem bezüglich der gewählten Parameter unlösbar oder unbeschränkt ist. Für ein Problem ohne Parameter übertragen sich die Begriffe „Lösung", „unlösbar" und „unbeschränkt" auf natürliche Weise auf das Problem selbst.

Satz 1.2.5 (Lineare ganzzahlige Optimierung ohne Parameter) *Sei ein lineares ganzzahliges Optimierungsproblem $(C,q,\emptyset)$ mit $\{x_1,\ldots,x_n\} = \mathcal{V}(q) \cup \bigcup_{\psi\in C} \mathcal{V}(\psi)$ gegeben. Definiere für $z \in \mathcal{V}$ mit $z \notin \{x_1,\ldots,x_n\}$*

$$\varphi = \exists x_1 \exists x_2 \ldots \exists x_n \left(\bigwedge_{\psi\in C} \psi \wedge z \geq q \right).$$

Dann ist φ linear quantifiziert.

(i) Das Problem $(C,q,\emptyset)$ ist genau dann unlösbar, wenn gilt $S(\varphi(z)) = \emptyset$.

(ii) Das Problem $(C,q,\emptyset)$ ist genau dann unbeschränkt, wenn gilt $S(\varphi(z)) \neq \emptyset$ und $S(\varphi(z))$ ist nach unten unbeschränkt.

(iii) Falls eine Lösung von $(C,q,\emptyset)$ existiert, so gilt für die Menge L aller *Lösungen von $(C,q,\emptyset)$*

$$L = S(((\bigwedge_{\psi\in C} \psi) \wedge q = \min S(\varphi(z)))(x_1,\ldots,x_n)).$$

Beweis: (i) Falls $(C,q,\emptyset)$ unlösbar ist, so ist bereits nach Definition 1.2.4 $\bigwedge_{\psi\in C} \psi \sim$ false und somit auch $\varphi \sim$ false. Falls $S(\varphi(z)) = \emptyset$, so ist entweder $\bigwedge_{\psi\in C} \psi \sim$ false oder $q \geq z \sim$ false. Die atomare Formel $q \geq z$ ist aber trivialerweise stets erfüllbar. Also gilt $\bigwedge_{\psi\in C} \psi \sim$ false. (ii) Sei das Problem $(C,q,\emptyset)$ unbeschränkt, aber $S(\varphi(z))$ nach unten beschränkt und nichtleer (der Fall, daß $S(\varphi(z)) = \emptyset$ ist, ist nach (i) bereits behandelt worden) und sei $m = \min S(\varphi(z))$. Dann existiert $\mathbf{z} = (z_1,\ldots,z_n) \in \mathbb{Z}^n$ mit $(\bigwedge_{\psi\in C} \psi)^{\mathbf{P}}(\mathbf{z}) = \top$ und $q^{\mathbf{P}}(\mathbf{z}) \geq m$. Nun ist aber $\mathbf{z} = (z_1,\ldots,z_n)$ wegen der Minimalität von m eine Lösung des Problems $(C,q,\emptyset)$, was ein Widerspruch zur Annahme ist. Umgekehrt existiert wegen (i) mindestens eine Zulässige Lösung. Angenommen $\mathbf{z}$ ist eine Lösung von $(C,q,\emptyset)$ und sei $q^{\mathbf{P}}(\mathbf{z}) = m$. Wähle ein $m' \in S(\varphi(z))$ mit $m' < m$. Es gibt offensichtlich ein $\mathbf{z}'$ mit $q^{\mathbf{P}}(\mathbf{z}') = m' < m = q^{\mathbf{P}}(\mathbf{z})$, was ein Widerspruch zur Optimalität von $\mathbf{z}$ ist. Also existiert keine Lösung des Problems. (iii) Sei $\mathbf{z}$ eine Lösung von $(C,q,\emptyset)$. Dann gilt $m = q^{\mathbf{P}}(\mathbf{z}) \leq q^{\mathbf{P}}(\mathbf{b})$ für jede zulässige Lösung $\mathbf{b}$. Somit ist $m = \min S(\varphi(z))$. Daraus folgt die Behauptung. Die Umkehrung läuft analog. □

Satz 1.2.5 liefert einen algorithmischen Zugang zur Bestimmung der Lösbarkeit linearer ganzzahliger Optimierungsprobleme ohne Parameter mit Hilfe eines Quantoreneliminationsverfahrens. Es reicht dazu offensichtlich mit Hilfe der Quantorenelimination eine äquivalente quantorenfreie Formel zur Formel φ aus Satz 1.2.5 zu finden und diese auf Beschränktheit nach unten zu untersuchen. Inwieweit dies in der Praxis durchführbar ist, wird in Kapitel 4 diskutiert. Die Vorgehensweise wird an einem Beispiel veranschaulicht.

Beispiel 1.2.6 Minimiere $x + y + z$, wobei für die Menge der Zusicherungen gilt

$$C = \{x > 0, y > 0, z > 0\}$$

und die Menge der Parameter P leer ist. Aus den Angaben läßt sich nach Satz 1.2.5 die Formel

$$\varphi = \exists x \exists y \exists z (0 < x \wedge 0 < y \wedge 0 < z \wedge x + y + z < c)$$

formulieren, deren Erfüllungsmenge, wie bereits am Anfang dieses Kapitels angemerkt wurde, durch die Formel $c > 3$ beschrieben wird. Es ist leicht einzusehen, daß eine erfüllende Belegung von

$$(0 < x \wedge 0 < y \wedge 0 < z \wedge x + y + z = 4)(x, y, z)$$

zum Beispiel durch den Wert $\mathbf{z} = (2, 1, 1)$ gegeben ist.

Als Nächstes wird noch der allgemeinere Fall der linearen Optimierungsprobleme mit Parametern behandelt.

Satz 1.2.7 (Lineare ganzzahlige Optimierung mit Parametern) *Sei ein lineares ganzzahliges Optimierungsproblem* (C, q, P) *mit* $P = \{p_1, \ldots, p_k\}$ *und* $\{x_1, \ldots, x_n\} = (\mathcal{V}(q) \cup \bigcup_{\psi \in C} \mathcal{V}(\psi)) \setminus P$ *gegeben. Definiere*

$$\varphi(p_1, \ldots, p_k, z) = (\exists x_1 \exists x_2 \ldots \exists x_n \bigwedge_{\psi \in C} (\psi \wedge z \geq q))(p_1, \ldots, p_k, z).$$

Dann ist φ *linear quantifiziert. Für jede Wahl von Parametern* $\mathbf{u} \in \mathbb{Z}^k$ *gelten folgende Aussagen.*

(i) Das Problem (C, q, P) *ist genau dann unlösbar, wenn gilt* $S_{\mathbf{u}}^{(p_1, \ldots, p_k)}(\varphi(p_1, \ldots, p_k, z)) = \emptyset$.

(ii) Das Problem (C, q, P) *ist genau dann unbeschränkt, wenn gilt* $S_{\mathbf{u}}^{(p_1, \ldots, p_k)}(\varphi(p_1, \ldots, p_k, z)) \neq \emptyset$ *und* $S_{\mathbf{u}}^{(p_1, \ldots, p_k)}(\varphi(p_1, \ldots, p_k, z))$ *ist nach unten unbeschränkt.*

(iii) Falls eine Lösung von (C, q, P) *existiert, so gilt für die Menge* L aller *Lösungen von* (C, q, P)

$$L = S(((\bigwedge_{\psi \in C} \psi) \wedge q = \min S(\varphi_{\mathbf{u}}^{p_1, \ldots, p_k}(p_1, \ldots, p_k, z)))(x_1, \ldots, x_n)).$$

Beweis: Für eine feste Wahl von Parameterwerten folgen die Behauptungen aus Satz 1.2.5. □

1.2.4 Abhängigkeitsanalye und Programmverifikation

In diesem Abschnitt werden zwei typische Problemstellungen betrachtet, die im Zusammenhang mit der Analyse von Programmtexten auftreten. *Abhängigkeitsanalyse* liefert Abhängigkeiten von Werten der Speicherzellen, die während des Programmablaufes auftreten, und *Programmverifikation* analysiert die Semantik eines gegebenen Programmtextes.

Bei der Parallelisierung von Programmabläufen auf Mehrprozessor- bzw. Mehrrechnersystemen ist die Bestimmung von Datenabhängigkeiten von zentraler Bedeutung [Fea91], [Gri04]. In diesem Abschnitt wird gezeigt, wie das Problem nach der Suche von parametrischen Datenabhängigkeiten als eine Presburger-Formel formuliert werden kann.

Datenabhängigkeiten entstehen, wenn auf eine Speicherzelle während des Programmablaufes mehrfach zugegriffen wird, wobei mindestens einer der Zugriffe schreibend ist. Besonders in Schleifen, deren Rumpf Zugriffe auf einen Vektor von Speicherzellen indiziert durch Schleifenvariablen enthält, ist das Bestimmen von Datenabhängigkeiten schwierig. Featurier gab 1991 in [Fea91] einen Algorithmus zur Bestimmung von Datenabhängigkeiten in einfachen Schleifenprogrammen, deren Abhängigkeitenstruktur sich als eine nicht uniforme Presburger-Formel formulieren läßt. Das Verfahren von Featurier läßt sich auf das Lösen von linearen Ungleichungssystemen zurückführen und kann mit Hilfe der Bibliothek PIP [Fea88], welche lediglich mit linearen multivariaten Polynomen mit ganzzahligen Koeffizienten umgehen kann, umgesetzt werden.

Beispiel 1.2.8 Betrachte den folgenden Programmabschnitt in einer vereinfachten imperativen Programmiersprache. Zu bestimmen sind die Abhängigkeiten zwischen den Werten des Arrays A während des Schleifenablaufes.

```
for I = 1 : N  do
    for J = 1 : I  do A[I,J] = A[I,N-J];
    od;
od;
```

Um die Abhängigkeiten zu bestimmen ist es erforderlich Werte der Variablen I und J zu bestimmen, bei denen auf die selbe Speicherzelle A[I,J] zugegriffen wird. Dies kann man durch die Formel φ ausdrücken mit

$$\begin{aligned}\varphi &= \exists i \exists j (1 \leq i \wedge i \leq n \wedge 1 \leq j \wedge j \leq i \\ &\wedge \quad 1 \leq i_1 \wedge i_1 \leq n \wedge 1 \leq j_1 \wedge j_1 \leq i_1 \\ &\wedge \quad i = i_1 \wedge j = n - j_1).\end{aligned}$$

Die Formel φ ist offensichtlich linear quantifiziert. Eine äquivalente quantorenfreie Ergebnisformel φ' beschreibt in Abhängigkeit vom Wert von N durch erfüllende Belegungen der Variablen i_1 und j_1 solche Werte der Schleifenvariablen, die bei einem Lesezugriff Stellen mit Abhängigkeiten indizieren. Quantifiziert man zusätzlich auch i_1 und j_1 existentiell, so beschreibt eine zu $\exists i_1 \exists j_1 \varphi$ äquivalente quantorenfreie Formel die Existenz von Datenabhängigkeiten in Abhängigkeit von n. In einer praktischen Anwendung wäre zusätzlich noch die Richtung der Abhängigkeiten von Bedeutung, was zu zusätzlichen Bedingungen an die Werte der Schleifenvariablen führt. Darauf wird hier nicht näher eingegangen.

Das folgende Beispiel ist eine Instanz des Problems der Abhängigkeitsanalyse, die *nicht* mit dem Verfahren von Featurier in [Fea88] lösbar ist.

Beispiel 1.2.9 Betrachte den folgenden Programmabschnitt in einer vereinfachten imperativen Programmiersprache. Zu bestimmen sind die Abhängigkeiten zwischen den Werten des Arrays A während des Schleifenablaufes.

```
for I = 1 : N  do
    for J = 1 : N  do
        A[I*N+J] = if (J < I)  then
            A[J*N+I];
        else
            A[J*N-I];
        fi;
    od;
od;
```

Dabei erzwingt die Syntax des Konditionsstatements, daß der Schreibzugriff stets mit den selben Indizes durchgeführt wird. Die Semantik der Abhängigkeiten kann man durch folgende Formel ausdrücken.

$$\begin{aligned}\varphi &= \exists i \exists j (1 \leq i \wedge i \leq n \wedge 1 \leq j \wedge j \leq n \\ &\wedge \quad 1 \leq i_1 \wedge i_1 \leq n \wedge 1 \leq j_1 \wedge j_1 \leq n \\ &\wedge \quad ((j_1 < i_1 \wedge in + j = j_1 n + i_1) \\ &\vee \quad (j_1 \geq i_1 \wedge in + j = j_1 n - i_1))).\end{aligned}$$

Die Formel φ ist ebenfalls linear quantifiziert, ist aber im Gegensatz zum Beispiel 1.2.8 uniform im Sinne der Definition 1.1.3. Analog zu oben liefert eine quantorenfreie äquivalente Formel Werte von I und J an denen in Abhängigkeit von n Datenabhängigkeiten vorliegen. An dieser Stelle ist es sinnvoll auf die Änderungen im Bezug auf das Beispiel 1.2.8 hinzuweisen.

(i) Das Programm enthält Konditionsstatements, was dazu führt, daß in der Ergebnisformel auch Disjunktionen vorhanden sind.

(ii) Das Programm enthält multiplikative parametrische Array-Indizes, was zu *nicht* konstanten Koeffizienten der quantifizierten Variablen führt.

Als Kontraargument für die Aussage (i) könnte man anführen, daß jede stark quantorenfreie Formel in eine DNF überführt werden kann. Da allerdings die Anzahl der atomaren Formeln einer DNF einer stark quantorenfreien Formel im Worst-Case exponentiell wächst, ist eine solche Vorgehensweise im Hinblick auf die Komplexität des Verfahrens vor allem bei großen Eingabeformeln bzw. langen Programmtexten bedenklich.

Als *automatische Programmverifikation* bezeichnet man den Vorgang der Überprüfung der Semantik eines durch einen Text angegebenen Programms auf Korrektheit. In diesem Abschnitt wird ein Beispiel für ein Programm angegeben, dessen semantische Bedeutung anhand des Programmtextes als eine Presburger-Formel formuliert werden kann um anschließend mit der erwünschten Semantik verglichen zu werden.

Beispiel 1.2.10 Bestimme die Semantik des folgenden Programms in einer vereinfachten imperativen Programmiersprache mit den Eingaben $A, B \in \mathbb{Z}$ und der Ausgabe $C \in \mathbb{Z}$.

if A $<$ B **then** X = A; Y = B **else** Y = A; X = B; **fi**;
while X $<$ Y
    X = X + 1;
    Y = Y - 1;
od;
if X = Y **then** C = X **else** C = Y; **fi**;

Sei x bzw. y der Anfangswert der Variablen X bzw. Y. Nach z vielen Schleifendurchläufen ist offenbar $u = x + z$ bzw. $v = y - z$ der Wert der Variable X bzw. Y. Somit terminiert das Programm mit der Ausgabe c genau dann, wenn die Formel $\varphi(a, b, c)$ an einer Stelle $\mathbf{z} \in \mathbb{Z}^3$ gilt mit

$$\begin{aligned}\varphi \quad &= \quad \exists x \exists y \exists u \exists v \exists z (((a < b \wedge x = a \wedge y = b) \vee (b \leq a \wedge y = a \wedge x = b)) \\ &\wedge \quad 0 \leq z \wedge u = x + z \wedge v = y - z \wedge v \leq u \wedge v + 1 > u - 1 \\ &\wedge \quad ((u = v \wedge c = u) \vee (\neg(u = v) \wedge (c = v)))).\end{aligned}$$

1.3 Zusammenfassung

In diesem Kapitel wurden zunächst die Ziele und der Aufbau der vorliegenden Arbeit umrissen.

Im ersten Abschnitt wurde die uniforme Presburger-Arithmetik als eine Struktur über der Sprache der Ringe, erweitert um die Symbole für (In-)Kongruenzen, eingeführt. Die Terme der Presburger-Formeln entsprechen dabei multivariaten ganzzahligen Polynomausdrücken. Es stellte sich heraus, daß der Begriff einer gewöhnlichen Kongruenz $a \cong_m b$, wie man sie aus Zahlentheorie kennt, für beliebige ganzzahlige Werte von m fortgesetzt werden kann. Dabei entspricht eine Kongruenz $a \cong_0 b$ der Gleichung $a = b$. Als neue syntaktische Objekte wurden gebundene Quantoren eingeführt.

Diese stellen intuitiv einerseits Konjunktionen bzw. Disjunktionen von Formeln dar, die dadurch entstehen, daß eine ausgezeichnete Variable durch erfüllende Werte einer Formel, genannt Bound, ersetzt wird. Andererseits sind gebundene Quantoren zu gewöhnlich quantifizierten Formeln äquivalent. Für eine Teilmenge von Formeln der eingeführten uniformen Presburger-Arithmetik ist ein Quantoreneliminationsverfahren angekündigt worden. Dieses wird im Kapitel 3 ausführlich behandelt.

Im zweiten Abschnitt sind einige Problemstellungen vorgestellt worden, die mit Hilfe von Formeln der uniformen Presburger-Arithmetik formuliert werden können. Neben kombinatorischen Problemen können parametrische Gleichungssysteme und lineare ganzzahlige Optimierungsprobleme mit und ohne Parameter in Termini der Presburger-Formeln formuliert werden. Weiterhin können für die Forschung auf dem Gebiet der Parallelisierung wichtige Probleme der Bestimmung von Datenabhängigkeiten in Programmen ebenfalls durch Presburger-Formeln ausgedrückt werden. Dabei können auch solche Probleme formuliert werden, die mit bisherigen Methoden nicht behandelt werden konnten. Bestimmung und Verifikation der Semantik eines durch Programmcode gegebenen Algorithmus ist unter anderem für den Compiler-Bau von Bedeutung. Semantik einfacher Programmabschnitte kann mit Hilfe der Presburger-Formeln formuliert werden. Anhand der schwach quantorenfreien äquivalenten Formel kann zum Beispiel die errechnete Semantik mit einer vorgegebenen erwünschten durch eine weitere Anwendung der Quantorenelimination verglichen werden. Eine andere Anwendung ist die Möglichkeit die Ausgabewerte des Programms zu errechnen, ohne das Programm zu starten. Dies liefert eine Hilfestellung für die Korrektur der Fehler in einem Programm.

Kapitel 2

Simplifikation

Aufgrund der hohen Komplexität der Quantorenelimination ist es wichtig die Zwischenergebnisse in einer möglichst einfachen Form zu halten. Ein Algorithmus, der einer Presburger-Formel eine bezüglich festgelegter Kriterien einfachere semantisch äquivalente Formel zuordnet, wird als *Simplifikationsalgorithmus* bezeichnet. Einige dieser Kriterien, die bereits in [DS97b] genannt wurden, werden hier aus Gründen der Vollständigkeit aufgelistet.

(i) Eine äquivalente Formel mit *einfacheren Termen* wird als einfacher angesehen. So ist $y^2+1 \cong_4 0$ der Formel $4yx^2 + y^2 + 1 \cong_4 0$ vorzuziehen. Strategien zur Simplifikation von atomaren Formeln werden in Abschnitt 2.1 angegeben.

(ii) Eine äquivalente Formel mit *weniger atomaren Formeln* wird als einfacher angesehen. So ist $t = 0$ der Konjunktion $t \leq 0 \wedge t \geq 0$ vorzuziehen. Der Algorithmus zur Simplifikation von stark quantorenfreien Formeln in Abschnitt 2.2 verfolgt unter anderem dieses Ziel.

(iii) Eine äquivalente Formel mit *verständlicher boolescher Struktur* wird als einfacher angesehen. Als verständlich werden unter anderem Formeln geringer maximaler Tiefe und Fallunterscheidungen angesehen. Dieses allgemeine Ziel wird nicht nur durch reine Simplifikationsmaßnahmen verfolgt. Die Erweiterungen der Quantorenelimination aus Kapitel 3 liefern unter anderem Formeln geringerer maximaler Tiefe, als das Kernverfahren.

Für folgende Ausführungen werden triviale Simplifikationen, wie etwa die Elimination von true und false bei der Anwendung der Dominanzgesetze $\text{true} \wedge \varphi \sim \varphi$, $\text{true} \vee \varphi \sim \text{true}$, $\text{false} \wedge \varphi \sim \text{false}$ und $\text{false} \vee \varphi \sim \varphi$ oder die Anwendung der Idempotenzgesetze, nicht explizit behandelt. Es wird vielmehr stets vorausgesetzt, daß diese sowie auf die Eingabe- als auch auf die Ausgabeformeln angewandt werden.

2.1 Simplifikation atomarer Formeln

In diesem Abschnitt werden Algorithmen zur Simplifikation von atomaren Formeln vorgestellt. Der komplexe Algorithmus wird als eine Komposition von Teilalgorithmen, den sogenannten Simplifikationsregeln, die jeweils ein Teilproblem behandeln, angegeben. Im Folgenden werden multivariate ganzzahlige Polynomausdrücke mit den entsprechenden Presburger-Termen identifiziert.

2.1.1 Normalformen und Auswertungen atomarer Formeln

Atomare Presburger-Formeln werden im Folgenden durch die Simplifikationsregel NF zu einer *Normalform* vereinfacht. Jeder Presburger-Term t mit $\mathcal{V}(t) = \{x_1, ..., x_n\}$ besitzt eine im Sinne der Gleichheit der Termfunktionen äquivalente *distributive Normalform* t' der Form

$$\sum_{\mathbf{p}=(p_1,\ldots,p_n)\in P} a_{\mathbf{p}} x_1^{p_1} ... x_n^{p_n},$$

wobei P endlich ist und $a_{\mathbf{p}} \in \mathbb{Z}$. Die Terme $a_{\mathbf{p}} x_1^{p_1} ... x_n^{p_n}$ werden als Monome und die Terme $a_{\mathbf{p}}$ als Koeffizienten von t bezeichnet. Als *führendes Monom* wird ein Monom bezeichnet, das bezüglich einer zulässigen totalen Ordnung auf Monomen unter allen Monomen des Terms maximal ist. Der Koeffizient des führenden Monoms wird als *führender Koeffizient* bezeichnet. Für Beispiele in den folgenden Ausführungen wird die lexikographische Ordnung auf Monomen unter Verwendung der alphabetischen Ordnung auf Variablenzeichen verwendet. In der Implementierung hängt die Darstellung der Polynome von der Ordnung der Variablen ab. Darauf wird hier nicht näher eingegengen.

Definition 2.1.1 (Normalform atomarer Formeln) Eine atomare Formel $\varphi = t_1 \ \rho \ t_2$ heißt in Normalform, falls $t_2 = 0$, t_1 in distributiver Normalform ist und der führende Koeffizient von t_1 positiv ist. Eine (In-)Kongruenz $t_1 \ \rho_m \ t_2$ heißt in Normalform, falls $t_2 = 0$, t_1 und m in distributiver Normalform sind und die führenden Koeffizienten von t_1 und m positiv sind.

Für Ungleichungen muß bei der Umformung in eine Normalform unter Umständen das Relationszeichen „$<$" durch „$>$", „$\leq$" durch „$\geq$" und umgekehrt ausgetauscht werden. Für (In-)Kongruenzen folgt die Existenz einer solchen Normalform aus Lemma 1.1.6. Weiterhin folgt für (In-)Kongruenzen in Normalform mit einem ganzzahligen Modulus m, daß $m \geq 0$ ist. Bei der Angabe von Simplifikationsregeln wird stets angenommen, daß die Eingabeformel in obiger Normalform vorliegt.

Beispiel 2.1.2 Folgende atomare Formeln sind in Normalform.

(i) $a^3b^2c^2 + a^2c^2 + c - 3 = 0$.

(ii) $a^3b^2c^2 + 2 \cong_{a^2-b^2+1} 0$.

Variablenfreie atomare Formeln sind stets entweder zu true oder zu false äquivalent. Die *Auswertung variablenfreier atomarer Formeln* wird durch die Regel VF abgekürzt.

Kongruenzen bzw. Inkongruenzen $t \ \rho_m \ 0$ mit $m = 0$ werden nach Lemma 1.1.6 zu Gleichungen bzw. Disgleichungen umgeformt. Die entsprechende Simplifikationsregel wird durch ZC abgekürzt.

Offensichtlich ist jede Kongruenz der Form $0 \cong_t 0$ zu true und jede Inkongruenz der Form $0 \ncong_t 0$ zu false äquivalent. Im Gegensatz dazu läßt sich im Allgemeinen eine beliebige (In-)Kongruenz mit $t \ \rho_m \ 0$ mit $\mathcal{V}(m) \neq \emptyset$ und t von der Form $z \in \mathbb{Z} \setminus \{0\}$ *nicht* zu true oder zu false vereinfachen.

Lemma 2.1.3 Für $m \in \mathcal{T}$ und $t \in \mathbb{Z} \setminus \{0\}$ gilt

$$t \cong_m 0 \sim \bigvee_{l|t} m = l \quad \text{bzw.} \quad t \ncong_m 0 \sim \bigwedge_{l|t} m \neq l.$$

Beweis: Nach Lemma 1.1.6(iii) gilt $\mathbf{P} \vDash (t \cong_m 0)(\mathbf{z})$ genau dann, wenn $m^{\mathbf{P}}(\mathbf{z}) \mid t$. Da man alle Teiler von $t \in \mathbb{Z} \setminus \{0\}$ explizit angeben kann, ist die Formel wohldefiniert. Die Behauptung für Inkongruenzen ergibt sich aus der Behauptung für Kongruenzen durch logische Negation der beiden Seiten. □

Beispiel 2.1.4 Die Kongruenz $4 \cong_{2a+1} 0$ ist nach Lemma 2.1.3 äquivalent zu

$$2a + 5 = 0 \vee 2a + 3 = 0 \vee 2a + 2 = 0 \vee 2a = 0 \vee 2a - 1 = 0 \vee 2a - 3 = 0.$$

Für betragsmäßig große Werte von t wächst die Anzahl der atomaren Formeln auf den rechten Seiten der Äquivalenzen aus Lemma 2.1.3 exponentiell. Die Anwendung des Lemmas 2.1.3 für große Werte von t scheint daher nicht sinnvoll zu sein. Die entsprechende Simplifikationsregel wird durch TC_t abgekürzt, wobei $t \in \mathbb{N}$ der maximale Wert von t ist, bei dem Lemma 2.1.3 angewandt wird.

2.1.2 Definitheit von Termen

Ein Presburger-Term t heißt *positiv definit* bzw. *negativ definit*, falls für alle $\mathbf{z} \in \mathbb{Z}^n$ für die Termfunktion von $t(x_1, ..., x_n)$ gilt

$$t^{\mathbf{P}}(\mathbf{z}) > 0 \quad \text{bzw.} \quad t^{\mathbf{P}}(\mathbf{z}) < 0.$$

Ein Presburger-Term t heißt *positiv semidefinit* bzw. *negativ semidefinit*, falls für alle $\mathbf{z} \in \mathbb{Z}^n$ für die Termfunktion von $t(x_1, ..., x_n)$ gilt

$$t^{\mathbf{P}}(\mathbf{z}) \geq 0 \quad \text{bzw.} \quad t^{\mathbf{P}}(\mathbf{z}) \leq 0.$$

Besitzt ein Presburger-Term eine der obigen Eigenschaften, so wird dieser als *definit* bezeichnet. Eine Auswahl von Regeln, die anwendbar sind, falls ein Term als positiv oder negativ definit erkannt wird, wird im folgenden Lemma vorgestellt.

Lemma 2.1.5 Sei t ein positiv definiter Term. Dann gelten folgende Äquivalenzen.

(i) $t = 0 \sim t < 0 \sim t \leq 0 \sim$ false.

(ii) $t \neq 0 \sim t > 0 \sim t \geq 0 \sim$ true.

(iii) $tq\ \rho\ 0 \sim q\ \rho\ 0$ für jedes $q \in \mathcal{T}$ und für jedes $\rho \in \{=, \neq, <, >, \leq, \geq\}$.

Analoges gilt auch für negativ definite Terme. Die Anwendung des Lemmas 2.1.5(i)-(ii) wird mit DT abgekürzt. Es existieren hinreichende Bedingungen um die Definitheit von Polynomen nachzuweisen.

Lemma 2.1.6 Sei t ein Presburger-Term in distributiver Normalform mit $\mathcal{V}(t) = \{x_1, ..., x_n\}$. Dann gilt:

(i) t ist positiv definit bzw. semidefinit, wenn in jedem Monom $a_{\mathbf{p}} x_1^{p_1} x_2^{p_2} ... x_n^{p_n}$ jeder Exponent p_i gerade ist und der Koeffizient $a_{\mathbf{p}}$ echt positiv bzw. positiv ist.

(ii) t ist negativ definit bzw. semidefinit, wenn in jedem Monom $a_{\mathbf{p}} x_1^{p_1} x_2^{p_2} ... x_n^{p_n}$ jeder Exponent p_i gerade ist und der Koeffizient $a_{\mathbf{p}}$ echt negativ bzw. negativ ist.

Für Terme mit vorkommenden ungeraden Exponenten ist nach Lemma 2.1.6 keine Aussage möglich. Die Anwendung des Lemmas 2.1.6 wird nun an Beispielen veranschaulicht.

Beispiele 2.1.7 Betrachte folgende Presburger Terme.

(i) Der Term $t_1 = 4x^6y^2 + 3x^4z^2 + 1$ ist nach Lemma 2.1.6(i) positiv definit.

(ii) Der Term $t_2 = -4x^6y^2 - 3x^4z^2$ ist nach Lemma 2.1.6(ii) negativ semidefinit.

(iii) Über den Term $t_3 = a^2x + b^2y + 1$ ist nach Lemma 2.1.6 keine Aussage möglich.

Wie man aus obigem Beispiel erkennt, erfüllen die für die Quantorenelimination relevanten Terme von atomaren Presburger-Formeln die Voraussetzungen des Lemmas 2.1.6 *nicht*, da in diesen mindestens eine Variable, die durch einen Quantor gebunden ist, linear, also mit ungeradem Exponenten, vorkommt.

2.1.3 Modulo-Reduktion

Die Technik jeden Koeffizienten des Terms $t = a_0 + a_1x_1 + ... + a_nx_n$ mit $a_i \in \mathbb{Z}$ für $0 \leq i \leq n$ durch den Rest bei der Teilung durch den Modulus einer (In-)Kongruenz $t\ \rho_m\ 0$ mit $m \in \mathbb{Z}$ zu ersetzen ist bereits für den nicht uniformen Fall als *Modulo-Reduktion* bekannt. In diesem Abschnitt wird eine Erweiterung dieser Technik für die uniforme Presburger-Arithmetik vorgestellt.

Beispiel 2.1.8 Die Kongruenz $\varphi = 5x + 6 \cong_4 0$ ist äquivalent zu $x + 2 \cong_4 0$.

Für einen ganzzahligen Modulus $m \in \mathbb{Z} \setminus \{0\}$ und einen beliebigen uniformen Presburger-Term t existiert eine analoge Vorgehensweise um t in einer (In-)Kongruenz $t\ \rho_m\ 0$ „Modulo m zu reduzieren".

Lemma 2.1.9 (Modulo-Reduktion) Seien $t \in \mathcal{T}$ in distributiver Normalform und $m \in \mathbb{Z}\setminus\{0\}$. Sei r der Term, der aus t entsteht, indem man in t jeden ganzzahligen Koeffizienten a durch den positiven Rest bei der Teilung von a durch m ersetzt. Dann gilt

$$t \cong_m 0 \sim r \cong_m 0 \quad \text{und} \quad t \ncong_m 0 \sim r \ncong_m 0$$

Beweis: Sei $t = a_1t_1 + ... + a_nt_n$, wobei a_it_i die Monome von t und $a_i \in \mathbb{Z} \setminus \{0\}$ für $1 \leq i \leq n$ die Koeffizienten von t sind. Sei weiterhin $a_i = l_im + a'_i$ mit $0 \leq a'_i < m$ für $1 \leq i \leq n$. Gelte zunächst $\mathbf{P} \vDash (t \cong_m 0)(\mathbf{z})$ für eine Stelle $\mathbf{z}$. Dann gilt

$$\sum_{i=1}^{n} a_i t_i^{\mathbf{P}}(\mathbf{z}) + mk = \sum_{i=1}^{n} (l_im + a'_i) t_i^{\mathbf{P}}(\mathbf{z}) + mk = \underbrace{\sum_{i=1}^{n} a'_i t_i^{\mathbf{P}}(\mathbf{z})}_{r^{\mathbf{P}}(\mathbf{z})} + m \underbrace{\left(\sum_{i=1}^{n} l_i t_i^{\mathbf{P}}(\mathbf{z}) + k \right)}_{k'}.$$

Somit folgt aus $t^{\mathbf{P}}(\mathbf{z}) + mk = 0$ auch $r^{\mathbf{P}}(\mathbf{z}) + mk' = 0$ mit k' wie oben. Gelte umgekehrt $\mathbf{P} \vDash (r \cong_m 0)(\mathbf{z})$ für ein $\mathbf{z} \in \mathbb{Z}$. Dann gilt

$$\sum_{i=1}^{n} a'_i t_i^{\mathbf{P}}(\mathbf{z}) + mk = \sum_{i=1}^{n} (a_i - l_im) t_i^{\mathbf{P}}(\mathbf{z}) + mk = \underbrace{\sum_{i=1}^{n} a_i t_i^{\mathbf{P}}(\mathbf{z})}_{t^{\mathbf{P}}(\mathbf{z})} + m \underbrace{\left(\sum_{i=1}^{n} -l_i t_i^{\mathbf{P}}(\mathbf{z}) + k \right)}_{k'}.$$

Somit folgt aus $r^{\mathbf{P}}(\mathbf{z}) + mk = 0$ auch $t^{\mathbf{P}}(\mathbf{z}) + mk' = 0$ mit k' wie oben. Für Inkongruenzen folgt die Behauptung aus der Äquivalenz

$$t \ncong_m 0 \sim \neg t \cong_m 0 \sim \neg r \cong_m 0 \sim r \ncong_m 0. \quad \square$$

Die Anwendung des Lemmas 2.1.9 wird mit der Regel MR abgekürzt.

Beispiele 2.1.10 Nach Lemma 2.1.9 gelten folgende Aussagen.

(i) $4x^2 + y \cong_4 0 \sim y \cong_4 0$.

(ii) $x^2y \cong_x 0$ kann nicht durch Modulo-Reduktion vereinfacht werden.

(iii) $4x + 5y + 10 \cong_3 0 \sim x + 2y + 1 \cong_3 0$.

(iv) $3x + 4 \ncong_3 0 \sim 1 \ncong_3 0 \sim \text{true}$.

Wie aus Beispiel 2.1.10(iii) erkennbar ist, erweitert das Konzept der Modulo-Reduktion nach Lemma 2.1.9 die Modulo-Reduktion für den nicht uniformen Fall. Aus Beispiel 2.1.10(iv) folgt, daß zur Sicherstellung der Idempotenz des Verfahrens der Anwendung von MR die Anwendung von VF nachgeschaltet werden muß.

2.1.4 Inhaltselimination

Der ganzzahlige Inhalt $\mathrm{cont}(t)$ eines Terms $t \in \mathcal{T}$ ist der positive größte gemeinsame Teiler aller ganzzahligen Koeffizienten in einer distributiven Normalform von t. Ein Term mit $\mathrm{cont}(t) = 1$ heißt *inhaltsfrei* oder *primitiv*. Der *primitive Anteil* $\mathrm{prim}(t)$ eines Terms t in distributiver Normalform ist der Term, der aus t entsteht, indem man jeden ganzzahligen Koeffizienten von t durch den Inhalt von t dividiert.

Bekannterweise kann für jede atomare Formel der Form $t\ \rho\ 0$ mit $\rho \in \{=, \neq, <, >, \leq, \geq\}$ aufgrund der Positivität von ggT der Term t stets durch seinen primitiven Anteil ersetzt werden, sodaß gilt

$$t\ \rho\ 0 \sim \mathrm{prim}(t)\ \rho\ 0.$$

Für Kongruenzen bzw. Inkongruenzen der Form $t\ \rho_m\ 0$ mit $m \in \mathbb{Z} \setminus \{0\}$ ist eine solche Umformung nur dann möglich, wenn $\mathrm{ggT}(\mathrm{cont}(t), m) = 1$ ist, denn dann ist aufgrund der Eigenschaften von $(\mathbb{Z}/m, 1, \cdot)$ der Wert $\mathrm{cont}(t)$ invertierbar. Speziell für einen Primmodulus ist die Ersetzung von t durch seinen primitiven Anteil, falls $t \neq 0$, stets möglich. Ist dies nicht der Fall, so läßt sich jedoch für (In-)Kongruenzen der ggT des Term- und des Modulusinhaltes kürzen, wie das folgende Lemma zeigt.

Lemma 2.1.11 Sei $t\ \rho_m\ 0$ eine (In-)Kongruenz und sei $g = \mathrm{ggT}(\mathrm{cont}(t), \mathrm{cont}(m)) > 1$. Dann gilt

$$t\ \rho_m\ 0 \sim \frac{\mathrm{cont}(t)}{g}\mathrm{prim}(t)\ \rho_{\frac{\mathrm{cont}(m)}{g}\mathrm{prim}(m)}\ 0.$$

Beweis: Nach Definition 1.1.5 gilt $\mathbf{P} \vDash (t\ \rho_m\ 0)(\mathbf{z})$ genau dann, wenn

$$\mathrm{cont}(t)(\mathrm{prim}(t))^{\mathbf{P}}(\mathbf{z}) + k\mathrm{cont}(m)(\mathrm{prim}(m))^{\mathbf{P}}(\mathbf{z}) = 0$$

für ein $k \in \mathbb{Z}$. Das Durchteilen mit g liefert unmittelbar die Behauptung für Kongruenzen. Für Inkongruenzen liefert die Anwendung der Äquivalenz $t \not\cong_m 0 \sim \neg t \cong_m 0$ die Behauptung. □

Offenbar ist der Fall der Ersetzung von t durch seinen primitiven Anteil für $\mathrm{ggT}(\mathrm{cont}(t), m) = 1$ *nicht* als Spezialfall in Lemma 2.1.11 enthalten. Man beachte, daß die Auswertung von $\mathrm{prim}(m)$ zu 0 keine Ausnahme für die Anwendung von Lemma 2.1.11 bildet, denn $g \mid \mathrm{cont}(t)$. Die Inhaltselimination für Kongruenzen ist in diesem Fall zu der von Gleichungen und Disgleichungen äquivalent. Das folgende Beispiel zeigt, daß für (In-)Kongruenzen die Überprüfung der Bedingung $\mathrm{ggT}(\mathrm{cont}(t), m) = 1$ für die Erhaltung der Idempotenz erst *nach* der Anwendung Inhaltselimination nach Lemma 2.1.11 durchgeführt werden muß.

Beispiel 2.1.12 Es gilt $4x \cong_{10} 0 \sim 2x \cong_5 0 \sim x \cong_5 0$. Dabei gilt die erstere Äquivalenz nach Lemma 2.1.11 und die zweite aufgrund des Primmodulus 5.

Sind beide Regeln in der gegebenen Reihenfolge angewandt worden, so führt eine erneuerte Anwendung einer der beiden Regeln zu *keinem* neuen Ergebnis. Ist nämlich nach der ersten Anwendung des Lemmas 2.1.11 $\mathrm{ggT}(\mathrm{cont}(t), m) \neq 1$, so sind für eine erneuerte Anwendung des Lemmas 2.1.11 die Voraussetzungen wegen $\mathrm{ggT}(\mathrm{cont}(t), \mathrm{cont}(m)) = 1$ nicht erfüllt. Sonst ist im Falle $\mathrm{ggT}(\mathrm{cont}(t), m) = 1$ nach der Division durch $\mathrm{cont}(t)$ der Term primitiv. Die Voraussetzung für Lemma 2.1.11 ist auch in diesem Fall nicht erfüllt. Für Ungleichungen kann viel mehr gemacht werden.

Lemma 2.1.13 Sei $t\ \rho\ 0$ eine Ungleichung mit $t = t' + a_0$ in distributiver Normalform, wobei a_0 der konstante Term von t ist und $\mathrm{cont}(t') > 1$.

(i) Falls $\rho \in \{>, \leq\}$, dann gilt $t\ \rho\ 0 \sim \mathrm{prim}(t') - \lfloor \frac{-a_0}{\mathrm{cont}(t')} \rfloor\ \rho\ 0$.

(ii) Falls $\rho \in \{<, \geq\}$, dann gilt $t\ \rho\ 0 \sim \mathrm{prim}(t') - \lceil \frac{-a_0}{\mathrm{cont}(t')} \rceil\ \rho\ 0$.

Beweis: Die Aussagen folgen direkt aus den Eigenschaften der Ordnung der ganzen Zahlen. □

Die Ersetzung der Terme durch ihren primitiven Anteil, falls dies möglich ist, und die Anwendung des Lemmas 2.1.13 auf Ungleichungen und des Lemmas 2.1.11 auf Kongruenzen in der oben angegebenen Reihenfolge wird mit CE abgekürzt.

2.1.5 Erfüllbarkeit von Gleichungen und (In-)Kongruenzen

Die Fragestellung nach der Erfüllbarkeit bzw. Allgemeingültigkeit von Gleichungen bzw. Disgleichungen ist in dem eingeführten Kalkül mit der nach der Existenz von ganzzahligen Nullstellen multivariater ganzzahliger Polynome äquivalent (10. Hilbertsche Problem) und somit in dieser Allgemeinheit nicht entscheidbar. Analoges gilt auch für Kongruenzen und Inkongruenzen. In Spezialfällen kann man allerdings nachweisen, daß eine gegebene atomare Formel nicht erfüllbar bzw. allgemeingültig ist.

Eine diophantische Gleichung $a_1x_1 + ... + a_nx_n + a_0 = 0$ mit ganzzahligen Koeffizienten $a_0, ..., a_n$ ist bekanntlich nicht erfüllbar, wenn $\mathrm{ggT}\{a_1, ..., a_n\} \nmid a_0$. Diese Beobachtung liefert eine Kondition für Erfüllbarkeit bzw. Allgemeingültigkeit von Gleichungen bzw. Disgleichungen, die die Aussage über diophantische Gleichungen auf den nicht uniformen Fall fortsetzt.

Lemma 2.1.14 Sei $t + a_0 \in \mathcal{T}$ in distributiver Normalform mit $a_0 \in \mathbb{Z}$ mit $\mathrm{cont}(t) \nmid a_0$. Dann gilt

$$t + a_0 = 0 \sim \mathrm{false} \quad \text{und} \quad t + a_0 \neq 0 \sim \mathrm{true}.$$

Beweis: Seien o.B.d.A. $a_1, ..., a_n \in \mathbb{Z}$ die Koeffizienten und $x_1, ..., x_l$ die Variablen des Terms t, welcher in distributiver Normalform und ohne konstanten Anteil vorliegt. Dann gilt für $t(x_1, ..., x_l)$

$$\{t^{\mathbf{P}}(\mathbf{z}) \mid \mathbf{z} \in \mathbb{Z}^l\} \subseteq \{\sum_{i=1}^{n} a_i u_i \mid u_1, ..., u_n \in \mathbb{Z}\} = \mathrm{Id}(a_1, ..., a_n) = \mathrm{ggT}(a_1, ..., a_n)\mathbb{Z} = \mathrm{cont}(t)\mathbb{Z}.$$

Somit gilt $M = \{(t + a_0)^{\mathbf{P}}(\mathbf{z}) \mid \mathbf{z} \in \mathbb{Z}^l\} \subseteq \mathrm{cont}(t)\mathbb{Z} + a_0$ für $(t + a_0)(x_1, ..., x_l)$. Falls $\mathrm{cont}(t) \nmid a_0$, so gilt für alle $k \in \mathbb{Z}$

$$0 \neq \mathrm{ggT}(a_1, ..., a_n)k + a_0$$

und somit $0 \notin M$. Daraus folgt unmittelbar die Behauptung. □

Die Anwendung des Lemmas 2.1.14 wird mit SE abgekürzt.

Beispiele 2.1.15 Nach Lemma 2.1.14 gelten folgende Äquivalenzen.

(i) $10x^2y + 5y^3x + 7 = 0 \sim \mathrm{false}$.

(ii) $10x^2y + 5y^3x + 7 \neq 0 \sim \mathrm{true}$.

Die Frage nach der Erfüllbarkeit einer Kongruenz $t \cong_m 0$ mit $m \in \mathbb{Z}$ ist mit der nach der Existenz von Werten der Termfunktion von $t(x_1, ..., x_l)$, welche durch durch m teilbar sind, äquivalent. Aus dieser Beobachtung resultiert das folgende Lemma.

Lemma 2.1.16 Seien $t + a_0 \in \mathcal{T}$ in distributiver Normalform mit $a_0, m \in \mathbb{Z}$, sodaß für alle $0 \leq j < m$ gilt

$$m \nmid a_0 + j\mathrm{cont}(t).$$

Dann gilt

$$t + a_0 \cong_m 0 \sim \mathrm{false} \quad \text{und} \quad t + a_0 \ncong_m 0 \sim \mathrm{true}.$$

Beweis: Gelte $\mathcal{V}(t+a_0) = \{x_1, ..., x_l\}$. Aus $m \nmid a_0 + j\mathrm{cont}(t)$ für $0 \leq j < m$ folgt $m \nmid a_0 + j\mathrm{cont}(t)$ für *alle* $j \in \mathbb{Z}$. Denn angenommen es gelte zusätzlich dieser Voraussetzung $m \mid a_0 + k\mathrm{cont}(t)$ für ein $k \in \mathbb{Z} \setminus \{0, ..., m-1\}$ mit etwa $k = lm + k'$ mit $0 \leq k' < m$. Dann folgt aus

$$m \mid k\mathrm{cont}(t) + a_0 = (lm + k')\mathrm{cont}(t) + a_0 = lm\mathrm{cont}(t) + a_0 + k'\mathrm{cont}(t),$$

daß $m \mid a_0 + k'\mathrm{cont}(t)$ für ein $0 \leq k' < m$ gelten muß, was offensichtlich ein Widerspruch ist. Aus Lemma 1.1.6(iii) und Definition 1.1.7 folgt für $(t + a_0)(x_1, ..., x_l)$ weiter $(t + a_0 \cong_m 0)^{\mathbf{P}}(\mathbf{z}) = \bot$ für ein $\mathbf{z} \in \mathbb{Z}^l$ genau dann, wenn $m \nmid t^{\mathbf{P}}(\mathbf{z}) + a_0$. Analog zum Beweis von Lemma 2.1.14 gilt

$$M = \{(t + a_0)^{\mathbf{P}}(\mathbf{z}) \mid \mathbf{z} \in \mathbb{Z}^l\} \subseteq a_0 + \mathrm{cont}(t)\mathbb{Z}.$$

Da nach obiger Überlegung für alle $j \in \mathbb{Z}$ gilt $m \nmid a_0 + j\mathrm{cont}(t)$, ist $km \notin M$ für alle $k \in \mathbb{Z}$. Das liefert unmittelbar die Behauptung für Kongruenzen. Für Inkongruenzen folgt die Behauptung aus der Äquivalenz $t \ncong_m 0 \sim \neg t \cong_m 0$. □

Beispiele 2.1.17 Nach Lemma 2.1.16 gelten folgende Äquivalenzen.

(i) $2x + 2y + 1 \cong_4 0 \sim \mathrm{false}$.

(ii) $2x + 2y + 1 \ncong_4 0 \sim \mathrm{true}$.

Die Anwendung des Lemmas 2.1.16 wird mit SEc abgekürzt.

2.1.6 Idempotenz der Simplifikation

Es ist nicht schwer zu sehen, daß die einzelnen Simplifikationsregeln so definiert sind, daß für jede Regel R außer TC_k für $k > 0$ gilt $\mathrm{R}(\mathrm{R}(\varphi)) = \mathrm{R}$. Die Regel TC_k liefert unter Umständen eine Disjunktion oder Konjunktion von Gleichungen, die mit Simplifikationstechniken aus Abschnitt 2.2 behandelt werden muß. Im Folgenden wird nur die Anwendung von TC_0 betrachtet. Offensichtlich liefern ferner alle Regeln als Ausgabe eine Formel in Normalform. Nun wird eine Reihenfolge für die Anwendung der beschriebenen Simplifikationsregeln angegeben, die ebenfalls einen idempotenten Algorithmus liefert.

Satz 2.1.18 (Simplifikation atomarer Formeln) *Sei $\varphi \in \mathcal{A}$. Die folgende Reihenfolge der Anwendung der Simplifikationsregeln liefert einen terminierenden idempotenten Algorithmus*[1].

(i) $\varphi \leftarrow \mathrm{NF}(\varphi)$*: Die Formel φ ist in Normalform.*

(ii) $\varphi \leftarrow \mathrm{TC}_0(\varphi)$*: Falls das Ergebnis* true *oder* false *ist, breche mit der Ausgabe φ ab.*

(iii) $\varphi \leftarrow \mathrm{VF}(\varphi)$*: Falls das Ergebnis* true *oder* false *ist, breche mit der Ausgabe φ ab.*

(iv) $\varphi \leftarrow \mathrm{ZC}(\varphi)$*: Das Ergebnis ist im Falle der Anwendbarkeit eine* nicht variablenfreie *Gleichung.*

(v) $\varphi \leftarrow \mathrm{DT}(\varphi)$*: Falls das Ergebnis* true *oder* false *ist, breche mit der Ausgabe φ ab.*

(vi) $\varphi \leftarrow \mathrm{MR}(\varphi)$

(vii) $\varphi \leftarrow \mathrm{VF}(\varphi)$*: Falls das Ergebnis* true *oder* false *ist, breche mit der Ausgabe φ ab.*

(viii) $\varphi \leftarrow \mathrm{CE}(\varphi)$

(ix) $\varphi \leftarrow \mathrm{SE}(\varphi)$ *falls φ eine Gleichung ist und* $\varphi \leftarrow \mathrm{SEc}(\varphi)$*, falls φ eine (In-)Kongruenz ist*

(x) Breche mit der Ausgabe φ ab.

[1] Die Textpassagen hinter dem Zeichen „:" sind als Kommentare bzw. Implementierungshinweise aufzufassen.

Beweis: Die Termination ist trivial, denn alle einzelnen Regeln terminieren und es gibt keine Schleifen in der obigen Angabe. Für den Beweis der Idempotenz wird o.B.d.A. angenommen, daß alle Regeln, die false oder true liefern können, bei der *ersten* Anwendung des Algorithmus *nicht* angewandt werden konnten. Somit gelten für die Ausgabe φ' des Algorithmus nach einfacher Anwendung offensichtlich folgende Eigenschaften.

(a) φ' ist weder false noch true.

(b) φ' ist in Normalform und nicht variablenfrei.

(c) Falls φ' eine Gleichung oder Disgleichung $t \ \rho \ 0$ ist, so ist t primitiv und kann nach Lemma 2.1.6 weder als positiv noch als negativ definit erkannt werden.

(d) Falls φ' eine Ungleichung $t + a_0 \ \rho \ 0$ mit $a_0 \in \mathbb{Z}$ ist, so sind t und $t + a_0$ primitiv. Der Term $t + a_0$ kann weiter nach Lemma 2.1.6 weder als positiv noch als negativ definit erkannt werden.

(e) Falls φ' eine (In-)Kongruenz $t \ \rho_m \ 0$ ist, so ist $\mathrm{ggT}\{\mathrm{cont}(t), \mathrm{cont}(m)\} = 1$.

(f) Falls φ' eine (In-)Kongruenz $t \ \rho_m \ 0$ ist mit $m \in \mathbb{Z}$, so ist jeder Koeffizient von t vom Betrag kleiner als m und $m > 1$.

(g) Falls φ' eine (In-)Kongruenz $t \ \rho_m \ 0$ ist, so ist φ' nicht variablenfrei. Weiter gilt $t \neq 0$ und $m \neq 0$.

Betrachtet man nun die *zweite* Anwendung des Algorithmus, so ist es nicht schwer zu sehen, daß *keine* der Voraussetzungen der Regeln aus den Schritten (i)-(ix) erfüllt ist. □

2.2 Simplifikation von Formeln

In diesem Abschnitt werden Simplifikationstechniken für beliebige Presburger-Formeln vorgestellt. Besonders interessant erweist sich dabei neben der Berechnung von Normalformen beliebiger Presburger-Formeln die Simplifikation von stark quantorenfreien Formeln und der gebundenen Quantoren.

2.2.1 Normalformen für Formeln

Die Einführung von gebundenen Quantoren als neue syntaktische Objekte macht sowohl eine Anpassung der Definitionen von Normalformen als auch der Algorithmen zu deren Berechnung notwendig. Für stark quantorenfreie Presburger-Formeln, also für Formeln in denen keine gebundenen Quantoren vorkommen, können die Normalformenalgorithmen zum Beispiel aus [Koe91] wörtlich übernommen werden. Somit ist das Ziel dieses Abschnittes die Ergänzungen der rekursiven Algorithmen für den Fall eines gebundenen Quantors anzugeben.

Definition 2.2.1 (Negationsnormalform) Eine Formel φ ist *positiv* oder in *Negationsnormalform (NNF)*, wenn in φ das Symbol „$\neg$" nicht vorkommt.

Speziell sind die Abkürzungen $\varphi \longrightarrow \psi$, $\varphi \longleftarrow \psi$ und $\varphi \longleftrightarrow \psi$ *nicht* in Negationsnormalform.

Satz 2.2.2 *Jede Presburger-Formel φ besitzt eine Negationsnormalform.*

Beweis: Ein gebundener Quantor ist nach Definition 2.2.1 positiv, wenn sowohl die Matrix als auch der Bound positiv sind. Eine Negation vor einem gebundenen Quantor wird nach Lemma

1.1.14(iii)-(iv) behandelt. Somit ergeben sich für den Algorithmus in [Koe91] auf Seite 42, der hier als NNF bezeichnet wird, folgende zusätzliche Fallunterscheidungen.

$$\mathrm{NNF}(\varphi) = \begin{cases} \ldots \\ \bigwedge_{(\mathrm{NNF}(\psi))(v)} \mathrm{NNF}(\gamma) & \text{falls } \varphi = \bigwedge_{\psi(v)} \gamma, \\ \bigvee_{(\mathrm{NNF}(\psi))(v)} \mathrm{NNF}(\gamma) & \text{falls } \varphi = \bigvee_{\psi(v)} \gamma, \\ \bigvee_{(\mathrm{NNF}(\psi))(v)} \mathrm{NNF}(\neg\gamma) & \text{falls } \varphi = \neg \bigwedge_{\psi(v)} \gamma \text{ und} \\ \bigwedge_{(\mathrm{NNF}(\psi))(v)} \mathrm{NNF}(\neg\gamma) & \text{falls } \varphi = \neg \bigvee_{\psi(v)} \gamma. \ \square \end{cases}$$

Obwohl jede linear quantifizierte Presburger-Formel aufgrund der Existenz eines Quantoreneliminationsverfahrens eine pränexe Normalform (PNF) besitzt, ist die Konstruktion einer solchen, ohne Quantorenelimination explizit anzuwenden, sehr aufwendig. Speziell für Formeln der Form

$$\bigvee_{\psi(v)} \forall x \varphi \quad \text{bzw.} \quad \bigwedge_{\psi(v)} \exists x \varphi$$

ist die Bestimmung einer PNF aufgrund der im Allgemeinen von Parametern abhängigen Erfüllungsmenge von ψ problematisch. Aus diesem Grund wird die Definition einer PNF dahingehend modifiziert, daß auch gebundene Quantoren in Quantorenblöcken auftreten dürfen. Ein existenzieller bzw. universeller *Quantorenblock* ist dann von der Form

$$\exists x_1 \exists x_2 ... \exists x_n \quad \text{oder} \quad \bigvee_{\psi_1(v_1)} \bigvee_{\psi_2(v_2)} ... \bigvee_{\psi_n(v_n)}$$

bzw.

$$\forall x_1 \forall x_2 ... \forall x_n \quad \text{oder} \quad \bigwedge_{\psi_1(v_1)} \bigwedge_{\psi_2(v_2)} ... \bigwedge_{\psi_n(v_n)} .$$

Definition 2.2.3 (Pränexe Normalform) Eine Formel φ in NNF heißt in *pränexer Normalform (PNF)*, wenn φ von der Form

$$B_1 B_2 ... B_n \psi$$

ist, wobei jedes B_i entweder ein existenzieller oder ein universeller Quantorenblock ist und ψ stark quantorenfrei ist. Die Formel ψ heißt dann die *Matrix* von φ.

Satz 2.2.4 *Jede Presburger-Formel φ besitzt eine pränexe Normalform nach Definition 2.2.3. Für die Konstruktion ist eine Anwendung der Quantorenelimination nicht erforderlich.*

Beweis: Es reicht für den Beweis anzumerken, daß der Algorithmus aus [Koe91] auf Seite 44 so modifiziert wird, daß dieser gebundene Quantoren, wie gewöhnliche Quantoren nach Lemma 1.1.14(i)-(ii) behandelt. $\square$

Man kann sich das Ziel setzen konjunktive und disjunktive Normalformen mit gebundenen Quantoren so einzuführen, daß das Ersetzen von gebundenen Quantoren durch ihre Expansionen für feste Werte der Bound-Parameter im Ergebnis die entsprechende Normalform im gewöhnlichen Sinne darstellt.

Definition 2.2.5 (Konjunktive und Disjunktive Normalform) Sei φ eine schwach quantorenfreie Presburger-Formel.

(i) Die Formel φ ist in *disjunktiver Normalform (DNF)*, falls φ von der Form

$$\bigvee_{i=1}^{k} \bigvee_{\psi_1(v_1)} ... \bigvee_{\psi_n(v_n)} \bigwedge_{\gamma_1(u_1)} ... \bigwedge_{\gamma_m(u_m)} \beta_i$$

ist, wobei jedes β_i eine stark quantorenfreie Konjunktion atomarer Formeln ist.

(ii) Die Formel φ ist in *konjunktiver Normalform (KNF)*, falls φ von der Form

$$\bigwedge_{i=1}^{k} \bigwedge_{\psi_1(v_1)} \cdots \bigwedge_{\psi_n(v_n)} \bigvee_{\gamma_1(u_1)} \cdots \bigvee_{\gamma_m(u_m)} \beta_i$$

ist, wobei jedes β_i eine stark quantorenfreie Disjunktion atomarer Formeln ist.

In obiger Definition werden atomare Formeln als triviale Disjunktionen und Konjunktionen angesehen. Der Algorithmus, der einer stark quantorenfreien Formel eine DNF bzw. KNF von der Form

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{k} \psi_{ij} \quad \text{bzw.} \quad \bigwedge_{i=1}^{n} \bigvee_{j=1}^{k} \psi_{ij}$$

zuordnet, wobei jede Formel $\psi_{i,j}$ für $1 \leq i \leq n$ und $1 \leq j \leq k$ eine atomare Formel ist, kann [Koe91] auf Seite 44 ff. entnommen werden. Für eine schwach quantorenfreie Formel mit nichtparametrischen Bounds kann man stets eine DNF bzw. KNF der Formel angeben, denn man kann die gebundenen Quantoren durch ihre Expansionen ersetzen. Für Formeln mit parametrischen Bounds ist die Frage nach einem konstruktiven Verfahren für die Bestimmung einer DNF bzw. KNF von Formeln der Form

$$\bigwedge_{\psi(v)} (\varphi_1 \vee \varphi_2) \quad \text{bzw.} \quad \bigvee_{\psi(v)} (\varphi_1 \wedge \varphi_2)$$

offen. Aus diesem Grund wird statt einer DNF bzw. KNF nach Definition 2.2.5 in der Implementierung eine Pseudo-DNF bzw. Pseudo-KNF einer schwach quantorenfreien Formel bestimmt, die pränex ist und deren stark quantorenfreie Matrix in DNF bzw. KNF ist. Diese wird absichtlich *nicht* zur Definition einer DNF bzw. KNF verwendet, denn Formeln in der so eingeführten Pseudo-DNF bzw. Pseudo-KNF besitzen *nicht* die typischen Eigenschaften einer DNF bzw. KNF.

Beispiel 2.2.6 Betrachte die schwach quantorenfreie Formel in Pseudo-DNF

$$\varphi = \bigwedge_{(0 \leq k \leq 1)(k)} (x = k \vee x = k + 1).$$

Offenbar kann die Erfüllungsmenge von $\varphi(x)$ *nicht* etwa aus der Vereinigung von Erfüllungsmengen von $(\bigwedge_{(0 \leq k \leq 1)(k)} x = k)(x)$ und $(\bigwedge_{(0 \leq k \leq 1)(k)} x = k + 1)(x)$ gebildet werden.

2.2.2 Theoriesimplifikation von stark quantorenfreien Formeln

In diesem Abschnitt wird anlehnend an [DS97b] die Theoriesimplifikation von stark quantorenfreien Formeln behandelt, wobei die Applikation der in [DS97b] vorgestellten Techniken im Vordergrund stehen soll. Bis zum Ende dieses Abschnittes wird unter „einer Formel" stets eine stark quantorenfreie Formel verstanden. Weiter sei für diesen Abschnitt stets vorausgesetzt, daß alle atomaren Formeln in der Eingabe in vereinfachter Form aus Abschnitt 2.1 vorliegen.

Als eine *Theorie* Φ wird in dieser Arbeit eine endliche Menge von atomaren Formeln bezeichnet. Eine Theorie heißt *widersprüchlich*, falls gilt

$$\mathbf{P} \vDash \bigwedge_{\psi \in \Phi} \psi \longrightarrow \text{false}.$$

Zwei Formeln φ und φ' heißt äquivalent bezüglich der Theorie Φ, wenn gilt

$$\mathbf{P} \vDash \bigwedge_{\psi \in \Phi} \psi \longrightarrow (\varphi \longleftrightarrow \varphi').$$

Dies wird mit $\Phi \vDash \varphi \longleftrightarrow \varphi'$ abgekürzt. Theoriesimplifikation basiert auf dem folgenden zentralen Lemma.

Lemma 2.2.7 Sei Φ eine Theorie, γ eine atomare Formel und sei φ eine Formel. Bezeichne mit γ' eine zu $\neg\gamma$ äquivalente atomare Formel.

(i) Falls $\Phi \cup \{\gamma\} \models \varphi \longleftrightarrow \varphi'$ für ein φ' gilt, dann gilt

$$\Phi \models \gamma \wedge \varphi \longleftrightarrow \gamma \wedge \varphi'.$$

(ii) Falls $\Phi \cup \{\gamma'\} \models \varphi \longleftrightarrow \varphi'$ für ein φ' gilt, dann gilt

$$\Phi \models \gamma \vee \varphi \longleftrightarrow \gamma \vee \varphi'.$$

Beweis: (i) Es gilt offenbar

$$\begin{aligned}(\gamma \wedge \bigwedge_{\psi\in\Phi} \psi) \longrightarrow (\varphi \longleftrightarrow \varphi') &\sim \bigwedge_{\psi\in\Phi} \psi \longrightarrow \neg\gamma \vee (\varphi \longleftrightarrow \varphi') \\ &\sim \bigwedge_{\psi\in\Phi} \psi \longrightarrow (\gamma \wedge \varphi \longleftrightarrow \gamma \wedge \varphi').\end{aligned}$$

(ii) Es gilt analog

$$\begin{aligned}(\neg\gamma \wedge \bigwedge_{\psi\in\Phi} \psi) \longrightarrow (\varphi \longleftrightarrow \varphi') &\sim \bigwedge_{\psi\in\Phi} \psi \longrightarrow \gamma \vee (\varphi \longleftrightarrow \varphi') \\ &\sim \bigwedge_{\psi\in\Phi} \psi \longrightarrow (\gamma \vee \varphi \longleftrightarrow \gamma \vee \varphi').\Box\end{aligned}$$

Das obige Lemma findet Anwendung, indem man während des rekursiven Abstiegs in die Formel eine Theorie, genannt *implizite Theorie*, anreichert. Diese wird um konjunktiv Verknüpfte atomare Formeln und um negiert äquivalente disjunktiv verknüpfte atomare Formeln vervollständigt und anschließend nach Möglichkeit vereinfacht. Dabei wird die Teilformel, aus der die atomare Formel stammt, durch eine Markierung, wie zum Beispiel die Tiefe der atomaren Formel, gespeichert. Das Lemma 2.2.7 liefert dann den Nachweis dafür, daß beim Wiederaufstieg die ursprünglichen Teilformeln durch ihre unter der angereicherten Theorie vereinfachten Äquivalente, die aus der Theorie mit Hilfe der Markierungen entnommen werden können, ersetzt werden dürfen.

Offensichtlich ist es nicht notwendig, daß die Theorie, mit der die Rekursion anfängt, leer ist. Eine beliebige Theorie Φ_0, genannt *explizite Theorie*, kann als Parameter zur Simplifikation verwendet werden. Das Ergebnis der Simplifikation φ' von φ ist dann gültig bezüglich Φ_0

$$\Phi_0 \models \varphi \longleftrightarrow \varphi'.$$

Beispiel 2.2.8 In diesem Beispiel werden, um Anschaulichkeit zu bewahren, atomare Formeln samt ihrer Tiefe als Markierung in Form von Tupeln in die Theorie eingetragen. Zu simplifizieren ist die Formel

$$\varphi = a \leq 0 \wedge (b \leq 0 \vee (a \geq 0 \wedge b = 0)).$$

Sei $\Phi_0 = \emptyset$. Die Formel $a \leq 0$ kann mit der Markierung 0 in die Theorie aufgenommen werden mit dem Ergebnis $\Phi_1 = \{(a \leq 0, 0)\}$. Nun wird die Teilformel $b \leq 0 \vee (a \leq 0 \wedge b = 0)$ mit Φ_1 simplifiziert. Bei der Aufnahme von $b > 0$, da $b \leq 0$ negiert aufgenommen werden muß, mit der Markierung 1 kann die Theorie nicht vereinfacht werden. Somit ergibt sich $\Phi_2 = \{(a \leq 0, 0), (b > 0, 1)\}$. Nun wird die Teilformel $a \geq 0 \wedge b = 0$ mit Φ_2 simplifiziert. Bei der Aufnahme von $a \geq 0$ mit der Markierung 2 ergibt sich

$$\Phi_2 \models a \geq 0 \longleftrightarrow a = 0.$$

Somit wird $a \leq 0$ durch $a = 0$ mit der Markierung 2 in der Theorie mit dem Ergebnis $\Phi_3 = \{(b > 0, 1), (a = 0, 2)\}$ ersetzt. Bei der Simplifikation von $b = 0$ mit Φ_3 ergibt sich allerdings

$$\Phi_3 \vDash b = 0 \longleftrightarrow \text{false}.$$

Somit ist die Theorie $\Phi_3 \cup \{(b = 0, 3)\}$ widersprüchlich. Beim Wiederaufstieg ergibt sich aus $a \leq 0 \wedge (b \leq 0 \vee (a = 0 \wedge \text{false}))$

$$\varphi' = a \leq 0 \wedge b \leq 0 \sim \varphi.$$

Würde man zum Beispiel mit $\Phi_0 = \{(a \geq 0, -1), (b \geq 0, -1)\}$ beginnen, so wäre das Ergebnis

$$\Phi_0 \vDash \varphi \longleftrightarrow (a = 0 \wedge b = 0).$$

Zur Spezifikation der Simplifikation mit Theorie wird eine Regel zur Aufnahme einer beliebigen atomaren Formel ψ in die Theorie Φ angegeben. Dies wird durch einen Vergleich von ψ mit jeder anderen Formel aus Φ realisiert. Analoges muß bei der Simplifikation der Theorie nach der Aufnahme einer Formel durchgeführt werden, da eine Veränderung der Theorie weitere Vereinfachungen nach sich ziehen kann. Es ist also zweckmäßig anzugeben, wie der Vergleich einer atomaren Formel $\varphi \in \Phi$ und einer atomaren Formel ψ, die zu dieser hinzugefügt werden soll, stattfindet. Dabei entstehen folgende Alternativen.

- Es findet keine Vereinfachung statt. Falls dieses Ergebnis für jede andere Formel aus Φ festgestellt wird, wird ψ zu Φ hinzugefügt.

false Die beiden Formeln widersprechen sich ($\mathbf{P} \vDash \varphi \wedge \psi \longleftrightarrow \text{false}$). Damit ist $\Phi \cup \{\psi\}$ widersprüchlich.

N Die Formel φ in der Theorie kann durch die neu entdeckte ersetzt werden ($\mathbf{P} \vDash \varphi \wedge \psi \longleftrightarrow \psi$).

O Die neue Formel muß nicht hinzugefügt werden ($\mathbf{P} \vDash \varphi \wedge \psi \longleftrightarrow \varphi$).

Nφ' Die Formel in der Theorie wird durch φ' ersetzt und die neue hinzugefügt ($\mathbf{P} \vDash \varphi \wedge \psi \longleftrightarrow \varphi' \wedge \psi$).

Oψ' Die Formel ψ' wird statt der neuen Formel ψ in die Theorie aufgenommen ($\mathbf{P} \vDash \varphi \wedge \psi \longleftrightarrow \varphi \wedge \psi'$).

Dγ Die Formel φ wird in der Theorie durch γ ersetzt und die neue Formel wird nicht hinzugefügt ($\mathbf{P} \vDash \varphi \wedge \psi \longleftrightarrow \gamma$).

Anhand der Beschaffenheit von φ und ψ wird die Bestimmung der zu durchführenden Aktion in folgende Fälle getrennt.

(i) Die atomaren Formeln besitzen identische Terme, aber unterschiedliche Relationszeichen.

(ii) Die atomaren Formeln sind (In-)Kongruenzen $\varphi = t\ \rho_n\ 0$ und $\psi = t\ \rho_m\ 0$ mit identischen Termen aber mit unterschiedlichen Moduli.

(iii) Die atomaren Formeln besitzen Terme, die sich um einen konstanten Term unterscheiden und unterschiedliche Relationszeichen aus $\{=, \neq, <, >, \leq, \geq\}$.

(iv) Die atomaren Formeln sind (In-)Kongruenzen, deren Moduli gleich sind und die Terme sich um einen konstanten Term unterscheiden.

(v) Die atomaren Formeln sind (In-)Kongruenzen, deren Moduli verschieden sind und die Terme sich um einen konstanten Term unterscheiden.

In den folgenden Ausführungen wird nur der Fall behandelt, daß die Moduli der vorkommenden Kongruenzen von 0 verschiedene natürliche Zahlen sind. Aufgrund der Annahme, daß alle atomaren Formeln in vereinfachter Form vorliegen, ist es nicht notwendig ganzzahlige Kongruenzen zu betrachten. Parametrische Moduli werden hier nicht betrachtet.

Identische Terme und verschiedene Relationszeichen

Seien $t, m \in \mathcal{T}$ und $\varphi = t\ \rho_1\ 0$ bzw. $\varphi = \rho_1(t, 0, m)$ und $\psi = t\ \rho_2\ 0$ bzw. $\psi = \rho_2(t, 0, m)$ mit $m \in \mathbb{N} \setminus \{0\}$. In diesem Fall kann die Aktion der Tabelle 2.1 entnommen werden.

				ρ_1				
ρ_2	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$	$\cong$	$\ncong$
$<$	O	N	false	N	false	false	-	-
$\leq$	O	O	O	$D_{t<0}$	$D_{t=0}$	false	-	$O_{t<0}$
$=$	false	N	O	false	N	false	N	false
$\neq$	O	$D_{t<0}$	false	O	$D_{t>0}$	O	-	O
$\geq$	false	$D_{t=0}$	O	$D_{t>0}$	O	O	-	$O_{t>0}$
$>$	false	false	false	N	N	O	-	-
$\cong$	-	-	O	-	-	-	O	false
$\ncong$	-	$N_{t<0}$	false	N	$N_{t>0}$	-	false	O

Tabelle 2.1: Identische Terme und verschiedene Relationszeichen

Die Korrektheit der Ersetzungsregeln ergibt sich für Gleichungen und Ungleichungen aus den Eigenschaften der Ordnung in $\mathbb{Z}$. Für Kongruenzen und Gleichungen und Disgleichungen folgt die Korrektheit aus Lemma 2.2.9. Ebenfalls aus Lemma 2.2.9 folgt auch die Korrektheit für Kongruenzen und Ordnungsrelationen mit Hilfe der Äquivalenzen $t \leq 0 \sim t < 0 \vee t = 0$ und $t \geq 0 \sim t > 0 \vee t = 0$.

Lemma 2.2.9 Seien $m, t \in \mathcal{T}$. Dann gelten folgende Äquivalenzen.

(i) $t = 0 \wedge t \cong_m 0 \sim t = 0$.

(ii) $t = 0 \wedge t \ncong_m 0 \sim$ false.

(iii) $t \cong_m 0 \wedge t \ncong_m 0 \sim$ false.

(iv) $t \neq 0 \wedge t \ncong_m 0 \sim t \ncong_m 0$.

Beweis: Der Beweis wird exemplarisch für (i) durchgeführt. Die restlichen Fälle beweist man analog. (i) Gelte zunächst $\mathbf{P} \vDash (t = 0)(\mathbf{z})$ und somit $t^{\mathbf{P}}(\mathbf{z}) = 0$. Betrachte die in $\mathbb{Z}$ äquivalente Gleichung $t^{\mathbf{P}}(\mathbf{z}) + m^{\mathbf{P}}(\mathbf{z})0 = 0$, woraus unmittelbar auch $\mathbf{P} \vDash (t \cong_m 0)(\mathbf{z})$ folgt. Nach der Definition der Interpretation von Formeln gilt $\mathbf{P} \vDash (t = 0 \wedge t \cong_m 0)(\mathbf{z})$. Die Umkehrrichtung ist aufgrund der Interpretation von Konjunktionen offensichtlich. □

Kongruenzen mit identischen Termen und verschiedenen Moduli

Seien $\rho_1, \rho_2 \in \{\cong, \ncong\}$ und $\varphi = \rho_1(t, 0, n)$ und $\psi = \rho_2(t, 0, m)$ mit $n, m \in \mathbb{N} \setminus \{0\}$. Unter diesen Annahmen kann die Aktion der Tabelle 2.2 entnommen werden.

Dabei erhält man m' wiefolgt aus m und n. Seien

$$n = p_1^{D_1} ... p_k^{D_k} q_1^{e_1} ... q_l^{e_l} r$$

$$m = p_1^{d_1} ... p_k^{d_k} q_1^{E_1} ... q_l^{E_l} s$$

die Faktorzerlegungen von m und n mit D_i,d_i,E_j,$e_j > 0$ und p_i, q_j Primzahlen, die in r und s nicht vorkommen für $1 \leq i \leq k$ und $1 \leq j \leq l$. Gelte weiterhin $D_i \geq d_i$ und $E_j > e_j$ für $1 \leq i \leq k$ und $1 \leq j \leq l$. Dann ist m' definiert durch

$$m' = q_1^{E_1} ... q_l^{E_l} s.$$

Analog erhält man n' aus m und n durch Vertauschen von n und m in den obigen Überlegungen. Die Korrektheit der Ersetzungen ergibt sich aus Lemma 2.2.10.

ρ_1	ρ_2	Aktion
$\cong$	$\cong$	$\mathrm{D}_{t\cong_{\mathrm{kgV}(n,m)}0}$
$\cong$	$\ncong$	$\begin{cases}\mathrm{D}_{t\cong_{2n}n} & \text{falls } m=2n \text{ und}\\ \mathrm{O}_{t\ncong_{m'}0} & \text{sonst.}\end{cases}$
$\ncong$	$\cong$	$\begin{cases}\mathrm{D}_{t\cong_{2m}m} & \text{falls } n=2m \text{ und}\\ \mathrm{N}_{t\ncong_{n'}0} & \text{sonst.}\end{cases}$
$\ncong$	$\ncong$	$\begin{cases}\mathrm{O} & \text{falls } n\mid m,\\ \mathrm{N} & \text{falls } m\mid n \text{ und}\\ - & \text{sonst.}\end{cases}$

Tabelle 2.2: Kongruenzen mit identischen Termen und verschiedenen Moduli

Lemma 2.2.10 Seien $m,n\in\mathbb{N}\setminus\{0\}$ und $t\in\mathcal{T}$. Dann gilt:

(i) $t\cong_m 0\wedge t\cong_n 0\sim t\cong_{\mathrm{kgV}(n,m)} 0$.

(ii) $t\cong_n 0\wedge t\ncong_{2n} 0\sim t\cong_{2n} n$.

(iii) Falls $m'\in\mathbb{Z}$ wie oben definiert ist, so gilt $t\cong_n 0\wedge t\ncong_m 0\sim t\cong_n 0\wedge t\ncong_{m'} 0$.

(iv) Falls $n\mid m$, so gilt $t\ncong_n 0\wedge t\ncong_m 0\sim t\ncong_n 0$.

Beweis: (i) Die Behauptung ist trivial. (ii) Es gilt

$$t\cong_n 0\wedge t\ncong_{2n} 0\sim t\cong_n 0\wedge\bigvee_{i=1}^{2n-1} t\cong_{2n} i\sim\bigvee_{i=1}^{2n-1}(t\cong_n 0\wedge t\cong_{2n} i).$$

In obiger Disjunktion ist nur das Disjunktionsglied $2t\cong_n 0\wedge t\cong_{2n} n$ nicht äquivalent zu false, denn aus $n\mid t$ und $2n\mid t-i$ folgt $kn=t$ und $2nl=t-i$ und somit $n(k-2l)=i$ für $k,l\in\mathbb{Z}$. Daraus folgt $n\mid i$ und mit der Voraussetzung $1\le i\le 2n-1$ auch $n=i$. Daher gilt weiter

$$\bigvee_{i=1}^{2n-1}(t\cong_n 0\wedge t\cong_{2n} i)\sim t\cong_n 0\wedge t\cong_{2n} n\sim t\cong_{2n} n.$$

(iii) Betrachte die oben definierten Faktorzerlegungen von n, m und m'. Dann gilt

$$\begin{aligned}
t\cong_n 0\wedge t\ncong_m 0 \;&\sim\; t\cong_r 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0\wedge\bigwedge_{i=1}^{l} t\cong_{q_i^{e_i}} 0\wedge(t\ncong_s 0\vee\bigvee_{i=1}^{k} t\ncong_{p_i^{d_i}} 0\vee\bigvee_{i=1}^{l} t\ncong_{q_i^{E_i}} 0)\\
&\sim\; t\cong_r 0\wedge\bigwedge_{i=1}^{l} t\cong_{q_i^{e_i}} 0\wedge((t\ncong_s 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0)\\
&\quad\vee(\bigvee_{i=1}^{k} t\ncong_{p_i^{d_i}} 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0)\vee(\bigvee_{i=1}^{l} t\ncong_{q_i^{E_i}} 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0))\\
&\sim\; t\cong_r 0\wedge\bigwedge_{i=1}^{l} t\cong_{q_i^{e_i}} 0\wedge((t\ncong_s 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0)\\
&\quad\vee(\bigvee_{j=1}^{k}\underbrace{\bigwedge_{i=1}^{k}(t\ncong_{p_j^{d_j}} 0\wedge t\cong_{p_i^{D_i}} 0)}_{\sim\text{false wegen } p_i<D_i})\vee(\bigvee_{i=1}^{l} t\ncong_{q_i^{E_i}} 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0))\\
&\sim\; t\cong_r 0\wedge\bigwedge_{i=1}^{k} t\cong_{p_i^{D_i}} 0\wedge\bigwedge_{i=1}^{l} t\cong_{q_i^{e_i}} 0\wedge(t\ncong_s 0\vee\bigvee_{i=1}^{l} t\ncong_{q_i^{E_i}} 0)\\
&\sim\; t\cong_n 0\wedge t\ncong_{m'} 0.
\end{aligned}$$

(iv) Die Behauptung ist trivial. □

(Dis-)Gleichungen und Ungleichungen mit verschiedenen konstanten Anteilen

Nun wird der Fall behandelt, bei dem sich die Terme der atomaren Formeln lediglich in ihren konstanten Anteilen unterscheiden. Sei $\varphi = t - a \; \rho_1 \; 0$ und $\psi = t - b \; \rho_2 \; 0$ mit $a, b \in \mathbb{Z}$. Sei ferner $a < b$. Dann kann man die gesuchte Aktion aus der Tabelle 2.3.

					ρ_1	
ρ_2	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$
$<$	O	O	O	-	$\begin{cases} \mathrm{D}_{t=a} & \text{falls } a+1=b \text{ und} \\ - & \text{sonst.} \end{cases}$	$\begin{cases} \text{false} & \text{falls } a+1=b \text{ und} \\ - & \text{sonst.} \end{cases}$
$\leq$	O	O	O	-	-	$\begin{cases} \mathrm{D}_{t=b} & \text{falls } a+1=b \text{ und} \\ - & \text{sonst.} \end{cases}$
$=$	false	false	false	N	N	N
$\neq$	O	O	O	-	-	-
$\geq$	false	false	false	N	N	N
$>$	false	false	false	N	N	N

Tabelle 2.3: Ordnungsrelationen mit verschiedenen konstanten Anteilen für $a < b$

Für $a > b$ ergibt sich analog die Tabelle 2.4

			ρ_1			
ρ_2	$<$	$\leq$	$=$	$\neq$	$\geq$	$>$
$<$	N	N	false	N	false	false
$\leq$	N	N	false	N	false	false
$=$	N	N	false	N	false	false
$\neq$	-	-	O	-	O	O
$\geq$	$\begin{cases} \mathrm{D}_{t=b} & \text{falls } b+1=a \text{ und} \\ - & \text{sonst.} \end{cases}$	-	O	-	O	O
$>$	$\begin{cases} \text{false} & \text{falls } b+1=a \text{ und} \\ - & \text{sonst.} \end{cases}$	$\begin{cases} \mathrm{D}_{t=a} & \text{falls } b+1=a \text{ und} \\ - & \text{sonst.} \end{cases}$	O	-	O	O

Tabelle 2.4: Ordnungsrelationen mit verschiedenen konstanten Anteilen für $a > b$

An dieser Stelle wird angemerkt, daß der Fall $a = b$ ist bereits behandelt worden ist. Die Korrektheit der Ersetzungsregeln ergibt sich aus den Eigenschaften der Ordnung in $\mathbb{Z}$.

(In-)Kongruenzen mit verschiedenen konstanten Anteilen

Die Aktionen zur Simplifikation von (In-)Kongruenzen werden von den Ordnungsrelationen getrennt behandelt. Seien hierzu $\rho_1, \rho_2 \in \{\cong, \ncong\}$. Seien wieder $\varphi = \rho_1(t - a, 0, n)$ und $\psi = \rho_2(t - b, 0, n)$ mit $a, b \in \mathbb{Z}$ und $n \in \mathbb{N} \setminus \{0\}$. Da die atomaren Formeln in vereinfachter Form vorliegen, kann man für a und b folgern, daß $0 \leq a < n$, $0 \leq b < n$ und $a \neq b$ gilt. Daraus folgt eine für die Simplifikation wichtige Folgerung

$$\mathbf{P} \vDash a \ncong_n b.$$

Unter diesen Annahmen, aus welchen auch direkt die Korrektheit der Ersetzungen folgt, kann man die Simplifikation durch die Tabelle 2.5 darstellen.

ρ_1	ρ_2	Aktion
$\cong$	$\cong$	false
$\cong$	$\ncong$	O
$\ncong$	$\cong$	N
$\ncong$	$\ncong$	-

Tabelle 2.5: (In-)Kongruenzen mit verschiedenen konstanten Anteilen und gleichen Moduli

Seien nun $\varphi = t - a \cong_n 0$ und $\psi = t - b \cong_m 0$ mit $a, b \in \mathbb{Z}$ und $n, m \in \mathbb{N} \setminus \{0\}$. Für den Fall, daß n und m teilerfremd sind ergibt sich mit Hilfe des chinesischen Restklassensatzes die Aktion $\mathrm{D}_{t' \cong_{mn} 0}$ nach Lemma 2.2.11.

Lemma 2.2.11 Seien $\varphi = t - a \cong_n 0$ und $\psi = t - b \cong_m 0$ mit $a, b \in \mathbb{Z}$ und $n, m \in \mathbb{N} \setminus \{0\}$, wobei n und m teilerfremd sind. Dann gilt

$$\varphi \wedge \psi \sim t \cong_{mn} an'm + bm'n,$$

wobei $n' = m^{-1} (\mathrm{mod}\ n)$ und $m' = n^{-1} (\mathrm{mod}\ m)$.

Beweis: Der Beweis ergibt sich unmittelbar aus dem chinesischen Restklassensatz. □

Die Inversen modulo n und m können algorithmisch leicht mit Hilfe des erweiterten euklidischen Algorithmus bestimmt werden. Dies wird nun ein einem einfachen Beispiel veranschaulicht.

Beispiel 2.2.12 Sei $\varphi = x \cong_6 3$ und $\psi = x \cong_5 2$. Da $\mathrm{ggT}(6, 5) = 1$ ergibt sich mit Hilfe des erweiterten euklidischen Algorithmus eine Darstellung von 1 als Linearkombination von 5 und 6 durch

$$\mathrm{ggT}(6, 5) = 1 = 1 \cdot 6 - 1 \cdot 5.$$

Somit gilt $n' = -1 (\mathrm{mod}\ 5)$ und $m' = 1 (\mathrm{mod}\ 6)$. Daraus ergibt sich nach Lemma 2.2.11 die Äquivalenz

$$x \cong_6 3 \wedge x \cong_5 2 \sim x \cong_{30} -3$$

2.2.3 Simplifikation von gebundenen Quantoren

Eine Formel φ wird als vereinfacht verstanden, wenn alle Teilformeln von φ, unter anderem auch alle atomaren Formeln, in vereinfachter Form vorliegen. Diese Überlegung bietet einen direkten Zugang zur Simplifikation von beliebigen Formeln mit Theorie. Dazu reicht es intuitiv aus algorithmischer Sicht rekursiv in die Formel abzusteigen und alle vorkommenden Teilformeln zu vereinfachen. Eine noch nicht behandelte Ausnahme bei dieser Strategie stellen gebundene Quantoren dar. Es stellt sich heraus, daß zusätzlich zur Auswertung trivialer Bounds und Bereiche von gebundenen Quantoren die Theorie in bestimmten Fällen mit atomaren Formeln aus dem Bound angereichert werden kann.

Lemma 2.2.13 (Triviale gebundene Quantoren) Es gelten folgende Äquivalenzen.

(i) $\bigwedge_{\psi(v)} \varphi \sim \mathrm{true}$, falls $\psi \sim \mathrm{false}$.

(ii) $\bigvee_{\psi(v)} \varphi \sim \mathrm{false}$, falls $\psi \sim \mathrm{false}$.

(iii) $\bigwedge_{\psi(v)} \varphi \sim \varphi$, falls $v \notin \mathcal{V}(\varphi)$, $\mathcal{V}(\psi) = \{v\}$ und $\psi \not\sim \mathrm{false}$.

(iv) $\bigvee_{\psi(v)} \varphi \sim \varphi$, falls $v \notin \mathcal{V}(\varphi)$, $\mathcal{V}(\psi) = \{v\}$ und $\psi \not\sim \mathrm{false}$.

(v) $\bigvee_{(v=t)(v)} \varphi \sim \varphi[t/v]$ für $t \in \mathcal{T}$.

Beweis: Der Beweis wird im Wesentlichen durch Anwendung des Lemmas 1.1.14 und durch elementare logische Operationen erbracht. (i) $\bigwedge_{\psi(v)} \varphi \sim \forall v(\text{false} \longrightarrow \varphi) \sim \forall v(\text{true}) \sim \text{true}$. (ii) Dieser Fall wird analog zu (i) gezeigt. (iii) $\bigwedge_{\psi(v)} \varphi \sim \forall v(\psi \longrightarrow \varphi) \sim \forall v(\neg\psi \vee \varphi)$. Daraus folgt wegen $v \notin \mathcal{V}(\varphi)$

$$\forall v(\neg\psi \vee \varphi) \sim \varphi \vee (\forall v \neg\psi).$$

Mit $\psi \not\sim \text{false}$ folgt wegen $\forall v \neg\psi \sim \text{false}$ unmittelbar die Behauptung. (iv) Der Fall wird analog zu (iii) gezeigt. (v) Die Behauptung folgt unmittelbar aus der Interpretation eines gebundenen Quantors. □

Es ist an dieser Stelle sinnvoll zu bemerken, daß die Fragestellung nach der Erfüllbarkeit vom Bound ψ mit der Bound-Variable v eines gebundenen Quantors in dem hier eingeführten Kalkül aufgrund möglicher nichtlinearer Bound-Parameter im Allgemeinen nicht entscheidbar ist. Da Simplifikation aufgrund praktischer Untersuchungen einer der am häufigsten aufgerufenen Algorithmen in REDLOG ist, ist es nicht empfehlenswert auch für ausschließlich lineare Parameter zum Zwecke der Vereinfachung von gebundenen Quantoren die Quantorenelimination aufzurufen. Sogar im Falle eines Bounds mit der leeren Menge der Bound-Parameter ist es aufgrund möglicherweise sehr großer Erfüllungsmengen des Bounds bezüglich der Bound-Variable *nicht empfehlenswert* diese für den Test auf Erfüllbarkeit explizit auszurechnen.

Neben diesen einfachen Simplifikationsregeln kann man ausgezeichnete Formeln in den Bound eines gebundenen Quantors übertragen. Da die neuen Teilformeln konjunktiv zum Bound hinzugefügt werden, wird die Erfüllungsmenge des Bounds bezüglich der Boundvariablen lediglich kleiner und somit immer noch endlich. Aufgrund dieser Beobachtung wird der oben skizzierte Vorgang als Bound-Verstärkung bezeichnet. Das folgende Lemma präzisiert die oben dargestellte Aussage.

Lemma 2.2.14 (Bound-Verstärkung) Sei γ eine stark quantorenfreie Formel mit $\{v\} = \mathcal{V}(\gamma)$. Es gelten dann folgende Äquivalenzen.

(i) $\bigwedge_{\psi(v)}(\gamma \vee \varphi) \sim \bigwedge_{(\psi \wedge \neg\gamma)(v)} \varphi$.

(ii) $\bigvee_{\psi(v)}(\gamma \wedge \varphi) \sim \bigvee_{(\psi \wedge \gamma)(v)} \varphi$.

Beweis: (i) Es gilt offenbar nach Lemma 1.1.14(ii)

$$\bigwedge_{\psi(v)} (\gamma \vee \varphi) \sim \forall v(\psi \longrightarrow (\gamma \vee \varphi)) \sim \forall v(\neg(\psi \wedge \neg\gamma) \vee \varphi) \sim \bigwedge_{(\psi \wedge \neg\gamma)(v)} \varphi.$$

(ii) Der Beweis läuft analog. □

Simplifikation von stark quantorenfreien Formeln mit Theorie kann zu einer Simplifikation von schwach quantorenfreien Formeln mit Theorie leicht erweitert werden. Dabei ist es sinnvoll die Information über die Erfüllungsmenge der Bound-Variablen auszunutzen indem man den Bereich des gebundenen Quantors mit einer expliziten Theorie simplifiziert, wie das folgende Lemma zeigt. Atomare Formeln in der expliziten Theorie Φ_0, die die Bound-Variable enthalten, müssen analog zur Simplifikation von regulären Quantoren mit Theorie, aus Φ_0 entfernt werden.

Lemma 2.2.15 Sei $\psi = \bigwedge_{i \in I} \psi_i$ mit $I = \{1, ..., n\}$ eine Konjunktion von atomaren Formeln. Bezeichne ψ_i' die zu $\neg\psi_i$ äquivalente atomare Formel. Setze $\Phi = \{\psi_i \mid i \in I\}$ und $\Phi' = \{\psi_i' \mid i \in I\}$.

(i) Sei φ' mit $\Phi \vDash \varphi \longleftrightarrow \varphi'$. Dann gilt

$$\bigwedge_{\psi(v)} \varphi \longleftrightarrow \bigwedge_{\psi(v)} \varphi'.$$

(ii) Sei φ' mit $\Phi' \models \varphi \longleftrightarrow \varphi'$. Dann gilt

$$\bigvee_{\psi(v)} \varphi \longleftrightarrow \bigvee_{\psi(v)} \varphi'.$$

Beweis: (i) Nach Lemma 1.1.14(ii) gilt

$$\bigwedge_{\psi(v)} \varphi \sim \forall v(\psi \longrightarrow \varphi) \sim \forall v \neg(\bigwedge_{i \in I} \psi_i) \vee \varphi \sim \forall v(\bigvee_{i \in I} \neg\psi_i) \vee \varphi \sim \forall v(\psi'_1 \vee (\psi'_2 \vee ...(\psi'_n \vee \varphi))).$$

Durch iterierte Anwendung des Lemmas 2.2.7(ii) erhält man mit der Voraussetzung $\Phi \models \varphi \longleftrightarrow \varphi'$

$$\forall v(\psi'_1 \vee (\psi'_2 \vee ...(\psi'_n \vee \varphi))) \sim \forall v(\psi'_1 \vee (\psi'_2 \vee ...(\psi'_n \vee \varphi'))) \sim \bigwedge_{\psi(v)} \varphi'.$$

(ii) Der Beweis für $\bigvee_{\psi(v)} \varphi$ läuft analog. □

Eine Verallgemeinerung von Lemma 2.2.15 für Bounds mit leerer Menge der Bound-Parameter ist die Anreicherung der Theorie mit atomaren Formeln, die man mit Hilfe der Äquivalenz

$$(z_1 \leq v \leq z_2) \vee (z_3 \leq v \leq z_4) \sim (z_1 \leq v \leq z_2) \wedge \bigwedge_{z=z_2+1}^{z_3-1} (v \neq z) \wedge (z_3 \leq v \leq z_4)$$

einer DNF des Bounds entnehmen könnte. Diese Vorgehensweise hat sich in der Praxis ebenfalls als *nicht empfehlenswert* gezeigt, da die Anzahl der atomaren Formeln, die aus „Lücken" zwischen benachbarten erfüllenden Intervallen eines Bounds entstehen können, im Vergleich zu der Anzahl atomarer Formeln der zu simplifizierenden Formel sehr groß werden kann. Ein Beispiel dafür ist

$$\bigvee_{(k=-1000 \vee k=1000)(k)} x = k.$$

2.3 Zusammenfassung

In diesem Kapitel wurden Algorithmen zur Simplifikation von Formeln der uniformen Presburger-Arithmetik vorgestellt.

Die hier vorgestellte Simplifikation atomarer Formeln besteht im Wesentlichen aus der Auswertung trivialer atomarer Formeln zu Wahrheitswerten, aus Behandlung von Kongruenzen durch Modulo-Reduktion und aus Inhaltselimination. Dabei ist der Algorithmus idempotent. Das bedeutet, daß mehrfache Anwendung der Simplifikation atomarer Formeln nicht notwendig ist. Die hier vorgestellte Auswahl von Simplifikationsregeln ist bei weitem nicht vollständig. So kann zum Beispiel die Inhaltselimination zur parametrischen Inhaltselimination erweitert werden, solange der gebildete Inhalt nicht 0 werden kann. Ferner kann das Konzept der Parity-Decomposition aus [DS97b] verwendet werden um weitere definite Terme zu entdecken oder den Grad der vorkommenden Polynomterme zu reduzieren. Ein Ausblick auf mögliche Erweiterungen ist in [Dol00] enthalten.

Bei der Bildung von Normalformeln stößt man unweigerlich bei der hier eingeführten Definition von Bounds, deren Erfüllungsmengen von Parametern abhängen, auf Schwierigkeiten. Obwohl jede Formel eine pränexe Normalform besitzt, ist es sehr schwierig einen Algorithmus zur Bildung dieser anzugeben, der keine Quantorenelimination benutzt. Analoges gilt für die Bildung von disjunktiven und konjunktiven Normalformen. Dabei läßt sich das bestehende Problem soweit reduzieren, daß zum Beispiel für die Bildung einer DNF die Bildung einer DNF für einen universellen gebundenen Quantor vor einer Disjunktion mit zwei Gliedern ausreichen würde. Für dieses Problem wurde im Rahmen dieser Arbeit keine konstruktive Lösung gefunden.

Simplifikation von stark quantorenfreien Formeln mit Theorie bildet ein mächtiges Werkzeug, wie Testbeispiele aus Kapitel 4 belegen. Man kann neben der gewöhnlichen Simplifikation auch eine Simplifikation mit einer expliziten Anfangstheorie vornehmen, wobei die Ergebnisse bezüglich dieser Theorie äquivalent sind. Dieses Konzept läßt sich erweitern auf die Simplifikation von gebundenen Quantoren und auch auf die Simplifikation von beliebigen Formeln, indem zur Simplifikation von Teilformeln der Form $\exists x\varphi$ bzw. $\forall x\varphi$ bzw. $\bigwedge_{\psi(v)} \varphi$ bzw. $\bigvee_{\psi(v)} \varphi$ einer Formel die bisher gebildete Theorie, reduziert um atomare Formeln, die die gebundene Variable enthalten, als explizite Theorie angegeben wird. Ferner können aus Bounds, die aus einer Konjunktion bestehen, atomare Formeln in die Theorie aufgenommen werden. Bei der Simplifikation mit Theorie ist das Problem der Spezifikation des Algorithmus auf den Vergleich zweier atomarer Formeln reduziert worden. Eine interessante Fragestellung ist, ob eine Verallgemeinerung dieser Vorgehensweise auf etwa 3, 4 bzw. n atomare Formeln signifikante Verbesserungen bringt.

Kapitel 3

Quantorenelimination

In diesem Kapitel wird im Rahmen des eingeführten Kalküls ein neues Quantoreneliminationsverfahren für linear quantifizierte Formeln der Presburger-Arithmetik vorgestellt. Das Verfahren wird als Quantorenelimination durch virtuelle Substitution von Testpunkten formuliert. Diese Sicht auf das Problem ermöglicht die Anwendung vieler bekannter Techniken, die bereits für reelle Quantorenelimination in [Dol00] beschrieben wurden. Dabei ist im Gegensatz zum Quantoreneliminationsverfahren aus [Wei97] die explizite Berechnung bzw. Abschätzung von Minima und Maxima der Erfüllungsmengen vorkommender gebundener Quantoren während der Quantorenelimination *nicht notwendig*. Es wird ferner gezeigt, wie man durch geringfügige Modifikationen des Verfahrens erfüllende Belegungen für den äußersten existentiellen und Gegenbeispiele für den äußersten universellen Quantorenblock in Abhängigkeit von Parametern erhalten kann.

Bei der Bildung von *strukturellen Eliminationsmengen* wird die boolesche Struktur der Eingabeformel in den Eliminationsvorgang einbezogen, indem die Berechnung einer Eliminationsmenge für eine Formel auf die Bestimmung von Eliminationsmengen von anderen, aus der Eingabeformel resultierenden Formeln zurückgeführt wird. Dies wird durch die Einführung einer speziellen rein syntaktisch definierten Eigenschaft von Teilformeln einer Formel, der *Konjunktiven Assoziiertheit*, erreicht. Die Bildung von strukturellen Eliminationsmengen kann die Erfüllungsmengen der Bounds in der schwach quantorenfreien Ergebnisformel signifikant verkleinern. Diese Beobachtung wird unter anderem durch Beispiele aus Kapitel 4 belegt. Bei der Gauss-Elimination führt eine getrennte Behandlung ausgezeichneter Teilformeln der Eingabeformel mit endlichen Erfüllungsmengen zu kleineren Eliminationsmengen und ebenfalls zu kürzeren Ausgabeformeln.

Strukturelles Condensing stellt einen Ersatz für die virtuelle Substitution dar, bei dem Teile der Formel, in die substituiert wird, durch false ersetzt werden. Dadurch wird eine Verkleinerung der Länge der quantorenfreien Ausgabeformel erreicht. Jeder Art der Bildung von strukturellen Eliminationsmengen kann eine entsprechende Condensing-Art zugeordnet werden. Eine Anwendung von Condensing für die Gauss-Elimination heißt Gauss-Condensing.

3.1 Quantorenelimination durch virtuelle Substitution

Da jede quantorenfreie Formel in pränexe Normalform gebracht werden kann, reicht es offenbar sich bei der Angabe eines Quantoreneliminationsverfahrens aufgrund der Äquivalenz $\forall x\varphi \sim \neg\exists x\neg\varphi$ auf die Angabe eines Verfahrens zur Elimination eines existenziellen Quantors $\exists x$ in einer *pränexen linear quantifizierten Formel* $\exists x\varphi$ *mit einem schwach quantorenfreien Bereich* φ zu beschränken. Aus Beispiel 1.1.16 folgt, daß es nicht möglich ist jeder quantorenfreien Formel eine stark quantorenfreie Formel zuzuordnen. Daher muß man bei der Angabe des Algorithmus explizit gebundene Quantoren behandeln. Im Folgenden soll bei der Angabe einer Presburger-Formel als „φ in $\exists x\varphi$", wenn nichts anderes angemerkt wird, stets vorausgesetzt sein, daß $\exists x\varphi$ *linear quantifiziert und pränex* und φ *schwach quantorenfrei* ist.

Bemerkung 3.1.1 Es ist durch bisherige Definitionen *nicht* explizit verboten, daß eine quantifizierte Variable im Bereich eines Quantors $\exists x$ bzw. $\forall x$ *im Bound* eines gebundenen Quantors als Bound-Parameter auftritt. In diesem Fall muß zur Quantorenelimination nach eingeführter Konvention der gebundene Quantor nach Lemma 1.1.14(i)-(ii) durch einen gewöhnlichen Quantor ersetzt und vorher eliminiert werden. Es wird im Folgenden angenommen, daß die quantifizierte Variable x *nicht* als Bound-Parameter in der Eingabeformel $\exists x\varphi$ auftritt.

Presburger-Formeln der Form $\exists x\varphi$ werden bei der Angabe des Quantoreneliminationsverfahrens in einer einheitlichen *Eliminationsnormalform* dargestellt.

Definition 3.1.2 (Eliminationsnormalform) Eine in x lineare atomare Formel φ heißt in *Eliminationsnormalform* bezüglich x, falls φ von der Form $nx\ \rho\ a$ für $\rho \in \{=, \neq, <, \leq, \geq, >\}$ bzw. $nx\ \rho_m\ a$ für $\rho \in \{\cong, \not\cong\}$ ist, sodaß $n, a, m \in \mathcal{T}$ und x in n, a und in m nicht vorkommt.

Eine Presburger-Formel φ in $\exists x\varphi$ heißt in Eliminationsnormalform bezüglich x, falls jede atomare Teilformel von φ, die x enthält, in Eliminationsnormalform bezüglich x ist. Falls die Variable x aus dem Kontext bekannt ist, so wird der Verweis darauf weggelassen.

Im diesem Kapitel wird aus Gründen der Anschauligkeit die abkürzende Schreibweise $|a|$ mit $a \in \mathcal{T}$ in atomaren Formeln verwendet. Die Schreibweise drückt semantisch die Anwendung der Betragsfunktion auf a aus. Diese Abkürzung ist legitim, da jede atomare Formel, die eine Abkürzung der Form $|a|$ enthält, stets mit Hilfe einer Fallunterscheidung nach dem Argument a der Betragsfunktion durch eine quantorenfreie Formel ohne Verwendung der Betragsfunktion dargestellt werden kann. So steht zum Beispiel die Abkürzung $|a| + x > 0$ für

$$(a \geq 0 \wedge a + x > 0) \vee (a < 0 \wedge -a + x > 0).$$

3.1.1 Eliminationsmengen und virtuelle Substitution

Quantorenelimination durch virtuelle Substitution basiert auf der intuitiven Idee für die Elimination von $\exists x$ in einer Formel $\exists x\varphi$ eine von Parametern $x_1, ..., x_l$ abhängige endliche Menge E von Testtermen anzugeben, sodaß für eine erfüllende Belegung $\mathbf{z} \in \mathbb{Z}^l$ von $(\exists x\varphi)(x_1, ..., x_l)$ für mindestens einen Testterm $t \in E$ gilt $\mathbf{P} \vDash (\varphi[t/x])(\mathbf{z})$. Aus der Gleichheit der Wahrheitswerte $(\varphi[t/x])^{\mathbf{P}}(\mathbf{z}) = \varphi^{\mathbf{P}}(\mathbf{z}, t^{\mathbf{P}}(\mathbf{z}))$ für die erweiterte Formel $\varphi(x_1, ..., x_l)$ bzw. $\varphi(x_1, ..., x_l, x)$ folgt dann unmittelbar

$$\exists x\varphi \sim \bigvee_{t \in E} \varphi[t/x].$$

Da die Testterme im Allgemeinen syntaktisch in Termini der Eingabeformel formuliert werden müssen, stößt man bei der Umsetzung obiger Idee für die uniforme Presburger-Arithmetik auf zahlreiche Probleme. In diesem Abschnitt werden Lösungen entstehender Probleme aufgezeigt und an Beispielen veranschaulicht.

Beispiel 3.1.3 Betrachte für $a \in \mathbb{Z} \setminus \{0\}$ und $b \in \mathcal{T}$ mit $\mathcal{V}(b) \subseteq \{x_1, ..., x_l\}$ die Elimination von $\exists x$ in

$$\exists x(ax = b).$$

Um einen Testterm syntaktisch für jede erfüllende Belegung von $(\exists x\varphi)(x_1, ..., x_l)$ angeben zu können, ist es naheliegend formulieren zu wollen, daß die formale Lösung $\frac{b}{a}$ für x Substituiert wird. Da Belegungen von x ebenfalls ganzzahlig sein müssen, ist eine solche Vorgehensweise nur dann möglich, wenn für die gegebene Stelle $\mathbf{z} \in \mathbb{Z}^l$ für den erweiterten Term $b(x_1, ..., x_l)$ gilt

$$a \mid b^{\mathbf{P}}(\mathbf{z}).$$

Ist a zusätzlich ein beliebiger Presburger-Term, so muß neben obiger Bedingung für den erweiterten Term $a(x_1, ..., x_l)$ auch noch $a^{\mathbf{P}}(\mathbf{z}) \neq 0$ vorausgesetzt sein. Es ist naheliegend solche Bedingungen

als Presburger-Formeln zu formulieren und in E syntaktisch zu kodieren. Für den letzten Fall wäre für die Substitution der formalen Lösung $\frac{b}{a}$

$$a \neq 0 \wedge b \cong_a 0$$

eine geeignete Bedingung. Ist diese für eine gegebene Stelle $\mathbf{z} \in \mathbb{Z}^l$ erfüllt, so kann $\frac{b^{\mathbf{P}}(\mathbf{z})}{a^{\mathbf{P}}(\mathbf{z})}$ zu einer ganzen Zahl ausgewertet und für x durch Substitution aus Definition 1.1.4 eingesetzt werden.

Im eingeführten Kalkül ist es allerdings nicht möglich Terme der Form $\frac{b}{a}$ auszudrücken. Es ist auch nicht empfehlenswert ein Funktionszeichen „$^{-1}$" für die Bildung des multiplikativen Inversen mit einer partiellen Termfunktion als Interpretation in die Sprache bzw. Struktur der Presburger-Arithmetik aufzunehmen. Die Ersetzung aus Beispiel 3.1.3 wird statt dessen als eine *erweiterte* Substitution von Termen aus einer um das zweistellige Funktionszeichen „/" expandierten Sprache aufgefaßt. Die Sprache der Presburger-Arithmetik bleibt dabei *unverändert*. Das syntaktische Objekt $/(b, a)$ für $a, b \in \mathcal{T}$ wird im Folgenden als *Pseudo-Term* bezeichnet und als $\frac{b}{a}$ für $a \neq 1$ und als b sonst geschrieben. Die beschriebene Vorgehensweise basiert auf der Tatsache, daß es stets möglich ist das Ergebnis der Substitution eines Pseudo-Terms in eine schwach quantorenfreie Presburger-Formel als eine schwach quantorenfreie Presburger-Formel auszudrücken. Dieser Vorgang wird zweckbedingt nur für Formeln in Eliminationsnormalform definiert und als *virtuelle Substitution* bezeichnet.

Definition 3.1.4 (Virtuelle Substitution) Sei $\varphi = n_i x \ \rho \ a_i$ bzw. $\varphi = n_i x \ \rho_m \ a_i$ in Eliminationsnormalform. Dann ist $\varphi[\frac{a_j}{n_j}/\!/x]$ definiert durch

$$\varphi\left[\frac{a_j}{n_j}/\!/x\right] = \begin{cases} n_i a_j \ \rho \ n_j a_i & \text{falls } \rho \in \{=, \neq\} \text{ und} \\ n_i n_j a_j \ \rho \ n_j^2 a_i & \text{falls } \rho \in \{<, >, \leq, \geq\} \text{ bzw.} \\ n_i a_j \ \rho_{m n_j} \ n_j a_i & \text{falls } \rho \in \{\cong, \ncong\}. \end{cases}$$

Für eine Presburger-Formel φ in $\exists x \varphi$ mit φ in Eliminationsnormalform bezüglich x wird mit $\varphi[\frac{a_j}{n_j}/\!/x]$ die Formel bezeichnet, die aus φ entsteht, indem jede atomare Teilformel ψ mit $x \in \mathcal{V}(\psi)$ durch $\psi[\frac{a_j}{n_j}/\!/x]$ ersetzt wird.

Die Semantik der virtuellen Substitution von $\frac{b}{a}$ für x mit $x \notin \mathcal{V}(b) \cup \mathcal{V}(a)$ in eine Formel φ in $\exists x \varphi$ ist offenbar verträglich mit der Auswertung des Pseudo-Terms $\frac{b}{a}$ im folgenden Sinne. Falls für die erweiterten Terme $a(x_1, ..., x_l)$ und $b(x_1, ..., x_l)$ und eine Stelle $\mathbf{z} \in \mathbb{Z}^l$ gilt $a^{\mathbf{P}}(\mathbf{z}) \neq 0$ und $a^{\mathbf{P}}(\mathbf{z}) \mid b^{\mathbf{P}}(\mathbf{z})$, so gilt auch für die Erweiterung $(x_1, ..., x_l)$

$$\left(\varphi\left[\frac{b}{a}/\!/x\right]\right)^{\mathbf{P}}(\mathbf{z}) = \left(\varphi\left[\frac{b^{\mathbf{P}}(\mathbf{z})}{a^{\mathbf{P}}(\mathbf{z})}/x\right]\right)^{\mathbf{P}}(\mathbf{z}).$$

Man beachte, daß im Ergebnis der virtuellen Substitution eines Pseudo-Terms $\frac{a_j}{n_j}$, in dem x nicht vorkommt, für x in φ die Variable x ebenfalls nicht mehr vorkommt. Daher ist es auch möglich für $\exists x \varphi$ und $\varphi[\frac{a_j}{n_j}/\!/x]$ die *gleiche* Erweiterung zu verwenden.

Offenbar muß bei der virtuellen Substitution $\varphi[\frac{a_j}{n_j}/\!/x]$ für einen parametrischen Nenner-Term n_j die Fallunterscheidung, ob dieser negative oder positive Werte annimmt, nicht gemacht werden. Es erscheint aus erster Sicht problematisch, daß im Ergebnis einer solchen Ersetzung n_j^2 vorkommt. Im Hinblick auf das gesetzte Ziel die Anzahl der atomaren Formeln möglichst klein zu halten ist diese Wahl allerdings vorteilhafter als die Alternative in Satz 4.2 in [Wei97], bei der sich die Anzahl der atomaren Formeln durch die Substitution im Wesentlichen verdoppelt. Die virtuelle Substitution $\varphi[\frac{a_j}{n_j}/\!/x]$ kann so durch Einführung geeigneter Fallunterscheidungen modifiziert werden, daß im Falle eines positiv oder negativ definiten Terms n_j dieser nicht quadriert werden muß. Dies liefert

folgende modifizierte Definition der virtuellen Substitution.

$$\varphi\left[\frac{a_j}{n_j}/\!/x\right] = \begin{cases} n_i a_j \ \rho \ n_j a_i & \text{falls } \rho \in \{=, \neq\}, \\ n_i a_j \ \rho \ n_j a_i & \text{falls } \rho \in \{<, >, \leq, \geq\} \text{ und } n_j \text{ positiv definit ist,} \\ n_i a_j \ \rho' \ n_j a_i & \text{falls } \rho \in \{<, >, \leq, \geq\} \text{ und } n_j \text{ negativ definit ist und} \\ n_i n_j a_j \ \rho \ n_j^2 a_i & \text{falls } \rho \in \{<, >, \leq, \geq\} \text{ und } n_j \text{ weder positiv} \\ & \text{noch negativ definit ist bzw.} \\ n_i a_j \ \rho_{mn_j} \ n_j a_i & \text{falls } \rho \in \{\cong, \not\cong\}. \end{cases}$$

Dabei bezeichne ρ' das bei der Multiplikation der atomaren Ungleichung mit -1 entstehende zu ρ duale Relationssymbol, wie zum Beispiel „$<$" zu „$>$". Es kann leicht verifiziert werden, daß das Ergebnis der so definierten Ersetzungsvorschrift $\varphi[\frac{a_j}{n_j}/\!/x]$ für einen Term a_j, der in allen Variablen linear ist, und $n_j \in \mathbb{N} \setminus \{0\}$ in eine in $\mathcal{V}(\varphi)$ lineare Formel eine mit der Ersetzung aus [Wei88] *syntaktisch gleiche* Formel liefert.

Es wird nun gezeigt, wie Quantorenelimination mit Hilfe der oben eingeführten Ersetzungsvorschrift umgesetzt werden kann. Als eine *Eliminationsmenge* für eine Formel $\exists x\varphi$ wird üblicherweise eine endliche Menge E mit

$$E = \{(\gamma, t) \mid \gamma \in \mathcal{F} \text{ stark quantorenfrei, } t \text{ ein Pseudo-Term}\}$$

bezeichnet, für die gilt

$$\exists x\varphi \sim \bigvee_{(\gamma,t)\in E} (\varphi[t/\!/x] \wedge \gamma).$$

Eliminationsmengen sind nicht eindeutig bestimmt, denn jede Obermenge einer Eliminationsmenge ist wieder eine Eliminationsmenge, wie man leicht nachprüfen kann. Ein Element (γ, t) einer Eliminationsmenge wird im Folgenden als *Testpunkt* bezeichnet.

Beispiel 3.1.5 Für die Formel $\exists x(ax = b)$ mit $a, b \in \mathcal{T}$ aus Beispiel 3.1.3 ist

$$E = \{(a \neq 0 \wedge b \cong_a 0, \frac{b}{a}), (\text{true}, 0)\}$$

eine Eliminationsmenge. Demnach gilt nach Anwendung der Simplifikation

$$\exists x(ax = b) \sim (a \neq 0 \wedge b \cong_a 0) \vee b = 0.$$

Für uniforme Presburger-Arithmetik werden allerdings Eliminationsmengen benötigt, deren Beschaffenheit abhängig von der gewählten Belegung der Parameter ist. Dies wird am folgenden sehr einfachen Beispiel veranschaulicht.

Beispiel 3.1.6 Betrachte die Elimination von $\exists x$ in der uniformen Presburger-Formel

$$\exists x(x \cong_a b \wedge x \cong_a c).$$

Eine naheliegende Vorgehensweise, falls a nicht zu 0 ausgewertet wird, ist für die Werte der zu substituierenden Terme die Restklassen von $\mathbb{Z}/a$ zu wählen. Falls a zu 0 ausgewertet wird, so ist b ein geeigneter Pseudo-Term. Obige Überlegungen liefern die schwach quantorenfreie Ergebnisformel

$$\exists x(x \cong_a b \wedge x \cong_a c) \sim \left(\bigvee_{(0\leq k<|a|)(k)} k \cong_a b \wedge k \cong_a c \right) \vee b \cong_a c.$$

Die Anzahl der Testpunkte hängt jedoch von der Auswertung des Terms a ab.

Es stellt sich also die Frage, wie man eine Eliminationsmenge, deren Testpunkte systematisch zu gebundenen Quantoren zusammengefaßt werden können, syntaktisch korrekt angeben kann. Zur Lösung des in Beispiel 3.1.6 aufgezeigten Problems ist es naheliegend eine syntaktische Kodierung der Bounds der entstehenden gebundenen Quantoren in die Testpunkte aufzunehmen. Demnach ist eine *parametrische Eliminationsmenge* E für $\exists x\varphi$ eine Menge der Form

$$E = \{(\gamma_i, t_i, \Psi_i) \mid i \in I\}.$$

Dabei ist γ_i eine stark quantorenfreie Formel und t_i ein Pseudo-Term für jedes $i \in I$. Weiter ist Ψ_i für jedes $i \in I$ ein Tupel der Form $((\psi_i^1, v_i^1), ..., (\psi_i^{q_i}, v_i^{q_i}))$, sodaß jedes ψ_i^j für $1 \leq j \leq q_i$ ein Bound eines gebundenen Quantors mit der Bound-Variable v_i^j ist. Ein solches Tupel $(\gamma_i, t_i, \Psi_i) \in E$ kann unmittelbar in eine schwach quantorenfreie Formel

$$\bigvee_{\psi_i^1(v_i^1)} ... \bigvee_{\psi_i^{q_i}(v_i^{q_i})} (\varphi[t_i /\!\!/ x] \wedge \gamma_i)$$

übersetzt werden. Die Syntax der Anwendung einer parametrischen Eliminationsmenge E ist also zu der aus [Wei88] und [Wei97] ähnlich mit

$$\exists x\varphi \sim \bigvee_{i \in I} \bigvee_{\psi_i^1(v_i^1)} ... \bigvee_{\psi_i^{q_i}(v_i^{q_i})} (\varphi[t_i /\!\!/ x] \wedge \gamma_i).$$

Es muß ferner auf die Reihenfolge der Formeln $\psi_i^1, ..., \psi_i^{q_i}$ und entsprechender gebundener Quantoren geachtet werden, da im Folgenden die Erfüllungsmengen der inneren Bounds von den Erfüllungsmengen der äußeren Bounds abhängen werden.

Beispiel 3.1.7 Für das Beispiel 3.1.6 ist nach obiger Konvention eine parametrische Eliminationsmenge E gegeben durch

$$E = \{(\text{true}, k, ((0 \leq k < |a|, k))), (\text{true}, b, \emptyset)\}.$$

Es ist an dieser Stelle sinnvoll zu bemerken, daß obwohl eine parametrische Eliminationsmenge E durch die hier vorgeschlagene Kodierung syntaktisch nicht mehr einer gewöhnlichen Eliminationsmenge gleicht, bei der Anwendung von E hauptsächlich die am Anfang dieses Abschnittes formulierte Eigenschaft verwendet wird. Gilt nämlich an einer Stelle $\mathbf{z} \in \mathbb{Z}^l$ die Formel $(\exists x\varphi)(x_1, ..., x_l)$, so gibt es ein $(\gamma_i, t_i, \Psi_i) \in E$, sodaß für eine Belegung $\mathbf{u} = (u_1, ..., u_{q_i}) \in \mathbb{Z}^{q_i}$ des Tupels der Bound-Variablen $(v_i^1, ..., v_i^{q_i})$ mit $u_j \in S^{(\mathbf{x}, v_1^1, ..., v_i^{j-1})}_{(\mathbf{z}, u_1, ..., u_{j-1})}(\psi_i^j(\mathbf{x}, v_i^1, ..., v_i^j))$ für den Pseudo-Term $t_s = t_i[u_1/v_i^1, ..., u_{q_i}/v_i^{q_i}]$ und $\gamma_s = \gamma_i[u_1/v_i^1, ..., u_{q_i}/v_i^{q_i}]$ gilt

$$\mathbf{P} \vDash (\varphi[t_s /\!\!/ x] \wedge \gamma_s)(\mathbf{z}).$$

Der restliche syntaktische Apparat wird lediglich dazu benötigt die Ergebnisformel durch gebundene Quantoren auszudrücken. Die Menge E kann also für jede Belegung der Parameter der Eingabeformel mit ganzzahligen Werten als eine Eliminationsmenge im gewöhnlichen Sinne angesehen werden. Im Folgenden wird für die Erfüllungsmenge eines Bounds ψ bezüglich der Bound-Variable v für eine feste aus dem Kontext bekannte Wahl der Bound-Parameter die etwas informelle Schreibweise $S(\psi(v))$ verwendet. So wird für eine parametrische Eliminationsmenge E in obiger Konvention mit $E(\mathbf{z})$ die Menge

$$E(\mathbf{z}) = \left\{(\gamma_i[u_1/v_i^1, ..., u_{q_i}/v_i^{q_i}], t_i[u_1/v_i^1, ..., u_{q_i}/v_i^{q_i}]) \mid i \in I, u_j \in S(\psi_i^j(v_i^j)) \text{ für } 1 \leq j \leq q_i\right\}$$

bezeichnet. Dabei gilt $S(\psi_i^j(v_i^j)) = S^{(\mathbf{x}, v_i^1, ..., v_i^{j-1})}_{(\mathbf{z}, u_1, ..., u_{j-1})}(\psi_i^j(\mathbf{x}, v_i^1, ..., v_i^j))$, wie aus dem Kontext ersichtlich ist. Weiter wird ein leeres Tupel Ψ_i mit der Formel true identifiziert, sodaß für das Tupel $(\gamma_i, t_i, \emptyset)$ in $E(\mathbf{z})$ lediglich der Testpunkt (γ_i, t_i) vorhanden ist. Man beachte, daß true in diesem Fall *nicht*

als Bound verwendet wird, da bei einem leeren Tupel Ψ_i keine gebundene Quantoren entstehen. Die Menge $E(\mathbf{z})$ ist nach obigen Überlegungen eine Eliminationsmenge für $(\exists x\varphi)[z_1/x_1, ..., z_l/x_l]$. Für eine parametrische Eliminationsmenge E für $\exists x\varphi$ wird für Beweise die folgende Charakterisierung ihrer Semantik verwendet. Für alle $\mathbf{z} \in \mathbb{Z}^l$ gilt bei gleicher Erweiterung $(x_1, ..., x_l)$

$$\mathbf{P} \models (\exists x\varphi)(\mathbf{z}) \quad \text{genau dann, wenn} \quad \mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E(\mathbf{z})} \varphi[t//x] \wedge \gamma \right)(\mathbf{z}).$$

Im Folgenden wird unter dem Begriff „Eliminationsmenge" stets eine parametrische Eliminationsmenge verstanden.

3.1.2 Uniforme Quantorenelimination

In diesem Abschnitt wird ein zu dem in [Wei97] vorgestellten ähnliches Quantoreneliminationsverfahren für die uniforme Presburger-Arithmetik angegeben. Es wird daran erinnert, daß das grundsätzliche Problem der Elimination eines existenziellen Quantors $\exists x$ in einer linear quantifizierten pränexen Formel $\exists x\varphi$ mit φ schwach quantorenfrei behandelt wird.

Für eine Formel φ in $\exists x\varphi$ in Eliminationsnormalform mit der Erweiterung $(x_1, ..., x_l, x)$ seien folgende Bezeichnungen gültig. Sei

$$\underset{\psi_1(v_1)}{Q_1} \ , ..., \ \underset{\psi_n(v_n)}{Q_n}$$

der Präfix von φ aus gebundenen Quantoren mit Bounds $\psi_1(v_1), ..., \psi_n(v_n)$. Bezeichne weiterhin die Menge $A = \{n_i x \ \rho_i \ a_i \mid i \in I\}$ die Teilmenge der atomaren Teilformeln von φ mit $\rho_i \in \{=, \neq, <, >, \leq, \geq\}$ für $i \in I$, die x enthalten. Weiterhin sei K die Menge der atomaren Teilformeln von φ mit $K = \{\rho_j(n_j x, a_j, m_j) \mid j \in J\}$ und $\rho_j \in \{\cong, \ncong\}$ für $j \in J$, die x enthalten. Dabei gelte $I \cap J = \emptyset$ und $I \cup J \neq \emptyset$.

Bemerkung 3.1.8 Jede atomare Teilformel von φ in obiger Konvention für $k \in I \cup J$ hat bis auf die Vertauschung von Summanden die Form

$$n_k x \ \rho_k \sum_{j=1}^{n} c_{kj} v_j + r_k \quad \text{bzw.} \quad \rho_k(n_k x, \sum_{j=1}^{n} c_{kj} v_j + r_k, m_k),$$

wobei x und jedes v_j für $1 \leq j \leq n$ in den restlichen Termen n_k, c_{ki} für $1 \leq i \leq n$, r_k und m_k nicht vorkommen. Der Term $\sum_{j=1}^{n} c_{kj} v_j$ wird im Folgenden mit t_k bezeichnet.

Zunächst wird eine Eigenschaft von (In-)Kongruenzen bewiesen, die im Laufe des Beweises für das Verfahren benötigt wird. Eine *m-periodische Teilmenge* $M \subseteq \mathbb{Z}$ ist eine Menge für die aus $a \in M$ auch $a \pm m \in M$ folgt.

Lemma 3.1.9 Seien $n_i, a_i, m_i \in \mathbb{N} \setminus \{0\}$ für $i \in J' \subseteq J$ nach obiger Konvention. Dann ist die Erfüllungsmenge von

$$\psi = \bigwedge_{j\in J'} \rho_j(n_j x, a_j, m_j)$$

bezüglich x eine periodische Menge mit einer Periode $m = \text{kgV}\{m_j \mid j \in J'\}$. Insbesondere sind Schnittmengen von Erfüllungsmengen atomarer (In-)Kongruenzen periodische Mengen.

Beweis: Endliche Schnitte periodischer Teilmengen von $\mathbb{Z}$ sind wieder periodische Mengen. Sei dazu etwa $P_1 \subseteq \mathbb{Z}$ eine p_1-periodische Menge und $P_2 \subseteq \mathbb{Z}$ eine p_2-periodische Menge. Dann ist die Schnittmenge p-periodisch mit $p = \text{kgV}\{p_1, p_2\}$. Dies ist wie folgt einzusehen. Aus $i \in P_1 \cap P_2$ folgt auch $i \in P_1$ und $i \in P_2$. Da $p = \text{kgV}\{p_1, p_2\}$ ein Vielfaches von p_1 und p_2 ist mit $p = q_1 p_1$ und $p_2 = q_2 p_2$ mit $q_1, q_2 \in \mathbb{Z}$, gilt auch $i \pm p = i \pm q_1 p_1 \in P_1$ und $i \pm p = i \pm q_2 p_2 \in P_2$. Somit

gilt $i \pm p \in P_1 \cap P_2$. Die Erfüllungsmengen von Kongruenzen und Inkongruenzen $\rho_j(n_j x, a_j, m_j)$ in obiger Form sind per Definition m_j-periodische Teilmengen von $\mathbb{Z}$. Somit ist die Erfüllungsmenge von ψ als endlicher Schnitt von m_j-periodischen Teilmengen von $\mathbb{Z}$ eine m-periodische Menge. Die Periode berechnet sich durch verschachtelte Anwendung der Bildung des kleinsten gemeinsamen Vielfachen mit $m = \text{kgV}\{m_j \mid j \in J'\}$. $\square$

Die Wahl der Testpunkte für das Verfahren beruht auf der Tatsache, daß die Erfüllungsmenge $S^{(x_1,...,x_l)}_{(z_1,...,z_l)}(\varphi(x_1,...,x_l,x))$ einer erweiterten schwach quantorenfreien Formel $\varphi(x_1,...,x_l,x)$ in obiger Konvention bezüglich x für eine feste Belegung $\mathbf{z} = (z_1,...,z_l)$ der Parameter eine endliche Vereinigung der Form

$$\bigcup_{i \in M} (L_i \cap S_i)$$

ist. Dabei ist jedes $L_i \subseteq \mathbb{Z}$ ein Intervall und jedes $S_i \subseteq \mathbb{Z}$ eine Schnittmenge von Erfüllungsmengen ausgewählter atomarer (In-)Kongruenzen. Da jede Menge S_i nach Lemma 3.1.9 eine periodische Menge ist, reicht es also um die durch die Terme n_k und a_k dargestellten „Ränder" des jeweiligen Intervalls L_i symmetrisch einen Bereich der Testpunkte zu wählen, der die Periode von S_i durchläuft. Eine schematische Darstellung der oben beschriebenen Strategie für den Intervallrand a der Formel $x \geq a \wedge x \leq b \wedge x \cong_3 0$ ist in Abbildung 3.1 angegeben.

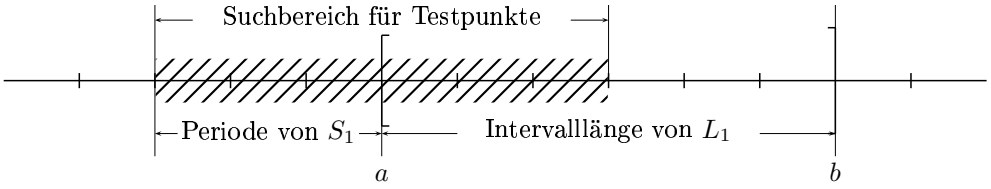


Abbildung 3.1: Schematische Darstellung der Wahl der Testpunkte

Eine Möglichkeit eine gemeinsame Periode beteiligter (In-)Kongruenzen abzuschätzen, ist die Bildung des polynomiellen kleinsten gemeinsamen Vielfachen *aller* in der Eingabe vorkommender Moduli. Diese Strategie wird im Abschnitt 3.2 verfeinert.

Lemma 3.1.10 Seien $a, b \in \mathcal{T}$ beliebige Presburger-Terme mit einer gemeinsamen Erweiterung $(x_1,...x_l)$. Dann gilt für jede Stelle $\mathbf{z} \in \mathbb{Z}^l$

$$|\text{kgV}\{a^{\mathbf{P}}(\mathbf{z}), b^{\mathbf{P}}(\mathbf{z})\}| \leq |(\text{kgV}\{a,b\})^{\mathbf{P}}(\mathbf{z})|.$$

Beweis: Im Folgenden seien Exponenten stets positive natürliche Zahlen. Da $\mathbb{Z}[x_1,...,x_l]$ ein faktorieller Bereich ist, existiert auch eine Primfaktorzerlegung von a und b von der Form $a = e_a p_1^{u_1}...p_i^{u_i}$ und $b = e_b p_1^{w_1}...p_i^{w_i}$. Dann ist

$$\text{kgV}\{a,b\} = e_c p_1^{\max\{w_1,u_1\}}...p_i^{\max\{w_i,u_i\}}.$$

Somit gilt $|(\text{kgV}\{a,b\})^{\mathbf{P}}(\mathbf{z})| = |e_c||p_1^{\mathbf{P}}(\mathbf{z})|^{\max\{w_1,u_1\}}...|p_i^{\mathbf{P}}(\mathbf{z})|^{\max\{w_i,u_i\}}$. Seien $q_1,...,q_j$ Primzahlen mit $p_k^{\mathbf{P}}(\mathbf{z}) = r_k q_1^{e_{1k}}...q_j^{e_{jk}}$. Dann gilt

$$\text{kgV}\{a^{\mathbf{P}}(\mathbf{z}), b^{\mathbf{P}}(\mathbf{z})\} = \text{kgV}\{r' q_1^{u_1e_{11}+...+u_ie_{1i}}...q_j^{u_1e_{j1}+...+u_ie_{ji}}, s' q_1^{w_1e_{11}+...+w_ie_{1i}}...q_j^{w_1e_{j1}+...+w_ie_{ji}}\}.$$

mit Einheiten $r', s' \in \mathbb{Z}$. Für die Potenz b_k von q_k in $|\text{kgV}\{a^{\mathbf{P}}(\mathbf{z}), b^{\mathbf{P}}(\mathbf{z})\}|$ gilt also

$$b_k = \max\{u_1e_{k1} + ... + u_ie_{ki}, w_1e_{k1} + ... + w_ie_{ki}\}.$$

Analog dazu gilt für die Potenz d_k von q_k in $|(\text{kgV}\{a,b\})^{\mathbf{P}}(\mathbf{z})|$

$$d_k = \max\{u_1, w_1\}e_{k1} + ... + \max\{u_i, w_i\}e_{ki}.$$

Offensichtlich gilt aufgrund der Positivität der Exponenten $d_k \geq b_k$ für alle $1 \leq k \leq j$. Das liefert die Behauptung. □

Die Tatsache, daß die parametrischen Moduli der (In-)Kongruenzen zu 0 ausgewertet werden können, stellt für eine solche Umsetzung des Verfahrens ein Hindernis dar. Dies wird am folgenden Beispiel sichtbar.

Beispiel 3.1.11 Betrachte die Elimination von $\exists x$ in $\exists x\varphi$ mit

$$\varphi = ((x \cong_a 0 \vee x \cong_b 0) \wedge x > 0).$$

Die naive Vorgehensweise die Periode etwa durch $\mathrm{kgV}\{a, b\} = ab$ abzuschätzen führt nicht zum Erfolg. Obige Abschätzung liefert das schwach quantorenfreie Ergebnis

$$\varphi' = \bigvee_{(-|ab| \leq k \leq |ab|)(k)} (k \cong_a 0 \vee k \cong_b 0) \wedge k > 0.$$

Dieses entsteht als Disjunktion von drei identischen Formeln. Eine davon repräsentiert den Randpunkt des Intervalls definiert durch die atomare Formel[1] $x > 0$ und die anderen jeweils die beiden Gleichungen, die aus den Kongruenzen durch Auswertung der Moduli zu 0 entstehen können. Offenbar gilt für die erweiterte Formel $\varphi(a, b, x)$

$$\mathbf{P} \models \varphi(0, 3, 3).$$

Die Formel $\varphi'[0/a, 3/b]$ ist allerdings nach Vereinfachung von der Form

$$\varphi' = \bigvee_{(k=0)(k)} (k = 0 \vee k \cong_3 0) \wedge k > 0 \sim \mathrm{false}.$$

Somit sind $\exists x\varphi$ und φ' nicht äquivalent im Sinne der Definition 1.1.9.

Es ist nicht empfehlenswert eine Fallunterscheidung nach den zu 0 auswertbaren Moduli als Teil des Verfahrens zu machen, da diese im Worst-Case $2^{|J|}$ viele Fälle umfaßt. Dem Problem aus Beispiel 3.1.11 wird dadurch begegnet, daß jeder Modulus durch einen festen positiv definiten Term abgeschätzt wird. Für den folgenden Satz wird zunächst die Abschätzung $\mathrm{kgV}\{|m_j| + 1 \mid j \in J\}$ verwendet. Diese artet immer noch aufgrund der abkürzenden Schreibweise durch das Auflösen der Beträge zu einer exponentiellen Fallunterscheidung aus. Nach dem Beweis der Gültigkeit des Verfahrens folgt eine Abschätzung dieser Größe, die eine exponentielle Fallunterscheidung auf Kosten der größeren Suchbereiche bzw. Erfüllungsmengen der Bounds vermeidet.

Satz 3.1.12 (Uniforme Quantorenelimination) *Sei φ in obiger Konvention gegeben. Seien r_k und t_k für $k \in I \cup J$ Teilterme von a_k nach Bemerkung 3.1.8. Seien $v'_1, ..., v'_n, v$ neue Variablen. Sei $\psi'_i = \psi_i[v'_1/v_1, ..., v'_n/v_n]$ für $1 \leq i \leq n$. Für ein $k \in I \cup J$ und $m = \mathrm{kgV}\{|m_j| + 1 \mid j \in J\}$ definiere $t'_k = t[v'_1/v_1, ..., v'_n/v_n]$ und*

$$\gamma_k = -|n_k|m \leq v - t'_k \leq |n_k|m.$$

Dann ist mit

$$E_r = \left\{ \left(n_k \neq 0 \wedge r_k + v \cong_{n_k} 0, \frac{r_k + v}{n_k}, ((\psi'_1, v'_1), ..., (\psi'_n, v'_n), (\gamma_k, v)) \right) \mid k \in I \cup J \right\}.$$

die Menge $E = E_r \cup \{(\mathrm{true}, 0, \emptyset)\}$ eine Eliminationsmenge für $\exists x\varphi$.

[1] Als Mittelpunkt des symmetrischen Bereiches ist 0 gewählt worden. Daher muß man allerdings die Werte $|ab|$ und $-|ab|$ in den Suchbereich einschließen um alle Kongruenzklassen durchlaufen zu können. Dies wird im Folgenden als Alternative zur Umwandlung aller Ordnungsrelationen in „$\geq$" oder „$\leq$" stets so gemacht.

Beweis: Sei zunächst $\mathbf{z} = (z_1, ..., z_l) \in \mathbb{Z}^l$, sodaß $\mathbf{P} \models (\exists x \varphi)(\mathbf{z})$. Gelte etwa $\mathbf{P} \models \varphi(\mathbf{z}, z)$ für ein $z \in \mathbb{Z}$. Zu zeigen ist, daß ein $(\gamma, t) \in E(\mathbf{z})$ existiert, sodaß

$$\mathbf{P} \models (\varphi[t/\!/x] \wedge \gamma)(\mathbf{z}).$$

Gelte nun $m_j^{\mathbf{P}}(\mathbf{z}) = 0$ genau dann, wenn $j \in J_0$. Betrachte die Formel φ_s, die aus $\varphi[z_1/x_1, ..., z_l/x_l]$ entsteht, indem man jede Kongruenz $\rho_j(n_j x, r_j + t_j, m_j)$ mit $j \in J_0$ durch eine Gleichung $n_j x = r_j + t_j$ und jede Inkongruenz durch eine entsprechende Ungleichung ersetzt. Definiere dabei auch ρ_j entsprechend neu. Es gilt dann offensichtlich $\varphi(\mathbf{z}, z') = \varphi_s(z')$ für alle $z' \in \mathbb{Z}$. Man bilde eine zu φ_s äquivalente Formel φ_e, die dadurch entsteht, daß man jeden gebundenen Quantor in φ_s durch seine Expansion ersetzt. Da φ_e stark Quantorenfrei ist, kann man eine disjunktive Normalform φ_d von φ_e bilden. Der Wert z erfüllt aufgrund der Äquivalenz von φ_s und φ_d somit mindestens ein Disjunktionsglied von φ_d. Dieses ist nach dem Entfernen von den zu true auswertbaren Teilformeln, die nach Simplifikation x nicht enthalten, von der Form

$$\omega = \underbrace{\left(\bigwedge_{i \in I'} \bigwedge_{\mathbf{u} \in U_i} n_i^{\mathbf{P}}(\mathbf{z}) x \ \rho_i \ t_i^{\mathbf{P}}(\mathbf{z}, \mathbf{u}) + r_i^{\mathbf{P}}(\mathbf{z}) \right)}_{\omega_1} \wedge \underbrace{\left(\bigwedge_{j \in J'} \bigwedge_{\mathbf{u} \in U_j} \rho_j(n_j^{\mathbf{P}}(\mathbf{z}) x, t_j^{\mathbf{P}}(\mathbf{z}, \mathbf{u}) + r_j^{\mathbf{P}}(\mathbf{z}), m_j^{\mathbf{P}}(\mathbf{z})) \right)}_{\omega_2}.$$

Dabei sind für $I' \subseteq I \cup J_0$, $J' \subseteq J \setminus J_0$ die endlichen Mengen U_i, $U_j \subseteq \mathbb{Z}^n$ für jedes $i \in I'$ bzw. $j \in J'$ so definiert, daß für jedes $\mathbf{u} = (u_1, ..., u_n)$ mit $\mathbf{u} \in U_i$ bzw. $\mathbf{u} \in U_j$ die Komponente[2] $u_k \in S(\psi_k(v_k))$ für jedes $1 \leq k \leq n$. Ferner gilt $n_i^{\mathbf{P}}(\mathbf{z}) \neq 0$ für $i \in I' \cup J'$. Nun folgt eine Fallunterscheidung nach der Form von $S(\omega(x)) \neq \emptyset$. Falls $S(\omega(x)) = \mathbb{Z}$, so ist das betrachtete Disjunktionsglied äquivalent zu true. Dann gilt

$$\top = \varphi^{\mathbf{P}}(\mathbf{z}, z) = \varphi_s^{\mathbf{P}}(z) = \omega(0) = \varphi_s^{\mathbf{P}}(0) = \varphi^{\mathbf{P}}(\mathbf{z}, 0).$$

Die Substitution des Testpunktes $(\text{true}, 0) \in E(\mathbf{z})$ liefert die Behauptung $\mathbf{P} \models \varphi[0/\!/x](\mathbf{z})$. Sei nun $S(\omega(x)) \neq \mathbb{Z}$.

(i) Falls $S(\omega_1(x)) = \mathbb{Z}$, so ist wegen $S(\omega_2(x)) \neq \mathbb{Z}$ die Menge J' nicht leer. Wähle dann für folgende Betrachtungen ein $i \in J'$. Dann gilt für ein $\mathbf{u} \in U_i$ und ein c' mit $-|n_i^{\mathbf{P}}(\mathbf{z})| \leq c' \leq |n_i^{\mathbf{P}}(\mathbf{z})|$

$$c = \frac{r_i^{\mathbf{P}}(\mathbf{z}) + t_i^{\mathbf{P}}(\mathbf{z}, \mathbf{u}) + c'}{n_i^{\mathbf{P}}(\mathbf{z})} \in \mathbb{Z}.$$

Die Menge $S(\omega_2(x))$ ist eine periodische Menge. Sei m' die kleinste Periode von $S(\omega_2(x))$. Sei weiter für folgende Betrachtungen $l = c + m'$.

(ii) Falls $S(\omega_1(x)) \neq \mathbb{Z}$, dann gibt es wegen $S(\omega(x)) \neq \emptyset$ ein $l \in S(\omega(x))$. Nach der Interpretation stark quantorenfreier Formeln gilt auch $l \in S(\omega_1(x))$. Die Erfüllungsmenge $S(\omega_1(x))$ ist eine Vereinigung endlich vieler Intervalle[3] in $\mathbb{Z}$, deren ganzzahlige Ränder durch atomare Formeln in ω_1 beschrieben werden. Daher liegt l in einem dieser Intervalle L, welches nach unten oder nach oben beschränkt ist. Sei L o.B.d.A. nach unten beschränkt, womit ein kleinstes Element $c \in L$ existiert. Da c durch eine der atomaren Formeln in ω_1 repräsentiert wird, gilt weiter für ein $i \in I \cup J_0$, ein $\mathbf{u} \in U_i$ und ein c' mit $-|n_i^{\mathbf{P}}(\mathbf{z})| \leq c' \leq |n_i^{\mathbf{P}}(\mathbf{z})|$

$$c = \frac{r_i^{\mathbf{P}}(\mathbf{z}) + t_i^{\mathbf{P}}(\mathbf{z}, \mathbf{u}) + c'}{n_i^{\mathbf{P}}(\mathbf{z})}.$$

[2] Man beachte, daß $S(\psi_k(v_k))$ nach Vereinbarung am Anfang des Kapitels eine informelle Notation ist.

[3] Eine Vereinigung entsteht aufgrund der möglichen Anwesenheit von Disgleichungen. Sind keine Disgleichungen in ω_1 vorhanden, ist die Erfüllungsmenge von ω_1 bezüglich x ein Intervall.

Beide Fälle werden nun gleichzeitig behandelt. In beiden Fällen gilt, wie bereits angemerkt, $n_i^{\mathbf{P}}(\mathbf{z}) \neq 0$. Da die Menge $S(\omega_2(x)) \neq \emptyset$ eine periodische Menge mit der kleinsten Periode m' mit $0 < m' \leq m^{\mathbf{P}}(\mathbf{z})$ ist, gilt für ein $z' \leq l$ mit

$$c \leq z' \leq c + m' < c + m^{\mathbf{P}}(\mathbf{z})$$

in beiden Fällen $\mathbf{P} \models \omega_2(z')$ und somit wegen $c \leq z' \leq l$ auch $\mathbf{P} \models \omega(z')$. Setze

$$k = c' + n_i^{\mathbf{P}}(\mathbf{z})(z' - c) + (t')^{\mathbf{P}}(\mathbf{z}, \mathbf{u}).$$

Dann gilt $-|n_i^{\mathbf{P}}(\mathbf{z})|m^{\mathbf{P}}(\mathbf{z}) \leq k - (t')^{\mathbf{P}}(\mathbf{z}, \mathbf{u}) \leq |n_i^{\mathbf{P}}(\mathbf{z})|m^{\mathbf{P}}(\mathbf{z})$. Somit gilt $k \in S(\gamma_i(v))$. Dann ergibt sich für den Testpunkt $(n_i \neq 0 \wedge r_i + k \cong_{n_i} 0, \frac{r_i+k}{n_i}) \in E(\mathbf{z})$ wegen $t^{\mathbf{P}}(\mathbf{z}, \mathbf{u}) = (t')^{\mathbf{P}}(\mathbf{z}, \mathbf{u})$

$$\frac{r_i^{\mathbf{P}}(\mathbf{z}) + k}{n_i^{\mathbf{P}}(\mathbf{z})} = \frac{r_i^{\mathbf{P}}(\mathbf{z}) + c' + n_i^{\mathbf{P}}(\mathbf{z})(z' - c) + t^{\mathbf{P}}(\mathbf{z}, \mathbf{u})}{n_i^{\mathbf{P}}(\mathbf{z})} = c + z' - c = z'.$$

Es gilt dann offensichtlich $n_i^{\mathbf{P}}(\mathbf{z}) \mid r_i^{\mathbf{P}}(\mathbf{z}) + k$. Das liefert schließlich mit $\varphi_d(z') = \varphi(\mathbf{z}, z') = \top$ die Behauptung

$$\mathbf{P} \models \left(\varphi \left[\frac{r_i + k}{n_i} /\!\!/ x\right] \wedge (n_i \neq 0 \wedge r_i + k \cong_{n_i} 0)\right)(\mathbf{z}).$$

Der Fall, daß L nach unten unbeschränkt und somit nach oben beschränkt ist, läuft analog. Die Umkehrung ist offensichtlich. □

Für eine Formel $\exists x \varphi$ in obiger Konvention liefert die Anwendung einer parametrischen Eliminationsmenge nach Satz 3.1.12 eine zu $\exists x \varphi$ äquivalente Formel φ' der Form

$$\varphi' = \left(\bigvee_{k \in I \cup J} \bigvee_{\psi_1'(v_1')} \ldots \bigvee_{\psi_n'(v_n')} \bigvee_{\gamma_k(v)} Q_1^{\psi_1(v_1)} \ldots Q_n^{\psi_n(v_n)} (\varphi[t_k /\!\!/ x] \wedge \gamma_k)\right) \vee \varphi[0 /\!\!/ x].$$

Dabei kommt die Variable v_i' für $1 \leq i \leq n$ in φ' nur in den Bounds ψ_j' und in γ_k vor. Das Eliminationsergebnis sollte dabei als *gebundene Quantorenelimination* und nicht als Expansion entsprechender gebundener Quantoren aufgefaßt werden. Obige Ergebnisform ist neben der Form der Beschreibung der Suchbereiche durch Bounds syntaktisch der Hauptunterschied zwischen diesem Verfahren und dem aus [Wei97]. Das Quantoreneliminationsverfahren aus Satz 3.1.12 verallgemeinert weiterhin das Verfahren aus [Wei88] im folgenden Sinne.

Korollar 3.1.13 (Nicht uniforme Elimination) Seien die Voraussetzungen aus Satz 3.1.12 erfüllt. Gelte zusätzlich $I \neq \emptyset$ und sei φ in allen Variablen linear. Gelte weiter $0 < n_i \in \mathbb{Z}$ und $0 < m_j \in \mathbb{Z}$ für alle $i \in I \cup J$ und $j \in J$. Sei $s_i^{\min} = \min(S(\psi_i'(v_i')))$ und $s_i^{\max} = \max(S(\psi_i'(v_i')))$ für $1 \leq i \leq n$. Sei weiter $s^{\min} = \min\{(t')^{\mathbf{P}}(u_1, ..., u_n) \mid u_i \in \{s_i^{\min}, s_i^{\max}\}\}$ und $s^{\max} = \max\{(t')^{\mathbf{P}}(u_1, ..., u_n) \mid u_i \in \{s_i^{\min}, s_i^{\max}\}\}$. Sei

$$\gamma_k = -n_j m + s^{\min} \leq v \leq n_j m + s^{\max}.$$

Dann ist eine Eliminationsmenge für $\exists x \varphi$ gegeben durch

$$E = \left\{\left(r_k + v \cong_{n_k} 0, \frac{r_k + v}{n_k}, ((\gamma_k, v))\right) \mid k \in I\right\}.$$

Beweis: Offenbar gilt $J_0 = \emptyset$ unter gegebenen Voraussetzungen im Beweis des Satzes 3.1.12. Daraus ergibt sich unmittelbar die Gestalt der Eliminationsmenge. Zu zeigen ist somit nur, daß für die durch die Anwendung des Satzes 3.1.12 entstehende Bound-Formel $\gamma_k' = -n_j m \leq v - t' \leq n_j m$ gilt

$$S(\gamma_k'(v)) \subseteq S(\gamma_k(v)).$$

Sei dazu ein $u \in S(\gamma'_k(v))$ gegeben. Dann gilt für $\mathbf{u} = (u_1, ..., u_n) \in \mathbb{Z}^n$ mit $u_i \in S(\psi'_i(v'_i))$ für $0 \leq i \leq n$

$$-n_j m \leq u - (t')^{\mathbf{P}}(u_1, ..., u_n) \leq n_j m$$

Es ist nicht schwer zu sehen, daß gilt

$$s^{\min} \leq (t')^{\mathbf{P}}(u_1, ..., u_n) \leq s^{\max}.$$

Somit läßt sich u durch $u \leq n_j m + (t')^{\mathbf{P}}(u_1, ..., u_n) \leq n_j m + s^{\max}$ und durch $u \geq -n_j m + (t')^{\mathbf{P}}(u_1, ..., u_n) \geq -n_j m + s^{\min}$ abschätzen. Somit gilt $u \in S(\gamma_k(v))$. □

Die Vorgaben des Korollars 3.1.13 entsprechen denen aus [Wei88]. Bemerkenswert ist, daß $n = \text{kgV}\{n_i \mid i \in I \cup J\}$ als Bestandteil der Eliminationsmenge und somit auch der schwach quantorenfreien Ergebnisformel *nicht* auftritt. Dies führt zu einer erheblichen Verkleinerung der Erfüllungsmengen von Bounds und somit auch der Anzahl der einzusetztenden Testpunkte bei der Auswertung der Ergebnisformel. Dieser Fortschritt ist für praktische Anwendungen von besonderer Bedeutung.

3.1.3 Modifikationen des Verfahrens

Es folgen nun einige implementierungsrelevante Modifikationen des Verfahrens, welche hauptsächlich die Laufzeit während der Auswertungsphase verbessern. Diese entstehen durch eine alternative Wahl der Abschätzung m und der Eliminationsmenge E_r. Beweise für folgende Aussagen können leicht dem Beweis des Kernverfahrens 3.1.12 durch Einschränkung auf die Spezialfälle entnommen werden.

Die Abschätzung $m = \text{kgV}\{|m_j| + 1 \mid j \in J\}$ liefert beim Ausformulieren entsprechender Formeln eine Fallunterscheidung mit $2^{|J|}$ Fällen. Dies ist aus Komplexitätsgründen nicht vorteilhaft. Die folgende Wahl von m bietet eine Lösung dieses Problems. Definiere für $j \in J$

$$m'_j = \begin{cases} m_i & \text{falls } m_i \text{ positiv definit ist,} \\ m_i + 1 & \text{falls } m_i \text{ positiv semidefinit ist,} \\ -m_i & \text{falls } m_i \text{ negativ definit ist,} \\ -m_i + 1 & \text{falls } m_i \text{ negativ semidefinit ist und} \\ m_i^2 + 1 & \text{sonst.} \end{cases}$$

Dann kann $m = \text{kgV}\{m'_j \mid j \in J\}$ als Abschätzung der Periode im Satz 3.1.12 verwendet werden. Der Term m' ist als kleinstes gemeinsames Vielfaches positiv definiter Terme selbst positiv definit. Die Abschätzung entspricht im Falle nichtparametrischer Moduli $m_i \in \mathbb{Z} \setminus 0$ genau der kleinsten gemeinsamen Periode aller (In-)Kongruenzen in der Eingabe.

Eine (In-)Kongruenz mit einem positiv oder negativ definiten Modulus kann im Beweis des Satzes 3.1.12 *nicht* in ω_1 als Gleichung oder Ungleichung auftauchen. Somit kann man für die Menge der Moduli $J^* \subseteq J$, die weder positiv noch negativ definit sind, die Menge E_r in Satz 3.1.12 modifizieren zu

$$E_r = \left\{ \left(n_k \neq 0 \wedge r_k + v \cong_{n_k} 0, \frac{r_k + v}{n_k}, ((\psi'_1, v'_1), ..., (\psi'_n, v'_n), (\gamma_k, v)) \right) \mid k \in I \cup J^* \right\}.$$

Alle genannten Verbesserungen können miteinander kombiniert werden. Als nächstes folgen zwei Beispiele der Anwendung des Satzes 3.1.12.

Beispiel 3.1.14 Die Elimination von $\exists x$ in $\exists x \varphi$ aus Beispiel 3.1.11 mit

$$\varphi = ((x \cong_a 0 \vee x \cong_b 0) \wedge x > 0).$$

liefert eine schwach quantorenfreie äquivalente Ergebnisformel

$$\varphi' = \bigvee_{(-(a^2+1)(b^2+1)\leq k\leq (a^2+1)(b^2+1))(k)} (k \cong_a 0 \vee k \cong_b 0) \wedge k > 0.$$

Speziell für die Belegung $(0,3)$ des Variablentupels (a,b) ergibt sich

$$\varphi'[0/a, 3/b] \sim \bigvee_{(-10\leq k\leq 10)(k)} (k = 0 \vee k \cong_3 0) \wedge k > 0 \sim \bigvee_{(-10\leq k\leq 10)(k)} k \cong_3 0 \wedge k > 0 \sim \text{true}.$$

Die Auswertung der schwach quantorenfreien Formel in Beispiel 3.1.14 macht die Kosten deutlich, die durch die großzügige Abschätzung der Periode durch Quadrieren der Moduli entstehen. Es wäre ausreichend, den Bound des gebundenen Quantors für die Auswertung oben durch $-3 \leq k \leq 3$ zu beschränken. Als nächstes wird gezeigt, welche Vorteile das Verfahren und seine Modifikationen im nicht uniformen Fall bringen.

Beispiel 3.1.15 Man betrachte die Elimination von $\exists x$ in $\exists x\varphi$ mit

$$\varphi = \bigwedge_{(-1\leq k\leq 1)(k)} \bigvee_{(1\leq l\leq 3)(l)} 2x > k + 100l + a.$$

Nach Satz 3.1.12 ist das Ergebnis der Elimination von $\exists x$ unter Anwendung obiger Optimierungsvorschläge die Formel

$$\varphi' = \bigvee_{(-1\leq k'\leq 1)(k')} \bigvee_{(1\leq l'\leq 3)(l')} \bigvee_{(-2\leq v-k'-100l'\leq 2)(v)} \bigwedge_{(-1\leq k\leq 1)(k)} \bigvee_{(1\leq l\leq 3)(l)} (v > k + 100l \wedge v + a \cong_2 0).$$

Bei genauer Untersuchung des Beispiels 3.1.15 fällt auf, daß bei der Auswertung der Formel, lediglich 15 Testpunkte tatsächlich durch virtuelle Substitution eingesetzt werden müssen. Dies ist im Vergleich zur Elimination der selben Formel mit dem Verfahren aus [Wei88], welches einen Substitutionsbereich von mehr als 400 Punkten liefert, ein deutlicher Fortschritt. Es empfiehlt sich daher für den Nutzer des Verfahrens für die Substitution der Punkte zunächst alle möglichen Werte der Terme t'_k, welche nur von den Bounds ψ'_i abhängen, zu bestimmen. Anschließend liefert die Auswertung der Formel γ_k sehr gute Ergebnisse, wie das Beispiel von oben zeigt. Eine solche Vorgehensweise liefert für den uniformen Fall *vermutlich* die kleinste mögliche Auswahl von symmetrischen Suchbereichen. Die Überprüfung dieser Vermutung würde den Rahmen dieser Arbeit sprengen und bleibt somit für weitere Forschung vorbehalten.

3.1.4 Erweiterte Quantorenelimination

Als nächstes wird gezeigt, wie man durch eine geringfügige Modifikation des Verfahrens Beispiele erfüllender Belegungen im Falle eines existentiell quantifizierten äußersten Quantorenblocks und Gegenbeispiele im Falle eines universell quantifizierten äußersten Quantorenblocks erhalten kann.

Eine parametrische Eliminationsmenge für $\exists x\varphi$ in obiger Konvention liefert für jedes $\mathbf{z} \in \mathbb{Z}^l$ durch $E(\mathbf{z})$ eine Menge von Testpunkten. Der Term eines jeden solchen Punktes (γ, t) ist ein Kandidat für die erfüllende Belegung der quantifizierten Variablen x unter der Bedingung, daß γ durch $\mathbf{z}$ erfüllt ist. Analog gilt bei der Elimination eines universellen Quantors $\forall x\varphi$ aufgrund der Äquivalenz $\forall x\varphi \sim \neg\exists x\neg\varphi$, daß ein Testterm in $E(\mathbf{z})$ ein Kandidat für ein Gegenbeispiel für die Gültigkeit von $\forall x\varphi$ ist.

Die Information über die Beschaffenheit der Testpunkte geht beim Anwenden der Eliminationsmenge, also bei der Bildung einer quantorenfreien äquivalenten Formel durch virtuelle Substitution, verloren. Es ist naheliegend die Testterme während der iterierten Anwendung der Elimination zu Speichern und als zusätzliche Information zur quantorenfreien Ergebnisformel auszugeben. Diese Modifikation des Verfahrens nennt man *Quantorenelimination mit Antworten* oder *erweiterte Quantorenelimination*.

Beispiel 3.1.16 Für die Elimination von $\exists x\varphi$ mit

$$\varphi = ax = b$$

ist $E = \{(true, b, \emptyset), (a \neq 0 \wedge b \cong_a 0, \frac{a}{b}, \emptyset)\}$ eine Eliminationsmenge. Es wäre naheliegend als Antwort der Elimination von $\exists x$ in $\exists x\varphi$ eine Menge M zu bezeichnen mit

$$M = \left[\begin{array}{c} (b = 0, \{x = 0\}), \\ (a \neq 0 \wedge b \cong_a 0, \{x = \frac{b}{a}\}) \end{array} \right].$$

Gilt dabei für ein Tupel $(\omega, T) \in M$ für eine Stelle $\mathbf{z} \in \mathbb{Z}^2$ für die erweiterte Formel $\omega(a, b)$

$$\mathbf{P} \models \omega(\mathbf{z}),$$

so findet man in T den passenden Term, der zu einer ganzen Zahl z ausgewertet werden kann, welche dann eine erfüllende Belegung von x in der erweiterten Formel $\varphi(a, b, x)$ mit $\mathbf{P} \models \varphi(\mathbf{z}, z)$ darstellt.

Bei der Umsetzung der erweiterten Quantorenelimination in der Presburger-Arithmetik ist es *nicht möglich* aufgrund der von Parametern abhängigen Eliminationsmengen die Beschaffenheit eines jeden Terms explizit abzuspeichern. Daher wird diese analog zu Eliminationsmengen ebenfalls durch Bounds parametrisiert.

Definition 3.1.17 (Antwort der Quantorenelimination) Als *Antwort* der Quantorenelimination von $\exists x$ in $\exists x\varphi$ nach obiger Konvention mit der Eliminationsmenge

$$E = \{(\gamma_i, t_i, ((\psi_i^1, v_i^1), ..., (\psi_i^{q_i}, v_i^{q_i}))) \mid i \in \{1, ..., p\}\}$$

wird eine Menge M bezeichnet mit

$$M = \left[\begin{array}{c} (\bigvee_{\psi_1^1(v_1^1)} ... \bigvee_{\psi_1^{q_1}(v_1^{q_1})} (\varphi[t_1 /\!\!/ x] \wedge \gamma_1), \{\psi_1^1, ..., \psi_1^{q_1}, x = t_1\}), \\ (\bigvee_{\psi_2^1(v_2^1)} ... \bigvee_{\psi_2^{q_2}(v_2^{q_2})} (\varphi[t_2 /\!\!/ x] \wedge \gamma_2), \{\psi_2^1, ..., \psi_2^{q_2}, x = t_2\}), \\ \vdots \\ (\bigvee_{\psi_p^1(v_p^1)} ... \bigvee_{\psi_p^{q_p}(v_p^{q_p})} (\varphi[t_p /\!\!/ x] \wedge \gamma_p), \{\psi_p^1, ..., \psi_p^{q_p}, x = t_p\}) \end{array} \right].$$

Offenbar ist die Disjunktion aller ersten Komponenten von M die quantorenfreie Ergebnisformel. Betrachte für ein $i \in \{1, ..., p\}$

$$\left(\bigvee_{\psi_i^1(v_i^1)} ... \bigvee_{\psi_i^{q_i}(v_i^{q_i})} (\varphi[t_i /\!\!/ x] \wedge \gamma_i), \{\psi_i^1, ..., \psi_i^{q_i}, x = t_i\} \right) \in M$$

mit $\mathbf{P} \models (\bigvee_{\psi_i^1(v_i^1)} ... \bigvee_{\psi_i^{q_i}(v_i^{q_i})} \varphi[t_i /\!\!/ x] \wedge \gamma_i)(\mathbf{z})$ für ein $\mathbf{z} \in \mathbb{Z}^l$. Dann gibt es ein $\mathbf{u} = (u_1, ..., u_{q_i}) \in \mathbb{Z}^{q_i}$ mit $u_j \in S(\psi_i^j(v_i^j))$ für $1 \leq j \leq q_i$ mit

$$\mathbf{P} \models (\varphi[t_i[u_1/v_1, ..., u_n/v_n] /\!\!/ x])(\mathbf{z}).$$

Auf Basis der Antwort der Quantorenelimination für die Elimination eines Quantors läßt sich bei iterierter Anwendung des Satzes 3.1.12 auf einen existentiellen Quantorenblock $\exists x_1 ... \exists x_m \varphi$ eine Menge der Form

$$M = \left[\begin{array}{c} (\omega_1, \Psi_1, \{x_1 = t_{11}, ..., x_m = t_{1m}\}), \\ (\omega_2, \Psi_2, \{x_1 = t_{21}, ..., x_m = t_{2m}\}), \\ \vdots \\ (\omega_p, \Psi_p, \{x_1 = t_{p1}, ..., x_m = t_{pm}\}) \end{array} \right]$$

mit ähnlicher Semantik angeben. Die Vereinigung der ersten Komponenten ergibt eine quantorenfreie zu $\exists x_1 ... \exists x_m \varphi$ äquivalente Formel. Ist für eine Stelle die erste Komponente eines Tupels $(\omega, T) \in M$ erfüllt, so liefert die zweite Komponente durch geeignete Auswertung der Bounds in Ψ_i eine endliche Menge von Belegungen von $x_1, ..., x_l$, von denen mindestens eine $\omega(x_1, ..., x_l)$ erfüllt. Analoges gilt auch für die Elimination eines universellen Quantorenblockes. Diese Vorgehensweise macht offensichtlich nur dann Sinn, wenn der betrachtete Quantorenblock der äußerste ist.

3.2 Strukturelle Eliminationsmengen

Das Verfahren in Satz 3.1.12 berücksichtigt die Eigenschaften der booleschen Struktur der Eingabeformel *nicht*. Die Bestimmung der Eliminationsmenge kann als eine Abbildung, die der Menge von atomaren Teilformeln von φ in $\exists x\varphi$ eine Eliminationsmenge zuordnet, angesehen werden. In diesem Abschnitt werden Techniken vorgestellt, welche die Struktur der Eingabeformel in den Eliminationsvorgang einbeziehen.

Betrachtet man die Formel φ in $\exists x\varphi$ mit $\varphi = \varphi_1 \vee \varphi_2$, so erkennt man, daß unter der Verwendung der Äquivalenz

$$\exists x\varphi \sim \exists x\varphi_1 \vee \exists x\varphi_2$$

die Bestimmung der Eliminationsmenge für $\exists x\varphi$ auf die Bestimmung der Eliminationsmengen für $\exists x\varphi_1$ und $\exists x\varphi_2$ zurückgeführt werden kann. Ist E_1 bzw. E_2 eine Eliminationsmenge für $\exists x\varphi_1$ bzw. $\exists x\varphi_2$, so ist eine Eliminationsmenge E für $\exists x\varphi$ definiert durch

$$E = E_1 \cup E_2.$$

Eine weitere Anwendung der Äquivalenz $\exists x\varphi \sim \exists x\varphi_1 \vee \exists x\varphi_2$ ist die auf das Ergebnis der Elimination eines existenziellen Quantors

$$\exists y \exists x \varphi \sim \bigvee_{i\in I} \exists y \bigvee_{\psi_i^1(v_i^1)} \ldots \bigvee_{\psi_i^{q_i}(v_i^{q_i})} (\varphi[t/\!/x] \wedge \gamma).$$

Bei einem Block von existenziellen Quantoren besteht zudem die Möglichkeit die Elimination durch Vertauschen der Reihenfolge der Quantoren zu beeinflussen. Die Anwendung obiger Regeln heißt *blockweise Elimination*. Blockweise Elimination bildet einen trivialen Spezialfall der Bildung von strukturellen Eliminationsmengen.

Für $\varphi = \varphi_1 \wedge \varphi_2$ in $\exists x\varphi$ ist im Allgemeinen weder die Vereinigung noch der Schnitt von Eliminationsmengen für $\exists x\varphi_1$ und $\exists x\varphi_2$ eine Eliminationsmenge für $\exists x\varphi$, wie das folgende Beispiel zeigt.

Beispiel 3.2.1 Sei $\varphi = (x \leq a \wedge x \geq b)$. Eine Eliminationsmenge für $\exists x(x \leq a)$ bzw. $\exists x(x \geq b)$ ist gegeben zum Beispiel durch

$$E_1 = \{(\text{true}, a-1, \emptyset)\} \quad \text{bzw.} \quad E_2 = \{(\text{true}, b+1, \emptyset)\}.$$

Weder der leere Schnitt noch die Vereinigung dieser Mengen ist eine Eliminationsmenge für die erweiterte Formel $\exists x\varphi$, denn für $(\exists x\varphi)(a,b)$ gilt $(\exists x\varphi)^{\mathbf{P}}(0,0) = \top$ aber $((-1 \leq 0 \wedge a - 1 \geq b) \vee (b+1 \leq a \wedge 1 \geq 0))^{\mathbf{P}}(0,0) = \bot$.

Es stellt sich die Frage, unter welchen Bedingungen man die Bestimmung einer Eliminationsmenge für $\exists x\varphi$ auf die Bestimmung von Eliminationsmengen für Formeln, die durch Umformungen aus $\exists x\varphi$ entstehen, zurückführen kann. Eliminationsmengen, die auf diese Weise gebildet werden, bezeichnet man als *strukturelle Eliminationsmengen*.

In den folgenden Abschnitten wird stets die Elimination von $\exists x$ in einer linear quantifizierten pränexen Formel der Form

$$\exists x \bigvee_{\psi_1(v_1)} \ldots \bigvee_{\psi_n(v_n)} \gamma$$

betrachtet, wobei φ positiv und stark quantorenfrei ist. Auf den Grund, warum die in diesem Abschnitt beschriebenen Techniken auf beliebige schwach quantorenfreie Formeln so *nicht* anwendbar sind, wird nach dem Beweis entsprechender Sätze hingewiesen. Die Verallgemeinerung von hier vorgestellten Techniken auf die Elimination von $\exists x$ in $\exists x\varphi$ für eine beliebige schwach quantorenfreie Formel φ stellt ein interessantes offenes Forschungsfeld dar. Im Folgenden wird also mit „φ in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren", falls nichts anderes explizit angemerkt wird, eine Formel in obiger Form mit einer positiven stark quantorenfreien Matrix verstanden.

3.2.1 Konjunktive Assoziiertheit

Eine mögliche Sicht auf die blockweise Elimination von $\exists x\varphi$ für $\varphi = \varphi_1 \vee \varphi_2$ ist die Ersetzung von φ_2 für die Bestimmung der Eliminationsmenge von $\exists x\varphi_1$ in $\exists x(\varphi_1 \vee \varphi_2)$ durch false und umgekehrt. In den folgenden Abschnitten wird untersucht, unter welchen Bedingungen Ersetzungen dieser Art zulässig sind.

Für folgende Betrachtungen ist es notwendig zwischen syntaktisch gleichen aber sich an unterschiedlichen Stellen in der Formel befindenden Teilformeln zu unterscheiden. Um dies auch formal handhaben zu können, wird der Begriff einer *markierten Teilformel* eingeführt.

Definition 3.2.2 (Markierte Teilformeln) Sei φ eine positive schwach quantorenfreie Formel. Die partielle Abbildung

$$\beta_\varphi : \bigcup_{i=0}^{\infty} \{0,1\}^i \to \mathcal{F}$$

sei durch folgende Eigenschaften definiert.

(i) $\beta_\varphi(\emptyset) = \varphi$.

(ii) Falls $\psi = \psi_0 \vee \psi_1$ bzw. $\psi = \psi_0 \wedge \psi_1$ eine Teilformel von φ mit $\beta_\varphi(\mathbf{a}) = \psi$ für $\mathbf{a} = (a_1, ..., a_k) \in \{0,1\}^k$ ist, so gilt $\beta_\varphi((a_1, ..., a_k, 0)) = \psi_0$ und $\beta_\varphi((a_1, ..., a_k, 1)) = \psi_1$.

(iii) Falls $\psi = \bigvee_{\psi(v)} \psi_0$ bzw. $\psi = \bigwedge_{\psi(v)} \psi_0$ mit $\beta_\varphi(\mathbf{a}) = \psi$ für $\mathbf{a} = (a_1, ..., a_k) \in \{0,1\}^k$, so gilt

$$\beta_\varphi((a_1, ..., a_k, 0)) = \psi_0.$$

Das Tupel $(\varphi, \mathbf{a})$ mit $\beta_\varphi(\mathbf{a}) = \psi$ wird mit $\psi_\varphi^{\mathbf{a}}$ abgekürzt und heißt *markierte Teilformel* von φ. Zwei syntaktisch gleiche Teilformeln, die sich an verschiedenen Stellen in φ befinden, bekommen unterschiedliche Markierungen zugewiesen. Bei der Angabe einer markierten Teilformel $(\varphi, \mathbf{a})$ durch $\psi_\varphi^{\mathbf{a}}$ wird im Folgenden das Wort „markiert" und zusätzlich auch der Index φ, falls φ aus dem Kontext bekannt ist, weggelassen. Eine markierte Teilformel $\psi_1^{\mathbf{a}}$ von φ heißt eine *Teilformel* von $\psi_2^{\mathbf{b}}$, falls ψ_1 eine Teilformel von ψ_2 ist und $\mathbf{b} = (b_1, ..., b_i) = (a_1, ..., a_i)$ für $\mathbf{a} = (a_1, ..., a_i, ..., a_k)$. Die markierte Formel $\psi_2^{\mathbf{b}}$ wird dann als *Oberformel* von $\psi_1^{\mathbf{a}}$ bezeichnet.

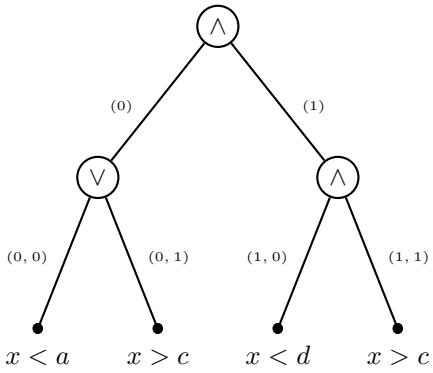


Abbildung 3.2: Baumdarstellung der Formel $\varphi = (x < a \vee x > c) \wedge (x < d \wedge x > c)$ mit zugehörigen Markierungen

Beispiel 3.2.3 Betrachte die Formel $\varphi = (x < a \vee x > c) \wedge (x < d \wedge x > c)$ veranschaulicht in Abbildung 3.2.

(i) $x < a \vee x > c$ besitzt in φ die Markierung (0), $x < d \wedge x > c$ die Markierung (1), $x < a$ die Markierung $(0,0)$ und $x < d$ die Markierung $(1,0)$.

(ii) Für die Formel $x > c$ gibt es zwei *verschiedene* markierte Formeln $(x > c)^{(0,1)}$ und $(x > c)^{(1,1)}$.

(iii) $(x < a)^{(0,0)}$ ist eine Teilformel von $(x < a \vee x > c)^{(0)}$. Alle markierten Teilformeln sind Teilformeln von $\varphi^{\emptyset}$.

(iv) $(x > c)^{(1,1)}$ ist *keine* Teilformel von $(x < a \vee x > c)^{(0)}$.

Definition 3.2.4 (Konjunktive Assoziiertheit) Sei $\varphi \in \mathcal{F}$ eine positive schwach quantorenfreie Formel. Zwei markierte Teilformeln $\psi_1^{\mathbf{a}}$ und $\psi_2^{\mathbf{b}}$ von φ heißen *konjunktiv assoziiert*,

(i) falls $\psi_1^{\mathbf{a}}$ oder $\psi_2^{\mathbf{b}}$ eine Teilformel von der anderen ist,

(ii) falls eine markierte Teilformel $\varphi^{\mathbf{c}}$ mit $\varphi = \varphi_1 \wedge \varphi_2$ existiert, sodaß $\psi_1^{\mathbf{a}}$ eine Teilformel von $\varphi_1^{(\mathbf{c},0)}$ ist und $\psi_2^{\mathbf{b}}$ eine Teilformel von $\varphi_2^{(\mathbf{c},1)}$ ist oder $\psi_1^{\mathbf{a}}$ eine Teilformel von $\varphi_2^{(\mathbf{c},1)}$ ist und $\psi_2^{\mathbf{b}}$ eine Teilformel von $\varphi_1^{(\mathbf{c},0)}$ ist oder

(iii) falls $\psi_1^{\mathbf{a}}$ und $\psi_2^{\mathbf{b}}$ einen universellen gebundenen Quantor als eine gemeinsame Oberformel haben.

Offenbar gilt für zwei *nicht konjunktiv* assoziierte Teilformeln $\psi_1^{\mathbf{a}}$ und $\psi_2^{\mathbf{b}}$ einer Formel φ in $\exists x \varphi$ stets, daß $\psi_1^{\mathbf{a}}$ keine Teilformel von $\psi_2^{\mathbf{b}}$ ist und umgekehrt.

Um die Begriffsdefinition zu verdeutlichen, folgen für jeden der in der Definition aufgeführten Fälle jeweils einige Beispiele. Man vergleiche dabei die intuitive Vorstellung der konjunktiven Assoziiertheit, etwa die Notwendigkeit der Gültigkeit beider konjunktiv assoziierter Teilformeln von φ für die Gültigkeit von φ für bestimmte Belegungen der Parameter, mit den folgenden Ergebnissen.

Beispiele 3.2.5 Betrachte folgende Formeln, veranschaulicht in Abbildung 3.3.

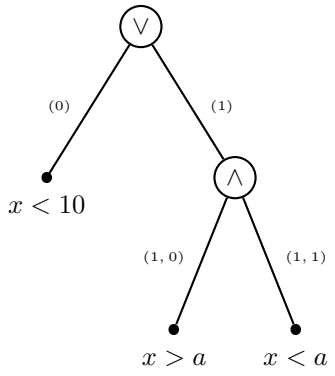
(i) Für $\varphi_1 = (x < 10 \vee (x > a \wedge x < a))$ sind $\varphi^{\emptyset}$ und $(x > a \wedge x < a)^{(1)}$ mit entsprechenden Markierungen konjunktiv assoziiert, obwohl die zweite Teilformel offensichtlich nicht erfüllbar ist.

(ii) Für $\varphi_2 = (x < a \wedge (x = b \vee (x < c \wedge x = b)))$ sind die beiden Teilformeln $(x = b)^{(1,0)}$ und $(x = b)^{(1,1,1)}$ nicht konjunktiv assoziiert.

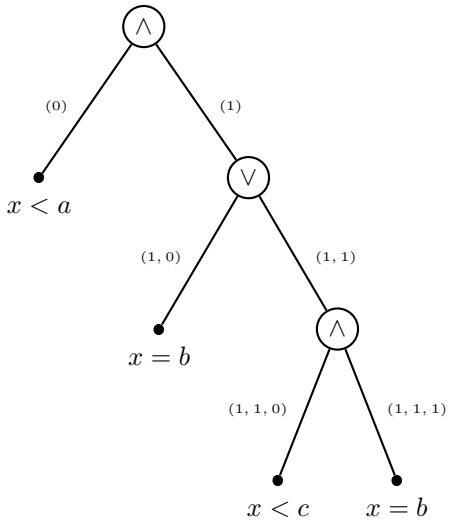
(iii) Für $\varphi_3 = ((x < a \vee x > b) \wedge x = c)$ sind sowohl die Teilformeln $(x < a)^{(0,0)}$ und $(x = c)^{(1)}$ als auch die Teilformeln $(x > b)^{(0,1)}$ und $(x = c)^{(1)}$ konjunktiv assoziiert, aber die Teilformeln $(x < a)^{(0,0)}$ und $(x > b)^{(0,1)}$ sind *nicht* konjunktiv assoziiert.

(iv) Für $\varphi_4 = \bigwedge_{\psi(k)}(x \leq k \vee x > k)$ mit $\psi = (-2 \leq k \leq 2)$ sind die Formeln $(x \leq k)^{(0,0)}$ und $(x > k)^{(0,1)}$ *konjunktiv assoziiert.*

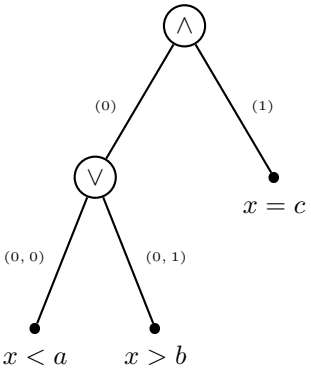
Aus Beispiel 3.2.5(i) folgt, daß die oben angegebene anschauliche Auslegung der konjunktiven Assoziiertheit nicht korrekt ist, denn eine nicht erfüllbare Teilformel von φ kann nicht für die Erfüllbarkeit der als positiv vorausgesetzten Oberformel φ notwendig sein. Das Beispiel 3.2.5(ii) zeigt, daß zwei identische Formeln mit verschiedenen Markierungen nicht konjunktiv assoziiert sein müssen. Diese Beobachtung ist der Grund für die Einführung von markierten Formeln. Aus Beispiel 3.2.5(iii) folgt, daß der Begriff „konjunktive Assoziiertheit" offenbar keine Äquivalenzrelation auf der Menge der markierten Teilformeln einer Formel induziert. Die Eigenschaften „Reflexivität" und „Symmetrie" der entsprechenden Relation sind allerdings erfüllt, was gerade die Möglichkeit erlaubt „$\psi^{\mathbf{a}}$ und $\psi^{\mathbf{b}}$ sind konjunktiv assoziiert" zu sagen. Das wäre im Falle einer nicht symmetrischen Relation nicht möglich. Das Beispiel 3.2.5(iv) zeigt, daß konjunktive Assoziiertheit nach obiger Definition nur für schwach quantorenfreie Formeln *ohne* universeller gebundener Quantoren



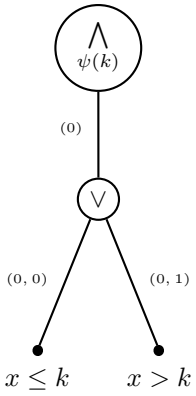
(i) $\varphi_1 = (x < 10 \vee (x > a \wedge x < a))$



(ii) $\varphi_2 = (x < a \wedge (x = b \vee (x < c \wedge x = b)))$



(iii) $\varphi_3 = ((x < a \vee x > b) \wedge x = c)$



(iv) $\varphi_4 = \bigwedge_{\psi(k)}(x \leq k \vee x > k)$

Abbildung 3.3: Baumdarstellung der Formeln aus Beispiel 3.2.5

eingesetzt werden kann. Der Grund für den Fall (iii) in Definition 3.2.4 wird später aus Beispiel 3.2.17 ersichtlich.

Es stellt sich die Frage nach einem Verfahren konjunktiv assoziierte bzw. nicht konjunktiv assoziierte Formeln zu einer gegebenen markierten Teilformel $\psi^{\mathbf{a}}$ von φ zu entdecken. Dies kann algorithmisch effizient realisiert werden, wie das folgende Lemma zeigt.

Lemma 3.2.6 Sei φ eine Formel in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei $\psi^{\mathbf{a}}$ mit $\mathbf{a} = (a_1, ..., a_k)$ eine markierte Teilformel von φ.

(i) Jede Teilformel $\omega^{\mathbf{v}}$ einer markierten Teilformel $\gamma^{(a_1,...,a_j,b)}$ mit $j < k$ und $b \neq a_{j+1}$, die in einer Konjunktion $\gamma \wedge \gamma'$ vorkommt ist zu φ *konjunktiv* assoziiert.

(ii) Jede Teilformel $\omega^{\mathbf{v}}$ einer markierten Teilformel $\gamma^{(a_1,...,a_j,b)}$ mit $j < k$ und $b \neq a_{j+1}$, die in einer Disjunktion $\gamma \vee \gamma'$ vorkommt ist zu φ *nicht konjunktiv* assoziiert.

Beweis: (i) Offenbar ist die Voraussetzung aus Definition 3.2.4(ii) für jedes $\omega^{\mathbf{v}}$ erfüllt. (ii) Analog dazu ist für jedes $\omega^{\mathbf{v}}$ keine der beiden Voraussetzungen aus Definition 3.2.4 erfüllt. □

Beispiel 3.2.7 Zu bestimmen sind alle konjunktiv assoziierte bzw. nicht konjunktiv assoziierte Formeln zu $\psi^{(1,0,1,0)}$ in

$$\varphi = (\psi_1 \wedge ((\psi_3 \wedge (\psi \vee \psi_4)) \vee \psi_2)).$$

Neben allen Oberformeln von $\psi^{(1,0,1,0)}$ sind in Abbildung 3.4 ausgefüllt dargestellte Teilformeln $\psi_1^{(0)}$ und $\psi_3^{(1,0,0)}$ zu $\psi^{(1,0,1,0)}$ konjunktiv assoziiert. Die Formeln $\psi_2^{(1,1)}$ und $\psi_4^{(1,0,1,1)}$ sind hingegen zu $\psi^{(1,0,1,0)}$ nicht konjunktiv assoziiert. Der Weg von der Wurzel zu $\psi^{(1,0,1,0)}$ ist in Abbildung 3.4 durch Strichlinien dargestellt.

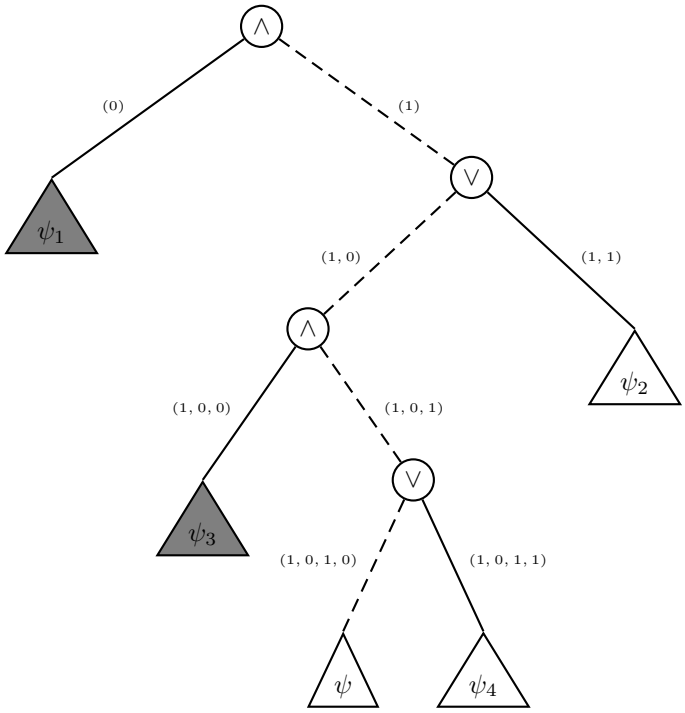


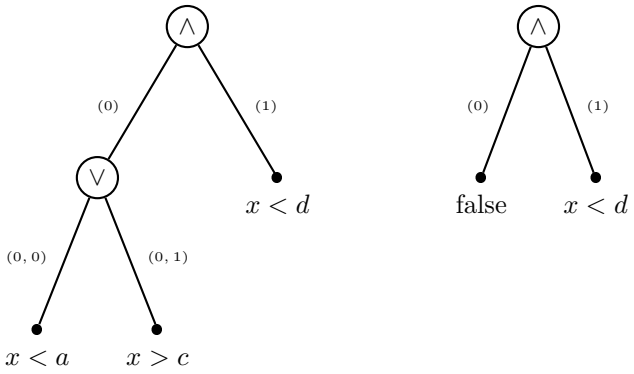
Abbildung 3.4: Bestimmung von zu $\psi^{(1,0,1,0)}$ konjunktiv assoziierten bzw. nicht konjunktiv assoziierten Formeln in $\varphi = (\psi_1 \wedge ((\psi_3 \wedge (\psi \vee \psi_4)) \vee \psi_2))$

3.2.2 Condensing Operator und Eliminationsmengen

Im Folgenden wird das Ziel verfolgt Eliminationsmengen aus Vereinigungen von Eliminationsmengen von Formeln, die durch Ersetzen ausgewählter markierter Teilformeln durch false entstehen, zu bilden. Der zugrunde liegende Ersetzungsvorgang wird als *Condensing* bezeichnet.

Definition 3.2.8 (Condensing-Operator) Sei φ eine positive schwach quantorenfreie Formel und M eine Menge von Markierungen. Dann bezeichnet man mit $\Gamma_M(\varphi)$ die Formel, die aus φ entsteht, indem man für jede Markierung $\mathbf{a} \in M$ alle markierten Teilformeln von φ mit der Markierung $\mathbf{a}$ durch false ersetzt. Die entsprechende Abbildung zwischen Formeln heißt dann der *Condensing-Operator* Γ_M.

Die Definition des Condensing-Operators Γ_M ist so ausgelegt, daß Teilformeln nicht anhand der Gleichheit ihrer Zeichenketten, sondern lediglich anhand ihrer Positionen in der Eingabeformel durch false ersetzt werden. Die Anwendung des Condensing-Operators wird in Abbildung 3.5 veranschaulicht. Falls die entsprechende Teilformel in φ nicht vorhanden ist, wie etwa die Teilformel mit der Markierung $(1,1)$ in Abbildung 3.5, so werden auch keine Teilformeln mit dieser Markierung durch false ersetzt.



(i) $\varphi = ((x < a \vee x > c) \wedge x < d)$ (ii) $\Gamma_M(\varphi) = (\text{false} \wedge x < d)$

Abbildung 3.5: Anwendung des Condensing-Operators Γ_M für $M = \{(0), (1, 1)\}$

Lemma 3.2.9 Sei φ eine positive schwach quantorenfreie Formel. Sei weiter M eine Menge von Markierungen und sei $\mathbf{a}$ eine weitere Markierung. Dann gilt

$$\Gamma_{M \cup \{\mathbf{a}\}}(\varphi) = \Gamma_{\{\mathbf{a}\}}(\Gamma_M(\varphi)).$$

Das Lemma 3.2.9 zeigt, daß Condensing iterativ in beliebiger Reihenfolge durchgeführt werden kann. Es stellt sich heraus, daß die intuitive Bedeutung der konjunktiven Assoziiertheit vor allem an dem Begriff „notwendig" scheitert. In folgender Definition wird die Aussage „eine Teilformel von φ ist für die Erfüllung von φ notwendig" mit Hilfe des Condensing-Operators präzise formuliert.

Definition 3.2.10 (Notwendig und hinreichend für Erfüllbarkeit) Sei $\varphi \in \mathcal{F}$ eine Formel in $\exists x \varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei weiter $\psi^{\mathbf{a}}$ eine markierte Teilformel von φ.

(i) Die Formel $\psi^{\mathbf{a}}$ heißt *notwendig* für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an der Stelle $\mathbf{z} \in \mathbb{Z}^l$, falls gilt

$$\mathbf{P} \models (\varphi \wedge \neg \Gamma_{\{\mathbf{a}\}}(\varphi))(\mathbf{z}).$$

(ii) Die Formel $\psi^{\mathbf{a}}$ heißt *hinreichend* für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an der Stelle $\mathbf{z} \in \mathbb{Z}^l$, falls ein $\mathbf{u} \in \mathbb{Z}^n$ existiert, sodaß für jede markierte Oberformel $\gamma^{\mathbf{b}}$ von $\psi^{\mathbf{a}}$ für die erweiterte Formel $\gamma(x_1, ..., x_l, v_1, ..., v_n)$ gilt

$$\mathbf{P} \models \gamma(\mathbf{z}, \mathbf{u}).$$

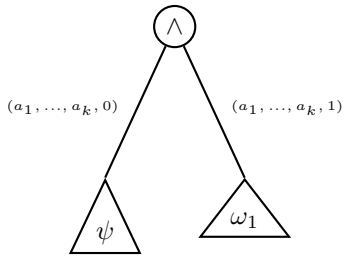
Man beachte bei der Definition des Begriffes „hinreichend für Erfüllbarkeit", daß per Definition auch $\psi^{\mathbf{a}}$ und $\varphi^{\emptyset}$ markierte Oberformeln von $\psi^{\mathbf{a}}$ sind. Die beiden in Definition 3.2.10 eingeführten Begriffe sollten *nicht* mit logischen Begriffen „notwendig" und „hinreichend" verwechselt werden. Der Zusammenhang zwischen den beiden Begriffen wird durch das folgende Lemma zunächst für stark quantorenfreie Formeln formuliert.

Lemma 3.2.11 Sei φ eine positive stark quantorenfreie Formel und sei $\psi^{\mathbf{a}}$ mit $\mathbf{a} = (a_1, ..., a_k)$ eine markierte Teilformel von φ. Ist $\psi^{\mathbf{a}}$ für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an einer Stelle $\mathbf{z} \in \mathbb{Z}^l$ notwendig, so ist $\psi^{\mathbf{a}}$ für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an der Stelle $\mathbf{z}$ auch hinreichend.

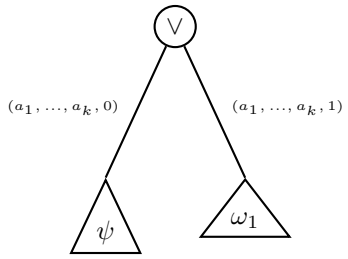
Beweis: Der Beweis wird durch Induktion nach k geführt. Falls $k = 0$, so ist die Behauptung wegen $\psi = \varphi$ trivial. Sei die Aussage nun für ein $k \in \mathbb{N}_0$ bewiesen. Betrachte für den Fall $\mathbf{a} = (a_1, ..., a_{k+1})$ die „Vaterformel von $\psi^{\mathbf{a}}$ in φ"

$$\omega = \beta_\varphi((a_1, ..., a_k)).$$

Sei o.B.d.A. $a_{k+1} = 0$. Die Teilformel ω ist dann von der Form $\psi \, \rho \, \omega_1$ mit $\rho \in \{\wedge, \vee\}$.



(i) $\omega = \psi \wedge \omega_1$



(ii) $\omega = \psi \vee \omega_1$

Es ist nicht schwer zu sehen, daß in jedem der zwei Fälle aufgrund der Voraussetzung $\mathbf{P} \models (\neg\Gamma_{\{\psi^{\mathbf{a}}\}}(\varphi))(\mathbf{z})$ und aufgrund der Positivität von φ gilt

$$\mathbf{P} \models (\varphi \wedge \neg\Gamma_{\{(a_1,\ldots,a_k)\}}(\varphi))(\mathbf{z}),$$

da man durch Ersetzen von ω durch false speziell auch ψ durch false ersetzt. Somit gilt nach Induktionsannahme die Behauptung für $\omega^{(a_1,\ldots,a_k)}$. Die Oberformeln von $\omega^{(a_1,\ldots,a_k)}$ und $\psi^{(a_1,\ldots,a_{k+1})}$ stimmen nach Definition bis auf $\psi^{(a_1,\ldots,a_{k+1})}$ überein. Somit ist die Behauptung nur noch für $\psi^{(a_1,\ldots,a_{k+1})}$ zu zeigen. Falls ω, wie im Fall (i), eine Konjunktion ist, so gilt trivialerweise $\mathbf{P} \models \psi(\mathbf{z})$, denn Gegenteiliges wäre ein Widerspruch zu der nach Induktionsannahme geltenden Aussage $\mathbf{P} \models \omega(\mathbf{z})$. Betrachte nun den Fall (ii), daß ω eine Disjunktion ist. Angenommen es gilt $\mathbf{P} \not\models \psi(\mathbf{z})$, woraus unmittelbar $\mathbf{P} \models \omega_1(\mathbf{z})$ folgt. Dies ist allerdings ein Widerspruch zu $\mathbf{P} \models (\neg\Gamma_{\{\mathbf{a}\}}(\varphi))(\mathbf{z})$, denn $\mathbf{z}$ ist mit

$$\mathbf{P} \models (\text{false} \vee \omega_1)(\mathbf{z})$$

gerade eine erfüllende Stelle für $\Gamma_{\{\mathbf{a}\}}(\varphi)$. □

Für Formeln mit ausschließlich existenziellen gebundenen Quantoren kann die Aussage des Lemmas 3.2.11 sinngemäß übernommen werden. Davon wird hier jedoch abgesehen.

Beispiel 3.2.12 Man betrachte das Beispiel 3.2.5(i). Die markierte Teilformel $(x > a \wedge x < a)^{(1)}$ ist für keine Stelle $\mathbf{z} \in \mathbb{Z}^2$ für die Erfüllbarkeit von $\varphi_1(x, a)$ notwendig, denn offenbar ist

$$\mathbf{P} \models (\varphi \wedge \neg\Gamma_{\{(1)\}}(\varphi))(\mathbf{z})$$

für keine Stelle $\mathbf{z} \in \mathbb{Z}^2$ erfüllt.

Die Umkehrung des Lemmas 3.2.11 ist im Allgemeinen falsch. Betrachte dazu die Formel $\varphi = (x = 0 \vee x = 0)$. Für die Stelle 0 sind beide markierten Teilformeln $(x = 0)^{(0)}$ und $(x = 0)^{(1)}$ für die Erfüllbarkeit von $\varphi(x)$ hinreichend aber jeweils nicht notwendig.

Lemma 3.2.13 Sei φ eine positive schwach quantorenfreie Formel. Es gilt für jede Menge von Markierungen

$$\mathbf{P} \models \Gamma_M(\varphi) \longrightarrow \varphi.$$

Die Umkehrung des Lemmas 3.2.13 ist im Allgemeinen falsch, wie man bereits an $\varphi = \text{true}$ und $M = \{\emptyset\}$ sieht. Mit Hilfe des Begriffs der konjunktiven Assoziiertheit kann man aber Bedingungen angeben, unter welchen eine zur Umkehrung der Bemerkung 3.2.13 ähnliche Beziehung wahr ist.

Satz 3.2.14 (Hauptsatz über Condensing) *Sei φ eine positive stark quantorenfreie Formel und $\psi_1^{\mathbf{a}}$ mit $\mathbf{a} = (a_1, ..., a_k)$ eine für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an der Stelle $\mathbf{z} \in \mathbb{Z}^l$ hinreichende markierte Teilformel von φ.*

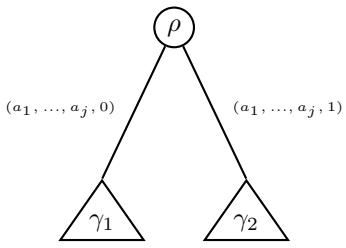
(i) Für jede zu $\psi_1^{\mathbf{a}}$ nicht konjunktiv assoziierte markierte Teilformel $\psi_1^{\mathbf{b}}$ von φ gilt

$$\mathbf{P} \models (\Gamma_{\{\mathbf{b}\}}(\varphi))(\mathbf{z}).$$

(ii) Für jede Teilmenge von zu $\psi_1^{\mathbf{a}}$ nicht konjunktiv assoziierten Teilformeln $\{\psi_i^{\mathbf{b}_i} \mid i \in I\}$ von φ für $M = \{\mathbf{b}_i \mid i \in I\}$ gilt

$$\mathbf{P} \models (\Gamma_M(\varphi))(\mathbf{z}).$$

Beweis: (i) Angenommen es gilt $\mathbf{P} \not\models (\Gamma_{\{\mathbf{b}\}}(\varphi))(\mathbf{z})$. Nach Definition 3.2.4 ist $\psi_1^{\mathbf{a}}$ keine Teilformel von $\psi_2^{\mathbf{b}}$ und umgekehrt. Daher gibt es nach Definition einer Presburger-Formel 1.1.3 eine markierte Oberformel $\gamma^{\mathbf{g}}$ von $\psi_1^{\mathbf{a}}$ mit $\gamma = (\gamma_1 \; \rho \; \gamma_2)$ mit markierten Formeln $\gamma_1^{\mathbf{e}}$ und $\gamma_2^{\mathbf{d}}$ mit $\rho \in \{\wedge, \vee\}$, sodaß $\psi_1^{\mathbf{a}}$ eine echte Teilformel von $\gamma_1^{\mathbf{e}}$ und $\psi_2^{\mathbf{b}}$ eine echte Teilformel von $\gamma_2^{\mathbf{d}}$ ist oder umgekehrt. Sei o.B.d.A. $\gamma = \beta_\varphi(a_1, ..., a_j)$ und $a_{j+1} = 0$, womit ersteres gilt.



Da $\psi_1^{\mathbf{a}}$ nach Voraussetzung für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an der Stelle $\mathbf{z}$ hinreichend ist, gilt nach Definition 3.2.10(ii) $\mathbf{P} \vDash \gamma(\mathbf{z})$ und zugleich $\mathbf{P} \vDash \gamma_1(\mathbf{z})$, da beide Teilformeln mit entsprechenden Markierungen markierte Oberformeln von $\psi^{\mathbf{a}}$ sind. Falls ρ das Symbol „$\vee$" ist, ergibt sich allerdings ein Widerspruch zur Annahme $\mathbf{P} \not\models (\Gamma_{\{\mathbf{b}\}}(\varphi))(\mathbf{z})$, denn die Ersetzung von $\psi_2^{\mathbf{b}}$ durch false nichts am Wahrheitswert $\gamma^{\mathbf{P}}(\mathbf{z})$ und somit auch wegen der Positivität von φ am Wahrheitswert von

$$\varphi^{\mathbf{P}}(\mathbf{z}) = (\Gamma_{\{\mathbf{b}\}}(\varphi))^{\mathbf{P}}(\mathbf{z}) = \top$$

ändert. Somit muß γ eine Konjunktion sein, woraus unmittelbar die konjunktive Assoziiertheit von $\psi_1^{\mathbf{a}}$ und $\psi_2^{\mathbf{b}}$ folgt. Das ist offenbar auch ein Widerspruch zur Annahme. (ii) Der Beweis wird durch Induktion nach $i = |M|$ geführt. Für $i = 0$ ist die Behauptung trivial. Für $i = 1$ ist die Behauptung gerade der Fall (i). Sei für $i \in \mathbb{N}_0$ die Behauptung bewiesen. Für $|M| = i + 1$ gilt für eine i-elementige Teilmenge $N \subseteq M$ nach Induktionsannahme

$$\mathbf{P} \models (\Gamma_N(\varphi))(\mathbf{z}).$$

Da $\Gamma_N(\varphi)$ stark quantorenfrei und positiv ist, folgt für $\psi_1^{\mathbf{a}}$ und $\psi_2^{\mathbf{b}} \in (M \setminus N)$ nach (i), da $\psi_1^{\mathbf{a}}$ und $\psi_2^{\mathbf{b}}$ in $\Gamma_N(\varphi)$ ebenfalls nicht konjunktiv assoziiert sind

$$\mathbf{P} \models (\Gamma_{\{\mathbf{b}\}}(\Gamma_N(\varphi)))(\mathbf{z}).$$

Mit $\Gamma_{\{\mathbf{b}\}}(\Gamma_N(\varphi)) = \Gamma_M(\varphi)$ nach Lemma 3.2.9 folgt die Behauptung für M. □

Es wird nun gezeigt, wie die Aussage des Satzes 3.2.14 auf schwach quantorenfreie Formeln mit ausschließlich existentiellen gebundenen Quantoren erweitert werden kann.

Lemma 3.2.15 Sei φ eine Formel in $\exists x \varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei $\psi_1^{\mathbf{a}}$ eine für die Erfüllbarkeit von $\varphi(x_1, ..., x_l)$ an der Stelle $\mathbf{z} \in \mathbb{Z}^l$ *hinreichende* markierte Teilformel von φ. Dann gilt für jede Teilmenge von zu $\psi_1^{\mathbf{a}}$ nicht konjunktiv assoziierten Teilformeln $\{\psi_i^{\mathbf{b}_i} \mid i \in I\}$ von φ für $M = \{\mathbf{b}_i \mid i \in I\}$

$$\mathbf{P} \models (\Gamma_M(\varphi))(\mathbf{z}).$$

Beweis: Offenbar sind Markierungen von gebundenen Quantoren in M aufgrund der Definition der konjunktiven Assoziiertheit nicht enthalten. Der Beweis der Aussage wird auf Satz 3.2.14 zurückgeführt. Sei $\mathbf{a} = (a_{l+1}, ..., a_k)$. Offenbar ist $\psi^{\mathbf{a}}$ für die erweiterte stark quantorenfreie Formel $\gamma(x_1, ..., x_l, v_1, ..., v_n)$ nach Definition 3.2.10(ii) an der Stelle $(\mathbf{z}, \mathbf{u})$ für ein $\mathbf{u} \in \mathbb{Z}^n$ hinreichend für

die Erfüllbarkeit. Setze $\mathbf{b}'_i = (b_{l+1}, ..., b_j)$ für $\mathbf{b}_i = (b_1, ..., b_j)$ und $M' = \{b'_i \mid i \in I\}$ Es folgt dann nach Satz 3.2.14(ii), daß auch

$$\mathbf{P} \models (\Gamma_{M'}(\gamma))(\mathbf{z}, \mathbf{u})$$

gilt. Daraus folgt die Behauptung wegen

$$\Gamma_M(\varphi) = \bigvee_{\psi_1(v_1)} ... \bigvee_{\psi_l(v_l)} \Gamma_{M'}(\gamma).\square$$

Da aus der Eigenschaft „notwendig für Erfüllbarkeit" nach Lemma 3.2.11 auch die Eigenschaft „hinreichend für Erfüllbarkeit" folgt, kann man den Satz 3.2.14 unmittelbar für Formeln anwenden, die „notwendig für Erfüllbarkeit" sind.

Korollar 3.2.16 Sei φ eine Formel in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren und sei $\psi_1^{\mathbf{a}}$ markierte Teilformel von φ. Dann gilt für jede Teilmenge von zu $\psi_1^{\mathbf{a}}$ nicht konjunktiv assoziierten Teilformeln $\{\psi_i^{\mathbf{b}_i} \mid i \in I\}$ von φ für $M = \{\mathbf{b}_i \mid i \in I\}$

$$\mathbf{P} \models (\varphi \wedge \neg\Gamma_{\{\psi_i^{\mathbf{a}}\}}(\varphi)) \longrightarrow \Gamma_M(\varphi).$$

Es ist naheliegend eine ähnliche Behauptung für beliebige schwach quantorenfreie Formeln zu formulieren. Allerdings läßt sich das Lemma 3.2.15 nicht unmittelbar für beliebige schwach quantorenfreie Formeln übernehmen. Das wird aus dem folgenden Beispiel deutlich.

Beispiel 3.2.17 Man betrachte die schwach quantorenfreie pränexe Formel

$$\varphi = \bigwedge_{(-2 \leq k \leq 2)(k)} (x \geq k \vee x < k).$$

Es gilt offenbar $\varphi \sim \text{true}$. Man betrachte $\varphi(x)$ und die Stelle 0. Offenbar gilt

$$\mathbf{P} \models (\varphi \wedge \neg\Gamma_{\{(1)\}}(\varphi))(0).$$

Allerdings gilt $(w \geq k)(k, x)$ nur für Belegungen $(w, 0)$, mit $w \leq 0$. Somit gilt auch $\mathbf{P} \not\models (\Gamma_{\{(0)\}}(\varphi))(0)$.

Für Formeln mit universellen gebundenen Quantoren hängt die Erfüllbarkeit von Oberformeln einer markierten Teilformel zusätzlich von den Auswertungen der Bounds universeller gebundener Quantoren ab. Dieses Verhalten kann man intuitiv in Termini der Definition 3.2.10 so formulieren, daß eine markierte Teilformel *nur für bestimmte Auswertungen* der Bound-Variablen notwendig bzw. hinreichend für die Erfüllbarkeit ist. Möchte man das Lemma 3.2.15 auf beliebige schwach quantorenfreie Formeln übertragen, so ist es naheliegend zur Expansion universeller gebundener Quantoren überzugehen. Dies ist allerdings aufgrund parametrischer Bounds im allgemeinen nicht möglich. Dies ist ein Grund dafür, daß in diesem Abschnitt lediglich schwach quantorenfreie Formeln mit ausschließlich existentiellen gebundenen Quantoren betrachtet werden.

Im Folgenden ist man daran interessiert, die Menge M für die Anwendung des Korollars 3.2.16 bzw. 3.2.14 möglichst groß zu wählen. Daher ist es sinnvoll für die zu einer Teilformel größte Menge von konjunktiv bzw. nicht konjunktiv assoziierten Teilformeln, nämlich für die Menge *aller* Teilformeln mit entsprechender Eigenschaft, eine eigene Bezeichnung einzuführen.

Definition 3.2.18 Sei φ eine positive schwach quantorenfreie Formel und sei $\psi^{\mathbf{a}}$ eine markierte Teilformel von φ. Die Menge der Markierungen aller zu $\psi^{\mathbf{a}}$ konjunktiv assoziierten bzw. nicht konjunktiv assoziierten markierten Teilformeln von φ bezeichnet man mit

$$L(\mathbf{a}) \text{ bzw. } \overline{L}(\mathbf{a}).$$

Nun kann das Hauptergebnis dieses Abschnittes vorgestellt werden. Der folgende Satz zeigt, wie die Bestimmung der Eliminationsmenge einer Formel $\exists x\varphi$ auf die Bestimmung von Eliminationsmengen für Formeln der Form $\exists x\Gamma_{M_i}(\varphi)$, die durch die Anwendung des Condensing-Operators aus φ entstehen, zurückgeführt werden kann.

Satz 3.2.19 (Strukturelle Eliminationsmengen) *Sei φ eine Formel in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei $\{\psi_i^{\mathbf{a}_i} \mid i \in I\}$ eine Menge von markierten Teilformeln von φ und $M = \{\mathbf{a}_i \mid i \in I\}$ die Menge entsprechender Markierungen. Bezeichne E^* eine Eliminationsmenge für die Formel $\exists x\Gamma_M(\varphi)$ und E_i eine Eliminationsmenge für die Formel $\exists x\Gamma_{\overline{L}(\mathbf{a}_i)}$. Dann ist*

$$E = E^* \cup \bigcup_{i\in I} E_i$$

eine Eliminationsmenge für $\exists x\varphi$.

Beweis: Gelte zunächst $\mathbf{P} \models (\exists x\varphi)(\mathbf{z})$. Gelte etwa $\mathbf{P} \models \varphi(\mathbf{z}, z)$. Falls $\mathbf{P} \models (\Gamma_M(\varphi))(\mathbf{z}, z)$ gilt, so gilt auch

$$\mathbf{P} \models (\exists x\Gamma_M(\varphi))(\mathbf{z}).$$

Da E^* eine Eliminationsmenge für $\exists x\Gamma_M(\varphi)$ ist, gilt dann auch

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

und somit $\mathbf{P} \models (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma)(\mathbf{z})$ für ein (γ, t) in $E^*(\mathbf{z})$. Mit Bemerkung 3.2.13 gilt dann auch $\mathbf{P} \models (\varphi[t/\!/x] \wedge \gamma)(\mathbf{z})$ und somit

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E^*(\mathbf{z})} (\varphi[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Daraus folgt die Behauptung für den betrachteten Fall. Falls hingegen $\mathbf{P} \not\models (\Gamma_M(\varphi))(\mathbf{z}, z)$, so gibt es eine Teilmenge von markierten Formeln $N \subseteq M$ und eine markierte Teilformel $\psi_i^{\mathbf{a}_i}$, sodaß gilt

$$\mathbf{P} \models (\Gamma_N(\varphi) \wedge \neg\Gamma_{\{\mathbf{a}_i\}}(\Gamma_N(\varphi)))(\mathbf{z}, z).$$

Daraus folgt mit Korollar 3.2.16 $\mathbf{P} \models (\Gamma_{\overline{L}(\mathbf{a}_i)}(\Gamma_N(\varphi)))(\mathbf{z}, z)$. Offensichtlich ist E_i aufgrund der Positivität von φ auch eine Eliminationsmenge für $\exists x\Gamma_{\overline{L}(\mathbf{a}_i)}(\Gamma_N(\varphi))$. Somit gilt

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\Gamma_N(\varphi))[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Dann gilt analog zum ersten Fall $\mathbf{P} \models (\Gamma_{\overline{L}(\mathbf{a}_i)}(\Gamma_N(\varphi))[t/\!/x] \wedge \gamma)(\mathbf{z})$ für ein $(\gamma, t) \in E_i(\mathbf{z})$. Das liefert mit Bemerkung 3.2.13 schliesslich die Behauptung. □

Eine Möglichkeit der Anwendung von strukturellen Eliminationsmengen ist die Verkleinerung des Wertes m in Satz 3.1.12. Dies wird an einem einfachen Beispiel demonstriert.

Beispiel 3.2.20 Man betrachte die Elimination von $\exists x$ in $\exists x\varphi$ mit

$$\varphi = ((x > a \vee x \cong_{10} 0) \wedge x < b).$$

Wähle $M = \{\mathbf{a}_1, \mathbf{a}_2\}$ mit $\mathbf{a}_1 = (0, 0)$ und $\mathbf{a}_2 = (0, 1)$. Dann ist $\Gamma_M(\varphi) \sim$ false und somit $E^* = \emptyset$. Die Eliminationsmenge E_1 für

$$\exists x\Gamma_{\overline{L}(\mathbf{a}_1)}(\varphi) = \exists x((x > a \vee \text{false}) \wedge x < b)$$

ist zum Beispiel durch $E_1 = \{(\text{true}, a+k, ((-1 \leq k \leq 1, k))), (\text{true}, b+k, ((-1 \leq k \leq 1, k)))\}$ gegeben. Die Eliminationsmenge E_2 für

$$\exists x \Gamma_{\overline{L(\mathbf{a}_2)}}(\varphi) = \exists x((\text{false} \vee x \cong_{10} 0) \wedge x < b)$$

ist zum Beispiel durch $E_2 = \{(\text{true}, b+k, ((-10 \leq k \leq 10, k)))\}$ gegeben. Die Gesamteliminationsmenge kann man dann angeben Durch

$$E = E_1 \cup E_2.$$

Der Modulus der Kongruenz 10 taucht dabei im Gegensatz zur Anwendung des Satzes 3.1.12 in $(\text{true}, a+k, ((-1 \leq k \leq 1, k)))$ *nicht* auf.

Die Bildung von strukturellen Eliminationsmenge „entkoppelt" also bei geschickter Wahl der Mengen M die Bildung des kleinsten gemeinsamen Vielfachen der Moduli. Ein solches Verhalten ist nur mit die Bildung einer disjunktiven Normalform der Eingabeformel vergleichbar. Die Bildung einer DNF führt im Worst-Case zu einem exponentiellen Wachstum der Eingabeformellänge und ist somit nicht empfehlenswert. Die einmalige[4] Anwendung des Satzes 3.2.19 führt allerdings zu *keinem* exponentiellen Wachstum der Eingabeformellänge.

Das Beispiel 3.2.20 deutet eine geschickte Strategie die Menge M durch einen Algorithmus A_M rekursiv zu wählen an. Dieser ist durch folgende Schritte gegeben.

(i) Falls $\varphi^{\mathbf{a}}$ atomar ist, so wähle $A_M(\varphi) = \{\mathbf{a}\}$.

(ii) Falls φ von der Form $\varphi_1 \wedge \varphi_2$ ist, so ***wähle*** zwischen $A_M(\varphi_1)$ und $A_M(\varphi_2)$ die Menge mit der größten Anzahl der Elemente.

(iii) Falls φ von der Form $\varphi_1 \vee \varphi_2$ ist, so vereinige $A_M(\varphi_1)$ und $A_M(\varphi_2)$.

Die Markierung $\mathbf{a}$ in Schritt (i) ist die Markierung von der Teilformel φ in der Eingabeformel, mit der die Rekursion startet. Die oben skizzierte Strategie ergibt für die Formel aus Beispiel 3.4 unter der Annahme, daß ψ_1,...,ψ_4 und ψ atomare Formeln sind, die Auswahl $M(\varphi) = \{(1,0,1,0),(1,0,1,1),(1,1)\}$.

Wie man bereits aus Beispiel 3.2.20 erkennen kann, führt naives Bilden der Vereinigung $E = E^* \cup \bigcup_{i \in I} E_i$ zu einer Eliminationsmenge, die mehrere Tupeln der Form (γ_i, t_i, Ψ_i) mit bis auf die Bound-Variablen identischen Testpunkten und Bedingungen γ_i und t_i aber verschiedenen Boundtupeln Ψ_i enthält. Bei der Bildung der Ergebnisformel entstehen also viele Teilformeln, die das Einsetzen von identischen und somit redundanten Testpunkten repräsentieren. Es ist naheliegend alle Tupeln in E mit identischen Bedingungen und Pseudo-Termen zu jeweils einem Tupel zu „vereinigen". Für parametrische Eliminationsmengen nach Satz 3.1.12 läßt sich diese Idee algorithmisch einfach umsetzen. Im Folgenden wird der Unterschied der Testpunkte, der durch unterschiedlich gewählte Bound-Variablen $v'_1, ..., v'_n$ und v in verschiedenen Instanzen des Satzes 3.1.12 entsteht, vernachlässigt.

Lemma 3.2.21 (Konflation) Sei $E = E^* \cup \bigcup_{i \in I} E_i$ eine strukturelle Eliminationsmenge $\exists x \varphi$, sodaß jedes E_i für $i \in I$ eine Eliminationsmenge nach Satz 3.1.12 ist mit

$$E_i = \{(\gamma_{ik}, t_{ik}, ((\psi'_1, v'_1), ..., (\psi'_n, v'_n), (\gamma_{ik}, v))) \mid k \in I \cup J\} \cup \{(\text{true}, 0, \emptyset)\}.$$

Sei P eine Partition von $\bigcup_{i \in I} E_i$ in Teilmengen mit identischen Testtermen t_p und Bedingungen γ_p für $p \in P$. Definiere für ein $p \in P$

$$B_p = \{\gamma_{ik} \mid (\gamma_{ik}, t_{ik}, ((\psi'_1, v'_1), ..., (\psi'_n, v'_n), (\gamma_{ik}, v))) \in p\}.$$

[4]Es besteht die Möglichkeit bei der Bildung der Eliminationsmengen für $\exists x \Gamma_{\overline{L(\mathbf{a}_i)}}$ wieder den Satz 3.2.19 anzuwenden. Dies sollte allerdings nicht gemacht werden, da die rekursive Anwendung des Satzes 3.2.19 zur Bildung einer DNF der Eingabeformel ähnlich und in bestimmten Fällen sogar äquivalent ist.

Dann ist für

$$E_p = \left\{ \left(\gamma_p, t_p, ((\psi'_1, v'_1), ..., (\psi'_n, v'_n), (\bigvee_{\gamma \in B_p} \gamma, v)) \right) \right\}$$

die Menge $E' = E^* \cup \{\text{true}, 0, \emptyset\} \cup \bigcup_{p \in P} E_p$ eine Eliminationsmenge für $\exists x \varphi$. Die Menge E' wird als *Konflation* von E bezeichnet.

Beweis: Es genügt die Gleichheit $E(\mathbf{z}) = E'(\mathbf{z})$ für jedes $\mathbf{z} \in \mathbb{Z}^l$ zu zeigen. Sei $\mathbf{z} \in \mathbb{Z}^l$. Sei zunächst ein $(\gamma, t) \in E(\mathbf{z})$. Dann gilt aufgrund der Form von Eliminationsmengen nach Satz 3.1.12, daß ein $u \in S(\gamma_{ik}(v))$ gibt mit $t = t_{ik}[v/u]$ und $\gamma = \gamma_{ik}[v/u]$. Wähle $p \in P$ mit $(\gamma_{ik}, t_{ik}, \Psi_{ik}) \in p$. Dann gibt es ein $(\gamma_p, t_p, \Psi_p) \in E'$. Dann ist aber γ_{ik} eine Teilformel von

$$\bigvee_{\gamma \in B_p} \gamma.$$

Somit ist $u \in S((\bigvee_{\gamma \in B_p} \gamma)(v))$. Damit ist $(\gamma, t) \in E'(\mathbf{z})$. Die Umkehrung läuft analog. □

Beispiel 3.2.22 Betrachte die Elimination von $\exists x$ in $\exists x \varphi$ aus Beispiel 3.2.20 mit

$$\varphi = ((x > a \vee x \cong_{10} 0) \wedge x < b)$$

und $M = \{\mathbf{a}_1, \mathbf{a}_2\}$ für $\mathbf{a}_1 = (0, 0)$ und $\mathbf{a}_2 = (0, 1)$. Die Partition P enthält dann für den Pseudo-Term $t_p = b + k$ und für die Bedingung $\gamma_k = \text{true}$ eine zweielementige Menge

$$\{(\text{true}, b + k, ((-1 \leq k \leq 1, k))), (\text{true}, b + k, ((-10 \leq k \leq 10, k)))\}$$

Daher kann man die Gesamteliminationsmenge nach Lemma 3.2.21 aufgrund der Äquivalenz $(-1 \leq k \leq 1) \vee (-10 \leq k \leq 10) \sim -10 \leq k \leq 10$ angeben durch

$$E = \{(\text{true}, a + k, ((-1 \leq k \leq 1, k))), (\text{true}, b + k, ((-10 \leq k \leq 10, k)))\}.$$

3.2.3 Gauss-Elimination

Gauss-Elimination kann als eine Form der Bildung von strukturellen Eliminationsmengen angesehen werden, bei der die Endlichkeit von Erfüllungsmengen ausgezeichneter Teilformeln der Eingabeformel bezüglich der zu eliminierenden Variablen ausgenützt wird. Betrachtet man die Formel

$$\exists x(mx = b \wedge \varphi),$$

wobei $b \in \mathcal{T}$ mit $x \notin \mathcal{V}(b) = \{x_1, ..., x_l\}$ und $m \in \mathbb{Z} \backslash \{0\}$, so erkennt man, daß $E = \{(b \cong_m 0, \frac{b}{m}, \emptyset)\}$ offensichtlich eine Eliminationsmenge für $\exists x(mx = b \wedge \varphi)$ ist. Es gilt somit

$$\exists x(mx = b \wedge \varphi) \sim \varphi \left[\frac{b}{m} /\!\!/ x \right] \wedge b \cong_m 0.$$

Dabei ist es für die Elimination von x *nicht* notwendig atomare Formeln in φ zu betrachten. Obiger Beobachtung liegt zugrunde, daß die Erfüllungsmenge $S = S^{\mathbf{x}}_{\mathbf{z}}((mx = b \wedge \varphi)(\mathbf{x}, x))$ für $\mathbf{x} = (x_1, ..., x_l)$ für jede Wahl von $\mathbf{z}$ *endlich* ist und jedes Element $z \in S$ durch einen geeigneten Pseudo-Term, nämlich durch $\frac{b}{m}$, dargestellt werden kann, sodaß für $b(x_1, ..., x_l)$ gilt $z = \frac{b^{\mathbf{P}}(\mathbf{z})}{m}$.

Definition 3.2.23 (Gauss-Formel) Eine schwach quantorenfreie Formel φ heißt eine *Gauss-Formel* bezüglich x, falls für $\varphi(\mathbf{x}, x)$ mit $\mathbf{x} = (x_1, ..., x_l)$ für jede Wahl von $\mathbf{z} \in \mathbb{Z}^l$ folgende Bedingungen erfüllt sind.

(i) $|S^{\mathbf{x}}_{\mathbf{z}}(\varphi(\mathbf{x}, x))| < \infty$.

(ii) Es gibt eine endliche Menge von Pseudo-Termen

$$T = \{\frac{b_i}{n_i} \mid i \in I\}$$

sodaß für jedes $z \in S^{\mathbf{x}}_{\mathbf{z}}(\varphi(\mathbf{x}, x))$ ein Term $\frac{b_i}{n_i} \in T$ existiert, sodaß für die erweiterten Terme $b_i(\mathbf{x})$ und $n_i(\mathbf{x})$ gilt

$$z = \frac{b_i^{\mathbf{P}}(\mathbf{z})}{n_i^{\mathbf{P}}(\mathbf{z})}.$$

Die Menge T heißt die Menge der Pseudo-Terme von ψ bezüglich x. Falls x aus dem Kontext bekannt ist, so wird der Verweis darauf weggelassen.

Das folgende Lemma 3.2.24 gibt hinreichende Bedingungen dafür, daß eine Formel eine Gauss-Formel ist. Diese sind im Allgemeinen *nicht* notwendig.

Lemma 3.2.24 (Hinreichende Bedingungen für Gauss-Formeln) Sei φ schwach quantorenfrei und in Eliminationsnormalform bezüglich x.

(i) Falls φ eine atomare Gleichung $ax = b$ mit $a \in \mathbb{Z} \setminus \{0\}$ ist, so ist φ eine Gauss-Formel bezüglich x.

(ii) Falls φ von der Form $\varphi_1 \wedge \varphi_2$ ist, sodaß φ_1 eine Gauss-Formel bezüglich x ist, so ist auch φ eine Gauss-Formel bezüglich x.

(iii) Falls φ von der Form $\varphi_1 \vee \varphi_2$ ist, sodaß φ_1 und φ_2 Gauss-Formeln bezüglich x sind, so ist auch φ eine Gauss-Formel bezüglich x.

(iv) Falls φ von der Form $\bigvee_{\psi(v)} \gamma$ bzw. $\bigwedge_{\psi(v)} \gamma$ ist, sodaß γ eine Gauss-Formel bezüglich x ist, so ist auch φ eine Gauss-Formel bezüglich x.

Beweis: Gelte o.B.d.A. $\mathcal{V}_f(\varphi) = \{x_1, ..., x_l, x\}$ und sei $\mathbf{x} = (x_1, ..., x_l)$. (i) Offenbar ist für ein $\mathbf{z} \in \mathbb{Z}^l$ die Erfüllungsmenge $S^{\mathbf{x}}_{\mathbf{z}}(\psi(\mathbf{x}, x))$ einelementig, falls $a \mid b^{\mathbf{P}}(\mathbf{z})$ und sonst leer. Ein geeigneter Pseudo-Term ist $\frac{b}{a}$. (ii) Für die Erfüllungsmenge von $\psi_1 \wedge \psi_2$ bezüglich x gilt für jede Wahl von $\mathbf{z} \in \mathbb{Z}^l$

$$S^{\mathbf{x}}_{\mathbf{z}}((\psi_1 \wedge \psi_2)(\mathbf{x}, x)) = S^{\mathbf{x}}_{\mathbf{z}}(\psi_1(\mathbf{x}, x)) \cap S^{\mathbf{x}}_{\mathbf{z}}(\psi_2(\mathbf{x}, x)) \subseteq S^{\mathbf{x}}_{\mathbf{z}}(\psi_1(\mathbf{x}, x)).$$

Somit ist $S^{\mathbf{x}}_{\mathbf{z}}((\psi_1 \wedge \psi_2)(\mathbf{x}, x))$ ebenfalls endlich. Sei T_1 die Menge der Pseudo-Terme für ψ_1. Dann ist T_1 offenbar eine geeignete Menge von Pseudo-Termen für $\psi_1 \wedge \psi_2$. (iii) Für die Erfüllungsmenge von $\psi_1 \vee \psi_2$ bezüglich x gilt für jede Wahl von $\mathbf{z} \in \mathbb{Z}^l$

$$S^{\mathbf{x}}_{\mathbf{z}}((\psi_1 \vee \psi_2)(\mathbf{x}, x)) = S^{\mathbf{x}}_{\mathbf{z}}(\psi_1(\mathbf{x}, x)) \cup S^{\mathbf{x}}_{\mathbf{z}}(\psi_2(\mathbf{x}, x)).$$

Somit ist $S^{\mathbf{x}}_{\mathbf{z}}((\psi_1 \vee \psi_2)(\mathbf{x}, x))$ ebenfalls endlich. Seien T_1, T_2 die Mengen der Pseudo-Terme für ψ_1 und ψ_2. Dann ist $T_1 \cup T_2$ offenbar eine geeignete Menge von Pseudo-Termen für $\psi_1 \vee \psi_2$. (iv) Für jedes $\mathbf{z} \in \mathbb{Z}^l$ ist die Expansion der Formel $\varphi[z_1/x_1, ..., z_l/x_l]$ eine endliche Disjunktion bzw. Konjunktion von Gauss-Formeln und somit nach wiederholter Anwendung von (ii) bzw. (iii) ebenfalls eine Gauss-Formel. Ist $T = \{\frac{b_i}{n_i} \mid i \in I\}$ die Menge der Pseudo-Terme für γ, so kann man die Menge von Pseudo-Termen für φ für $\mathbf{z} \in \mathbb{Z}^l$ angeben durch

$$T' = \left\{ \frac{b_i[u/v]}{n_i[u/v]} \mid i \in I, u \in S^{\mathbf{x}}_{\mathbf{z}}(\psi(\mathbf{x}, v)) \right\}. \square$$

Als nächstes wird gezeigt, wie man parametrische Eliminationsmengen für Gauss-Formeln analog zum Satz 3.1.12 angeben kann. Es wird daran erinnert, daß unter einer Formel φ in $\exists x \varphi$ mit ausschließlich existentiellen gebundenen Quantoren eine Formel der Form

$$\exists x \bigvee_{\psi_1(v_1)} ... \bigvee_{\psi_n(v_n)} \gamma$$

verstanden wird.

Lemma 3.2.25 (Eliminationsmengen für Gauss-Formeln) Sei φ in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei γ die stark quantorenfreie Matrix von φ. Sei $T = \{\frac{b_i}{c_i} \mid i \in I\}$ die Menge der Pseudo-Terme für γ. Seien $v'_1, ..., v'_n$ neue Variablen. Sei weiter $\psi'_i = \psi_i[v'_1/v_1, ..., v'_n/v_n]$ für $1 \leq i \leq n$ und $b'_i = b_i[v'_1/v_1, ..., v'_n/v_n]$ für $i \in I$. Dann ist die Menge

$$E = \{(c_i \neq 0 \wedge b'_i \cong_{c_i} 0, \frac{b'_i}{c_i}, ((\psi'_1, v'_1), ..., (\psi'_n, v'_n))) \mid i \in I\}$$

eine Eliminationsmenge für $\exists x\varphi$.

Beweis: Sei $\mathbf{z} \in \mathbb{Z}^l$ eine erfüllende Stelle von $(\exists x\varphi)(x_1, ..., x_l)$. Gelte etwa $\mathbf{P} \models \varphi(\mathbf{z}, z)$ für $z \in \mathbb{Z}$. Dann gibt es für die erweiterte Formel $\gamma(x_1, ..., x_l, v_1, ..., v_n, x)$ ein $\mathbf{u} = (u_1, ..., u_n) \in \mathbb{Z}^n$ mit $\mathbf{P} \models \gamma(\mathbf{z}, \mathbf{u}, z)$. Da γ eine Gauss-Formel bezüglich x ist, gibt es in T einen Pseudo-Term $t = \frac{b_i}{c_i}$ mit

$$z = \frac{(b_i)^{\mathbf{P}}(\mathbf{z}, \mathbf{u})}{c_i^{\mathbf{P}}(\mathbf{z})}.$$

Dabei gilt $u_i \in S(\psi_i(v_i))$ für $1 \leq i \leq n$. Offensichtlich gilt dann auch $u_i \in S(\psi'_i(v'_i))$. Somit gilt auch $c_i^{\mathbf{P}}(\mathbf{z}) \mid b_i^{\mathbf{P}}(\mathbf{z}, \mathbf{u})$ und es gibt ein $(\gamma, t) \in E(\mathbf{z})$ mit $t = \frac{b'_i[u_1/v'_1, ..., u_1/v'_n]}{c_i}$, sodaß

$$z = \frac{(b'_i[u_1/v'_1, ..., u_1/v'_n])^{\mathbf{P}}(\mathbf{z})}{c_i^{\mathbf{P}}(\mathbf{z})}.$$

Somit gilt $\mathbf{P} \models (\varphi[t/\!/x](\mathbf{z}) \wedge \gamma)(\mathbf{z})$. □

Es ist an dieser Stelle sinnvoll zu bemerken, daß die Terme c_i der Pseudo-Terme in Lemma 3.2.25 nach der hier eingeführten Konvention *keine* Bound-Variablen enthalten. Gegenteiliges würde einen multiplikativen Zusammengang zwischen Bound-Variablen und x beschreiben.

Lemma 3.2.26 Sei φ eine Formel in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei weiter $\psi^{\mathbf{a}}$ eine markierte Gauss-Formel bezüglich x. Sei E die Eliminationsmenge für

$$\bigvee_{\psi_1(v_1)} ... \bigvee_{\psi_n(v_n)} \psi.$$

Dann ist die Formel $\Gamma_{\overline{L(\mathbf{a})}}(\varphi)$ eine Gauss-Formel mit der *Gauss-Eliminationsmenge* E.

Beweis: Der Beweis wird nach Induktion nach der Länge k des Tupels $\mathbf{a} = (a_1, ..., a_k)$ geführt. Für $k \leq n$ ist die Aussage wegen $\Gamma_{\overline{L(\mathbf{a})}}(\varphi) = \varphi$ trivial. Sei die Aussage für $k \in \mathbb{N}$ mit $k > n$ gezeigt. Betrachte nun die Formel $\Gamma_{\overline{L(\mathbf{a})}}$ für $k+1$ mit $\mathbf{a} = (a_1, ..., a_{k+1})$. Dann ist die Formel $\gamma = \beta(a_1, ..., a_k)$ von der Form $\gamma_1 \, \rho \, \gamma_2$ mit $\rho \in \{\wedge, \vee\}$, wobei o.B.d.A. $\psi = \gamma_1$. Falls γ eine Disjunktion ist, so folgt nach Lemma 3.2.6(ii), daß $\gamma_2 = \text{false}$. Falls γ eine Konjunktion ist, und somit auch für jede Wahl von $\rho \in \{\wedge, \vee\}$ ist γ nach Lemma 3.2.24 eine Gauss-Formel mit der Eliminationsmenge E. Nach Induktionsannahme folgt, daß $\Gamma_{\overline{L((a_1,...,a_k))}}$ und somit auch $\Gamma_{\overline{L((a_1,...,a_{k+1}))}}$ eine Gauss-Formel mit Eliminationsmenge E ist. □

Ein mögliches Ziel bei der Gauss-Elimination besteht darin, möglichst *tiefe* Gauss-Formeln mit möglichst *kleinen* und *einfachen* Eliminationsmengen zu finden. Der folgende Satz zeigt, wie man das Konzept der Gauss-Eliminationsmengen in den regulären Eliminationsvorgang einbeziehen kann.

Satz 3.2.27 (Gauss-Elimination) *Sei φ eine Formel in $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei weiter $\{\psi_i^{\mathbf{a}_i} \mid i \in I\}$ eine Menge von markierten Gauss-Formeln von φ und $G = \{\mathbf{a}_i \mid i \in I\}$ die Menge entsprechender Markierungen. Sei E_i eine Gauss-Eliminationsmenge für ψ_i in $\exists x\varphi$ mit $i \in I$ bezüglich x. Sei weiterhin E^* die Eliminationsmenge für $\exists x\Gamma_G(\varphi)$. Dann ist*

$$E = E^* \cup \bigcup_{i \in I} E_i$$

eine Eliminationsmenge für $\exists x\varphi$.

Beweis: Nach Satz 3.2.19 setzt sich die Eliminationsmenge von $\exists x\varphi$ aus den Eliminationsmengen für $\exists x\Gamma_G(\varphi)$ und der Vereinigung von Eliminationsmengen für $\exists x\Gamma_{\overline{L(\mathbf{a}_i)}}(\varphi)$ für $i \in I$. Nun ist aber die Formel

$$\Gamma_{\overline{L(\mathbf{a}_i)}}(\varphi)$$

nach Lemma 3.2.26 eine Gauss-Formel mit der Eliminationsmenge E_i. Das liefert unmittelbar die Behauptung. □

Nun kann man ein weiteres Beispiel für die Anwendung des Satzes 3.2.19 bzw. 3.2.27 angeben.

Beispiel 3.2.28 Betrachte die Elimination von $\exists x$ in $\exists x\varphi$ mit

$$\varphi = \bigvee_{(0\leq v\leq 2)(v)} ((x = b \vee x > v) \wedge v < a).$$

Die Formel $x = b$ ist eine Gauss-Formel in φ bezüglich x. Eine Eliminationsmenge für

$$\exists x \bigvee_{(0\leq v\leq 2)(v)} ((x = b \vee \text{false}) \wedge v < a)$$

ist durch $E_1 = (\text{true}, b, \emptyset)$ gegeben. Eine Eliminationsmenge für

$$\exists x \bigvee_{(0\leq v\leq 2)(v)} ((\text{false} \vee x > v) \wedge v < a)$$

ist durch $E^* = \{(\text{true}, k, ((-1 \leq k \leq 3, k)))\}$ gegeben. Somit ist

$$E = E_1 \cup E_2$$

ist eine Eliminationsmenge von $\exists x\varphi$. Das liefert nach Simplifikation die Ergebnisformel

$$\varphi' = \bigvee_{(0\leq v\leq 2)(v)} (b > v \wedge v < a) \vee \bigvee_{(-1\leq k\leq 3)(v)} \bigvee_{(0\leq v\leq 2)(v)} ((k = b \vee k > v) \wedge v < a).$$

Eine bemerkenswerte Folgerung aus Satz 3.2.27 und Lemma 3.2.26 ist, daß die Elimination von tief in der Eingabeformel liegenden Gauss-Formeln auf die Elimination einer Gauss-Formel reduziert werden kann. Es wäre denkbar die Elimination von schwach quantorenfreien Formeln mit ausschließlich existentiellen gebundenen Quantoren auf die Elimination entsprechender stark quantorenfreier Formeln unter der Anwendung der Regel

$$\exists x \bigvee_{\psi(v)} \varphi \sim \bigvee_{\psi(v)} \exists x\varphi$$

zurückführen. Dies ist allerdings nicht empfehlenswert, da dadurch die Information, daß die Bound-Variablen der gebundenen Quantoren durch endliche Wertemengen beschränkt sind, der Elimination von $\exists x\varphi$ nicht verfügbar ist.

3.2.4 Condensing

Bei der Bildung von strukturellen Eliminationsmengen betrachtet man Formeln, die durch die Anwendung des Condensing-Operators auf ausgewählte Mengen von Teilformeln der Eingabeformel, also durch die Ersetzung der Teilformeln in diesen Mengen durch false, entstehen. Die Eliminationsmenge für $\exists x\varphi$ wird dabei aus Eliminationsmengen für $\exists x\Gamma_{M_i}(\varphi)$ für ausgezeichnete Mengen M_i gebildet. Dabei trifft man intuitiv die Annahme, daß alle durch false ersetzten Teilformeln in $\Gamma_{M_i}(\varphi)$ für die Erfüllbarkeit von φ an einer festen Stelle $\mathbf{z}$ „nicht notwendig" sind und somit

als nicht erfüllt vorausgesetzt werden können. Es ist naheliegend Gebrauch von dieser Ersetzung für den Substitutionsvorgang zu machen. Diese Beobachtung läßt sich offenbar für jede Instanz des Satzes 3.2.19 umsetzen, also auch zum Beispiel bei der Anwendung der Gauss-Elimination. In diesem Abschnitt wird gezeigt, wie die Annahme, daß bestimmte Teilformeln der Eingabeformel φ für die Erfüllbarkeit von $\varphi(x_1, ..., x_n, x)$ an einer gegebenen Stelle nicht notwendig sind bei der Substitution von Testpunkten umgesetzt werden kann.

Satz 3.2.29 (Strukturelles Condensing) *Sei φ eine Formel $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei $\{\psi_i^{\mathbf{a}_i} \mid i \in I\}$ eine Menge von markierten Teilformeln von φ und sei $M = \{\mathbf{a}_i \mid i \in I\}$ die Menge entsprechender Markierungen. Bezeichne E^* die Eliminationsmenge für die Formel $\exists x \Gamma_M(\varphi)$ und E_i die Eliminationsmenge für die Formel $\exists x \Gamma_{\overline{L}(\mathbf{a}_i)}$. Sei weiter $E = E^* \cup \bigcup_{i\in I} E_i$. Für die Erweiterung $(x_1, ..., x_l)$ und jede Stelle $\mathbf{z} \in \mathbb{Z}^l$ gilt $\mathbf{P} \models (\exists x\varphi)(\mathbf{z})$ genau dann, wenn gilt*

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \vee \bigvee_{i\in I} \bigvee_{(\gamma,t)\in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Beweis: Die Richtung von Rechts nach Links ergibt sich aus Bemerkung 3.2.13. Die Menge E ist nach Satz 3.2.19 eine Eliminationsmenge für $\exists x\varphi$. Gelte $(\exists x\varphi)^{\mathbf{P}}(\mathbf{z}) = \top$, was gleichbedeutend ist mit $\mathbf{P} \models (\exists x\varphi)(\mathbf{z})$. Sei etwa $z \in \mathbb{Z}$ mit $\mathbf{P} \models \varphi(\mathbf{z}, z)$. Falls analog zum Beweis des Satzes 3.2.19 $\mathbf{P} \models (\Gamma_M(\varphi))(\mathbf{z}, z)$ gilt, so gilt

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Falls hingegen $\mathbf{P} \not\models (\Gamma_M(\varphi))(\mathbf{z})$, so gibt es eine Teilmenge von markierten Formeln $N \subseteq M$ und eine markierte Teilformel $\psi_i^{\mathbf{a}_i}$, sodaß gilt

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\Gamma_N(\varphi))[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Dann gilt $\mathbf{P} \models (\Gamma_{\overline{L}(\mathbf{a}_i)}(\Gamma_N(\varphi))[t/\!/x] \wedge \gamma)(\mathbf{z})$ für ein $(t, \gamma) \in E_i(\mathbf{z})$. Das liefert mit Bemerkung 3.2.13 schliesslich

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Dies liefert unmittelbar die Behauptung. □

Satz 3.2.29 besagt, daß man für jede Stelle $\mathbf{z} \in \mathbb{Z}^l$ bei der Substitution von Testpunkten aus der Menge $E_i(\mathbf{z})$ bzw. $E^*(\mathbf{z})$ anstatt der Eingabeformel φ die Formel $\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)$ bzw. $\Gamma_M(\varphi)$ verwenden kann. Da dies für jede Stelle $\mathbf{z} \in \mathbb{Z}^l$ gilt, kann dieser Sachverhalt auch *syntaktisch* in der Ausgabeformel kodiert werden. Strukturelles Condensing kann also stets dann angewandt werden, wenn die Eliminationsmenge nach Satz 3.2.19 eine strukturelle Eliminationsmenge ist.

Beispiel 3.2.30 Betrachte die Elimination von $\exists x$ in $\exists x\varphi$ aus Beispiel 3.2.20 mit

$$\varphi = ((x > a \vee x \cong_{10} 0) \wedge x < b).$$

Wie in Beispiel 3.2.20 gilt mit $M = \{\mathbf{a}_1, \mathbf{a}_2\}$ mit $\mathbf{a}_1 = (0, 0)$ und $\mathbf{a}_2 = (0, 1)$ zum Beispiel $E^* = \emptyset$, $E_1 = \{(\text{true}, a + k, ((-1 \leq k \leq 1, k))), (\text{true}, b + k, ((-1 \leq k \leq 1, k)))\}$ und $E_2 =$

$\{(\text{true}, b+k, ((-10 \leq k \leq 10, k)))\}$. Nun ergibt sich nach Satz 3.2.29 für die schwach quantorenfreie Ergebnisformel φ'

$$\varphi' = \bigvee_{\psi_1(k)} (k > 0 \wedge a + k < b) \vee \bigvee_{\psi_2(k)} (b + k > a \wedge k < 0) \vee \bigvee_{\psi_3(k)} (b + k \cong_{10} 0 \wedge k < 0).$$

Dabei sind die Bounds ψ_1, ψ_2 und ψ_3 durch die entsprechenden Tupel gegeben. Es gilt $\psi_1 = -1 \leq k \leq 1$, $\psi_2 = -1 \leq k \leq 1$ und $\psi_3 = -10 \leq k \leq 10$. Um die Leistungsfähigkeit des Simplifikation zu demonstrieren wird nochmal nachträglich die Bound-Verstärkung aus Lemma 2.2.14 angewandt mit dem Ergebnis

$$\varphi'' = \bigvee_{(k=1)(k)} (a + k < b) \vee \bigvee_{(k=-1)(k)} (b + k > a) \vee \bigvee_{(-10 \leq k < 0)(k)} (b + k \cong_{10} 0).$$

Durch das Anwenden des Lemmas 2.2.13(v) erhält man schließlich das Ergebnis

$$\exists x \varphi \sim a + 1 < b \vee \bigvee_{(-10 \leq k < 0)(k)} (b + k \cong_{10} 0).$$

Dieses enthält im Vergleich zum Ergebnis der regulären Elimination nach Satz 3.1.12

$$\exists x \varphi \sim \bigvee_{(-10 \leq k < 0)(k)} (a - b < k \vee b + k \cong_{10} 0) \vee \bigvee_{(-10 \leq k \leq 10)(k)} ((a + k \cong_{10} 0 \vee k > 0) \wedge a + k < b)$$

nur 4 (statt 9) atomare Formeln und führt zur Substitution von nur 9 (statt 29) Testpunkten.

Man erkennt, daß im Beispiel 3.2.30 die Eliminationsmenge mit der einer disjunktiven Normalform der Eingabeformel übereinstimmt. In der Tat stellt die Bildung von strukturellen Eliminationsmengen intuitiv eine Zwischenstufe zwischen der Elimination der Eingabeformel und ihrer DNF dar. Man erkennt weiter, daß obwohl die Tupeln $(\text{true}, b + k, ((-1 \leq k \leq 1, k)))$ und $(\text{true}, b + k, ((-10 \leq k \leq 10, k)))$ bis auf die Bounds übereinstimmen und durch Konflation nach Satz 3.2.21 verschmolzen werden können, die entsprechenden Testpunkte nach Satz 3.2.29 in *unterschiedliche* Formeln $x > a \wedge x < b$ bzw. $x \cong_{10} 0 \wedge x < b$ eingesetzt werden. Bemerkenswert ist, daß Konflation und Condensing trotzdem auf eine natürliche Weise gleichzeitig angewandt werden können.

Lemma 3.2.31 (Konflation und Condensing) Sei E' eine Konflation einer strukturellen Eliminationsmenge $E = E^* \cup \bigcup_{i \in I} E_i$ nach Satz 3.2.19 bzw. 3.2.21. Sei für die in Satz 3.2.21 definierte Partition P die Markierung einer gemeinsamen Oberformel $\mathbf{a}_p$ von allen $\psi_i^{\mathbf{a}_i}$ mit $\gamma_{ik} \in B_p$ für jedes $p \in P$. Für die Erweiterung $(x_1, ..., x_l)$ und jede Stelle $\mathbf{z} \in \mathbb{Z}^l$ gilt $\mathbf{P} \models (\exists x \varphi)(\mathbf{z})$ genau dann, wenn gilt

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t) \in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \vee \bigvee_{p \in P} \bigvee_{(\gamma,t) \in E_p(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_p)}(\varphi)[t_p/\!/x] \wedge \gamma_p) \right)(\mathbf{z}).$$

Beweis: Die Richtung von Rechts nach Links ergibt sich aus Bemerkung 3.2.13. Gelte nun $\mathbf{P} \models (\exists x \varphi)(\mathbf{z})$. Nach Satz 3.2.29 gilt dann

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t) \in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \vee \bigvee_{i \in I} \bigvee_{(\gamma,t) \in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Falls gilt

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t) \in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}),$$

so ist die Behauptung trivial. Gelte anderenfalls

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Dann gilt $\mathbf{P} \models (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t/\!/x]\wedge\gamma)(\mathbf{z})$ für ein $(\gamma,t) \in E_i(\mathbf{z})$. Es gibt also ein $U = \{(\gamma_{ik}, t_{ik}, \Psi_{ik})\} \subseteq E_i$ mit $(\gamma,t) \in U(\mathbf{z})$. Dann gibt es ein $p \in P$ mit der Eigenschaft $\gamma_p = \gamma_{ik}$ und $t_p = t_{ik}$ und $E_p = \{(\gamma_p, t_p, \Psi_p)\}$. Nach Definition von E_p gilt offensichtlich dann

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E_p(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t_p/\!/x] \wedge \gamma_p) \right)(\mathbf{z}).$$

Das liefert mit Bemerkung 3.2.13 schliesslich die Behauptung, da die Formel mit Markierung $\mathbf{a}_p$ eine Oberformel von $\mathbf{a}_i$ ist und somit $\overline{L}(\mathbf{a}_p) \subseteq \overline{L}(\mathbf{a}_i)$. □

Einen weiteren Fall der Anwendung des Satzes 3.2.29 stellt Condensing in Verbindung mit der Gauss-Elimination dar. Bei der Bildung von Eliminationsmengen für Gauss-Elimination in Satz 3.2.27 wurden für die Bestimmung der „Resteliminationsmenge" E^* mit Hilfe des Condensing-Operators alle entdeckten Gauss-Formeln der Ausgangsformel durch false ersetzt. Diese Teilformeln können bei der Substitution von Testpunkten aus E^* offensichtlich ebenfalls durch false ersetzt werden. Dies stellt eine *triviale* und sehr leicht programmiertechnisch realisierbare Anwendung von Gauss-Condensing.

Satz 3.2.32 *Sei φ eine Formel $\exists x\varphi$ mit ausschließlich existentiellen gebundenen Quantoren. Sei $\{\psi_i^{\mathbf{a}_i} \mid i \in I\}$ eine Menge von markierten Gauss-Formeln von φ und sei $M = \{\mathbf{a}_i \mid i \in I\}$ die Menge entsprechender Markierungen. Bezeichne E^* die Eliminationsmenge für die Formel $\exists x\Gamma_M(\varphi)$ und E_i die Eliminationsmenge für die Formel $\exists x\Gamma_{\overline{L}(\mathbf{a}_i)}$. Sei weiter $E = E^* \cup \bigcup_{i\in I} E_i$. Für die Erweiterung $(x_1, ..., x_l)$ und jede Stelle $\mathbf{z} \in \mathbb{Z}^l$ gilt $\mathbf{P} \models (\exists x\varphi)(\mathbf{z})$ genau dann, wenn gilt*

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t)\in E^*(\mathbf{z})} (\Gamma_M(\varphi)[t/\!/x] \wedge \gamma) \vee \bigvee_{i\in I} \bigvee_{(\gamma,t)\in E_i(\mathbf{z})} \varphi[t/\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Beweis: Der Behauptung folgt unmittelbar aus Satz 3.2.29. □

Man kann nun vermuten, daß eine Verallgemeinerung obiger Technik darin bestehen könnte, zusätzlich zur Anwendung des Satzes 3.2.32 bei der Substitution eines Testpunktes aus der Eliminationsmenge E_i einer Gauss-Formel ψ_i alle anderen Gauss-Formeln ψ_j für $i \neq j \in I$ durch false zu ersetzen. Diese Vorgehensweise erweist sich im Allgemeinen als falsch, wie das folgende Beispiel zeigt.

Beispiel 3.2.33 Betrachte die Elimination von $\exists x\varphi$ mit

$$\varphi = (x < -10 \vee (x = 1 \wedge x = a)) \wedge ((x = 1 \wedge x = b) \vee x > 10).$$

Es ist leicht einzusehen, daß $\psi_1 = (x = 1\wedge x = a)$ und $\psi = (x = 1\wedge x = b)$ zwei Gauss-Formeln von φ bezüglich x sind. Die folgende Beziehung ist leicht durch Fallunterscheidungen zu überprüfen:

$$\exists x\varphi \sim a = 1 \wedge b = 1.$$

Die Formel $\Gamma_{\{(0,1),(1,0)\}}(\varphi)$ ist äquivalent zu false. Daher ist eine geeignete Eliminationsmenge für $\exists x\varphi$ nach Satz 3.2.27 nur aus Eliminationsmengen der Gauss-Formeln ψ_1 und ψ_2 definiert zum Beispiel durch

$$E = E_1 \cup E_2 = \{(\text{true}, 1, \emptyset), (\text{true}, b, \emptyset)\},$$

wobei $E_1 = \{(\text{true}, 1, \emptyset\}$ und $E_2 = \{(\text{true}, b, \emptyset)\}$. Man beachte, daß diese Auswahl nicht die kleinste bezüglich der Kardinalität der Gesamteliminationsmenge ist. Ersetzt man allerdings beim Substituieren vom Testpunkt aus E_1 die Teilformel ψ_2 durch false und umgekehrt, so erhält man nach Vereinfachen

$$((1 < -10 \vee a = 1) \wedge 1 > 10)) \vee (b < -10 \wedge (b = 1 \vee b > 10)) \sim \text{false}.$$

Letzteres Ergebnis ist offensichtlich nicht korrekt.

Obiger Ansatz versagt offenbar dann, wenn für eine bestimmte Wahl von Parametern für die Erfüllung der Ausgangsformel die Erfüllung von zwei oder mehreren Gauss-Formeln notwendig ist. Durch Ersetzen einer von denen durch false ist speziell für die genannte Wahl von Parametern die Ergebnisformel äquivalent zu false. Das Gegenbeispiel 3.2.33 war ursprünglich die Motivation für die Einführung des Begriffs „konjunktive Assoziiertheit".

Satz 3.2.34 (Gauss-Condensing) *Sei φ eine schwach quantorenfreie Formel in $\exists x \varphi$. Sei $\{\psi_i^{\mathbf{a}_i} \mid i \in I\}$ eine Menge von markierten Gauss-Formeln von φ und sei $M = \{\mathbf{a}_i \mid i \in I\}$ die Menge entsprechender Markierungen. Bezeichne E^* die Eliminationsmenge für die Formel $\exists x \Gamma_M(\varphi)$ und E_i die Eliminationsmenge für die Formel $\exists x \Gamma_{\overline{L}(\mathbf{a}_i)}$. Sei weiter $E = E^* \cup \bigcup_{i \in I} E_i$. Für die Erweiterung $(x_1, ..., x_l)$ und jede Stelle $\mathbf{z} \in \mathbb{Z}^l$ gilt $\mathbf{P} \models (\exists x \varphi)(\mathbf{z})$ genau dann, wenn gilt*

$$\mathbf{P} \models \left(\bigvee_{(\gamma,t) \in E^*(\mathbf{z})} (\Gamma_G(\varphi)[t/\!\!/x] \wedge \gamma) \vee \bigvee_{i \in I} \bigvee_{(\gamma,t) \in E_i(\mathbf{z})} (\Gamma_{\overline{L}(\mathbf{a}_i)}(\varphi)[t/\!\!/x] \wedge \gamma) \right)(\mathbf{z}).$$

Beweis: Die Behauptung folgt direkt aus Satz 3.2.29. □

3.3 Zusammenfassung

In diesem Kapitel ist ein Quantoreneliminationsverfahren für die Menge der linear quantifizierten Formeln der uniformen Presburger-Arithmetik vorgestellt worden. Ferner sind einige Techniken zur Verbesserung der praktischen Anwendbarkeit dieses Verfahrens vorgestellt worden.

Das Quantoreneliminationsverfahren wurde als Elimination durch virtuelle Substitution von Testpunkten dargestellt. Es wurde eine geeignete Substitutionsvorschrift für die uniforme Presburger Arithmetik angegeben, die ohne Fallunterscheidungen bezüglich des Nennerterms auskommt. Mit Hilfe dieser Vorschrift wurden parametrische Eliminationsmengen für linear quantifizierte Formeln der Presburger-Arithmetik angegeben.

Die Bildung von strukturellen Eliminationsmengen erweist sich als uniformes Konzept und kann im Wesentlichen unverändert auf ein *beliebiges* Eliminationsverfahren durch virtuelle Substitution angewandt werden. Es hat sich gezeigt, daß durch die Einführung von konjunktiver Assoziiertheit die in [Dol00] beschriebene partielle Gauss-Elimination auf die Elimination einer existenziell quantifizierten Gauss-Formel zurückgeführt werden kann. Das Konzept der konjunktiven Assoziiertheit scheint die Bildung von strukturellen Eliminationsmengen stark auszuloten. In wie weit das Konzept der strukturellen Eliminationsmengen verallgemeinert bzw. verbessert werden kann ist eine interessante offene Frage.

Condensing bildet bezüglich der Länge der Ausgabeformel einen günstigeren Ersatz für die virtuelle Substitution und ist ebenfalls uniform für jedes Eliminationsverfahren durch virtuelle Substitution anwendbar. Zu jeder Form von strukturellen Eliminationsmengen kann eine Condensing-Art angegeben werden. Ein wichtiges Beispiel dafür ist Gauss-Condensing. Es ist naheliegend auch andere Teilmengen von markierten Formeln für strukturelle Eliminationsmengen und Condensing zu verwenden. Dabei kann man sich nicht nur durch die Suche nach strukturellen Eliminationsmengen und anschließende Einführung einer Condensing-Art leiten lassen. Auch der umgekehrte

Weg ist denkbar. Unabhängig von der Bildung von strukturellen Eliminationsmengen kann man Condensing auch für andere Eliminationsmengen anwenden. So ist auch eine direkte Anwendung des Hauptsatzes über Condensing auf atomare Formeln einer Formel denkbar. Diese Vorgehensweise liefert eine Technik, welche in [Dol00] beschrieben wurde und dort als „Positional-Condensing" bezeichnet wird. Alle hier angegebenen Ansätze sprengen den Rahmen dieser Arbeit und bleiben somit offen für weitere Forschung.

Sowohl die Bildung von strukturellen Eliminationsmengen als auch Condensing können effizient implementiert werden. Die Implementierung muß dabei im Wesentlichen nur über die Positionen von ausgewählten markierten Teilformeln, welche durch die Markierungen explizit gegeben sind, verfügen. Bei beiden Konzepten reicht es für die Realisierung dieser auf dem Weg zur gegebenen Teilformel nicht konjunktiv assoziierte Äste auszulassen bzw. durch false zu ersetzen.

Kapitel 4

Implementierung und Testergebnisse

Alle in dieser Arbeit beschriebenen Ansätze wurden im Computeralgebra-System REDUCE als Teil des Paketes REDLOG implementiert. Mit Hilfe der Implementierung wurde das in dieser Arbeit beschriebene Quantoreneliminationsverfahren an einer Auswahl von Problemstellungen erprobt. Dabei stand die Auswertung der Verbesserungen durch eingeführte Techniken im Vordergrund. In diesem Abschnitt werden die Ergebnisse der durchgeführten Tests und die Implementierung beschrieben.

4.1 Computeralgebra und Logik System REDLOG

Computeralgebra- und Logik-System REDLOG ist in das weit verbreitete Softwaresystem für Symbolmanipulation und algebraische Berechnungen REDUCE eingebettet. REDUCE wurde von Anthony C. Hearn von der RAND Korporation ursprünglich zur Lösung von Problemen in der Hochenergiephysik konzipiert, hat sich jedoch seither darüber hinaus allgemein in dem Wissensteilgebiet ausgezeichnet bewährt, das nun als Computer-Algebra bezeichnet wird. REDUCE ist neben dem hier beschriebenen Anwendungsgebiet für viele weitere Anwendungen sowohl der reinen wie auch der angewandten Mathematik, wie etwa Untersuchungen der diskreten Mathematik oder Berechnungen der Ingenieurwissenschaften, nützlich.

4.1.1 Benutzerschnittstellen von REDUCE

REDUCE verfügt über einen algebraischen und über einen symbolischen Modus. Der algebraische Modus ist vor allem dazu gedacht, in einer Read-Evaluate-Schleife Eingaben vom Benutzer oder aus einem Eingabestream zu empfangen und unmittelbar nach dem Ende der Berechnung ein Ergebnis in den Ausgabestream zu liefern. Der symbolische Modus ist für Programmierer gedacht, hat eine Pascal-ähnliche Syntax mit funktionalen Elementen und besitzt zudem eine weitere Schnittstelle zu Portable Standard LISP (PSL). In diesem Kapitel wird *ausschließlich* auf die Verwendung der Software im algebraischen Modus, also aus Benutzersicht, eingegangen.

Neben der üblichen Konsolenschnittstelle, bei der die Ausgabesymbole auf den Standardzeichensatz reduziert sind, verfügt REDUCE über eine graphische Einbettung in das Programm TeXmacs[1], in dem die Ergebnisse in einer Form angezeigt werden können, die zu der in dieser Arbeit verwendeten sehr ähnlich ist. Aufgrund des stetigen Wandels der Software und neu hinzukommender Funktionalitäten muß zur Zeit die Anzeigeumgebung von TeXmacs modifiziert werden, damit die Ergebnisse korrekt angezeigt werden. Eine Einleitung hierzu befindet sich auf der Homepage von Andreas Seidl[2].

[1] `www.texmacs.org`
[2] `http://www.fmi.uni-passau.de/~seidl/public/technotes/texmacs.pub.txt`

REDLOG ist ein Logik-Paket für REDUCE, welches den Umgang mit Formeln erster Stufe über einer Fülle von verschiedenen Sprachen bzw. Strukturen ermöglicht. Im Zentrum des Pakets steht zu jeder Struktur ein Quantoreneliminationsverfahren zur Verfügung. Eine aktuelle Version von REDLOG kann auf Anfrage kostenlos erhalten[3] werden.

4.1.2 Presburger Arithmetik in REDLOG

Presburger-Arithmetik ist als Kontext unter dem Namen PASF (Presburger Arithmetic Standard Form) in REDLOG eingebettet. Eine Dokumentation der generellen Ein- und Ausgabesyntax kann der aktuellen Bedienungsanleitung von REDLOG 3.0 in REDUCE 3.8 entnommen werden. In diesem Abschnitt werden die wesentlichen Neuigkeiten, die im Zusammenhang mit der Uniformität der Presburger Arithmetik hinzugekommen sind und in obiger Anleitung *nicht* enthalten sind, beschrieben. In Abbildung 4.1 ist der Start einer REDLOG-Sitzung in REDUCE abgebildet.

```
REDFRONT 2.0s by A. Dolzmann and T. Sturm, built 22-Sep-2004 ...
Loading image file :/usr/share/reduce/lisp/psl/linux/red/reduce.img

REDUCE 3.8, 15-Apr-2004, patched to 5-Aug-2004 ...

1: load redlog;

2: rlset pasf;

*** turned on switch rlsusi

{}

3:
```

Abbildung 4.1: Start einer REDLOG-Sitzung

Atomare Formeln können mit beliebigen multivariaten Polynomausdrücken eingegeben werden. Dies trifft auch für (In-)Kongruenzen mit parametrischen Moduli zu. Es können auch *beliebige* im Sinne der Definition 1.1.3 bzw. 1.1.10 korrekt gebildete Formeln eingegeben werden. Dabei sollte der Benutzer beachten, daß auch *nicht* linear quantifizierte Formeln ebenfalls eine korrekte Eingabe darstellen. Sollte allerdings im Laufe der Sitzung ein Versuch gestartet werden eine nicht linear quantifizierte Formel als Eingabe für die Quantorenelimination zu verwenden, so führt dies zu einem Abbruch mit einer Fehlermeldung. Abbildung 4.2 zeigt die Eingabe zweier Formeln und einen anschließenden Aufruf der Quantorenelimination. Im zweiten Fall führt die Eingabe der nach Beispiel 1.1.19(iii) nicht linear quantifizierten Formel zum Abbruch der Quantorenelimination mit einer Fehlermeldung.

Es ist auch möglich gebundene Quantoren mit *beliebigen* stark quantorenfreien Bounds einzugeben. Es findet zwar eine heuristische Prüfung statt, ob der Bound der Eingabe eine endliche Erfüllungsmenge besitzt. Diese, wie bereits im Zusammenhang mit der Simplifikation gebundener Quantoren angemerkt, beinhaltet *keinen* Aufruf der Quantorenelimination und kann daher *nicht* alle inkorrekt eingegebenen Formeln erkennen. Es ist vielmehr dem Benutzer überlassen korrekt gebildete Bounds als Eingabe zu verwenden. Erfüllt ein gebundener Quantor nicht die Voraussetzungen der Definition 1.1.10, so sind die erhaltenen Ergebnisse nicht verwertbar. Somit ist es generell *nicht empfehlenswert* andere gebundene Quantoren, als die mit Hilfe der Quantorenelimination als Ausgabe erhaltene, als Eingabe zu verwenden.

[3] `http://www.fmi.uni-passau.de/~redlog/`

```
4: f := ex(x,cong(x,0,a));

f := ex x (x ~a~ 0)

5: g := ex(a,all(b,a^2*x+b^2*y=r^2));

                     2    2      2
g := ex a all b (x*a  + b *y - r  = 0)

6: rlqe f;

true

7: rlqe g;

***** Illegal UPrA formula : Quantified variable b with degree 2
```

Abbildung 4.2: Eingabe von Formeln und Aufruf der Quantorenelimination

Der Kontext PASF bietet dem Benutzer die Möglichkeit durch Veränderung der Werte von globalen Schaltern, im Folgenden *Switches* genannt, die Funktionalität der Implementierung zu parametrisieren. Nun werden die *neuen* Switches und deren Bedeutung kurz vorgestellt. Jeder Funktionalität wird für Testläufe eine Abkürzung zugeordnet, die die Verwendung der entsprechenden Funktionalität symbolisiert. Ein Testergebnis wird dabei durch eine Kombination aus Abkürzungen markiert. Fehlt eine Abkürzung, so ist die entsprechende Funktionalität abgeschaltet. Das Symbol „-" steht dabei für die Abschaltung sämtlicher Techniken.

(i) **Simplifikation (S):** Damit wird die Anwendung *aller* im Kapitel 2 beschriebenen Maßnahmen zur Simplifikation von Zwischenergebnissen verstanden. Sowohl die Ein- als auch die Ausgabe werden stets vereinfacht (Switch `rlpasfsimplify`).

(ii) **Gauss-Elimination (G):** Damit wird die Bildung von Gauss-Eliminationsmengen nach Satz 3.2.27 verstanden (Switch `rlpasfgauss`). Mit der Gauss-Elimination wird in der Implementierung *stets* das triviale Gauss-Condensing nach Satz 3.2.32 angewandt.

(iii) **Gauss-Condensing (Cg):** Damit wird die Anwendung des Satzes 3.2.34 verstanden (Switch `rlpasffgc`).

(iv) **Bound-Approximation (B):** Damit wird im nicht uniformen Fall die Anwendung des Korollars 3.1.13 verstanden (Switch `rlpasfbapprox`).

(v) **Strukturelle Eliminationsmengen (E):** Damit wird die Bildung von strukturellen Eliminationsmengen und vom strukturellen Condensing verstanden (Switches `rlpasfses` bzw. `rlpasfsc`).

(vi) **Konflation (K):** Damit wird die Konflation von strukturellen Eliminationsmengen nach Satz 3.2.21 verstanden. Ist ein Testlauf sowohl mit E als auch mit K markiert, so findet der Satz 3.2.31 Anwendung (Switch `rlpasfkonf`).

4.2 Laufzeituntersuchung

In Abschnitt 1.2 wurde eine Vielzahl von verschiedenen Problemstellungen angegeben, die mit Hilfe der Presburger-Arithmetik als Formeln modelliert werden können. Dadurch wurde die Entwicklung praxisorientierter Methoden für die Simplifikation und Quantorenelimination in Presburger-Arithmetik motiviert. Nachdem die Erweiterungen vorgestellt wurden, stellt sich die Frage, welchen

Anteil diese zur Verbesserung des Verhaltens der Software beim Lösen von Problemstellungen beitragen. Neben einer systematischen Untersuchung werden zusätzlich einige Beispiele aus Abschnitt 1.2 explizit mit Angabe der Ergebnisformeln gelöst.

In diesem Abschnitt werden stets zwei Kriterien untersucht, nämlich die *Laufzeit* und die *Ausgabelänge* der quantorenfreien Ergebnisformel[4] φ'. Dabei wird zur Messung der Ausgabelänge die Anzahl der atomaren Formeln $|\mathrm{at}(\varphi')|$ herangezogen. Zur Messung der Laufzeit werden getrennt die Laufzeit der Quantorenelimination und, je nach Problemstellung, auch die Laufzeit der Auswertung der schwach quantorenfreien Ergebnisformel untersucht.

Alle Testreihen wurden auf einem Rechner mit einem AMD-Athlon (XP Mobile) 2500+ GHz Prozessor unter Verwendung des Betriebssystems Linux 2.6 durchgeführt.

4.2.1 Untersuchung des Quantoreneliminationsverfahrens

In diesem Abschnitt wird das Quantoreneliminationsverfahren nach Satz 3.1.12 untersucht. Als Vergleich dient dabei für nicht uniforme Probleme die Implementierung der Quantorenelimination aus REDUCE 3.8. Lineare Optimierungsprobleme bieten sich als eine geeignete Quelle von Eingabeformeln. Dies ist durch die einfache Struktur der Eingabeformeln begründet, welche stets existentiell quantifizierte Konjunktionen von atomaren Formeln sind. Es werden weiterhin *ausschließlich* Ungleichungen betrachtet. Die Vorgehensweise wird an einem einfachen Beispiel veranschaulicht.

Beispiel 4.2.1 Minimiere die Funktion $x + y$ bezüglich der Zusicherungen $\{y \geq -3x + 4, 2y \geq x + 2\}$. Nach Definition 1.2.4 ist die Menge der zulässigen Lösungen die Erfüllungsmenge der Formel

$$y \geq -3x + 4 \wedge 2y \geq x + 2.$$

Nach Satz 1.2.5 wird das nicht parametrische Problem durch die Formel

$$\exists x \exists y (y \geq -3x + 4 \wedge 2y \geq x + 2 \wedge z \geq x + y)$$

modelliert. Die Quantorenelimination liefert nach weniger als 10 ms eine schwach quantorenfreie Formel mit 35 atomaren Formeln. Die Expansion und anschließende Simplifikation der Ergebnisses, im Folgenden als *Auswertungsphase* bezeichnet, liefert nach 30 ms die Formel

$$z - 2 > 0 \vee (z + 2 \cong_3 0 \wedge z - 2 = 0),$$

welche offenbar zu $z > 2$ äquivalent ist. Durch Quantorenelimination mit Antworten erhält man dann für $z = 3$ nach weniger als 10 ms die Antwort $x = 1$ und $y = 2$. Das Ergebnis ist in Abbildung 4.3 veranschaulicht. Dabei repräsentieren die schwarzen Punkte zulässige Lösungen und der weiße Punkt die optimale Lösung.

Im Folgenden wird das Augenmerk *nicht* auf das Erlangen eines Zahlentupels als Lösung sondern auf die Laufzeiten und die Ergebnislängen der Quantorenelimination und der Auswertungsphase gelegt. Man beachte, daß die Auswertungsphase *keinen* Aufruf der Quantorenelimination enthält. Aus Beispiel 4.2.1 deutet sich die Vermutung an, daß die Auswertungsphase am meisten Zeit in Anspruch nimmt. Dies wird durch folgende Testreihen bestätigt.

Zunächst wird das Problem der Existenz mindestens einer zulässigen Lösung für nicht parametrische und nicht uniforme Probleme betrachtet. Dies liefert Formeln der Form

$$\exists x_1 ... \exists x_m \bigwedge_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_j \leq 0.$$

[4]Man beachte stets, daß true und false per Definition *keine* atomaren Formeln sind.

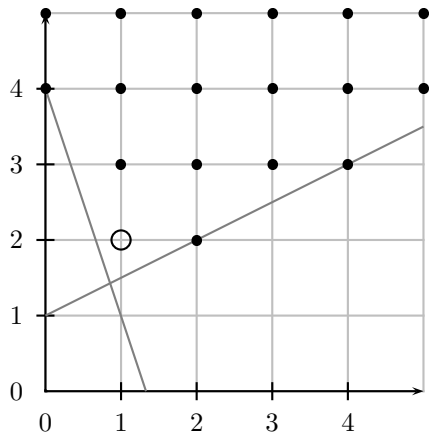


Abbildung 4.3: Schematische Darstellung des linearen Optimierungsproblems aus Beispiel 4.2.1

Die Anzahl der Zusicherungen n und die Anzahl der Variablen m stellen dabei die Parameter der Testreihen dar. Die Koeffizienten sind vom Betrag für $m = 2$ nicht größer als 5 und für $m \geq 3$ nicht größer als 3 gewählt. Dadurch wird der Vergleich mit der Implementierung in REDUCE 3.8 ermöglicht. Für die Testreihen mit $m = 4$ ist kein Vergleich angestellt worden, da die Expansion gebundener Quantoren in REDUCE 3.8 mehr als 10 min Zeit für jedes Optimierungsproblem benötigt. Jede Testreihe besteht aus 100 Optimierungsproblemen. Da die stark quantorenfreien Ergebnisformeln der Auswertungsphase variablenfrei sind, sind diese nach Anwendung der Simplifikation entweder zu true oder zu false gleich. Somit wird die Ergebnislänge der Auswertungsphase nicht explizit aufgelistet. Die Ergebnisse der Testreihen sind in Tabelle 4.1 zusammengefaßt.

		Laufzeit der QE		Ergebnislänge der QE		Laufzeit der Auswertung	
m	n	SB	REDUCE 3.8	SB	REDUCE 3.8	S	REDUCE 3.8
2	2	$<$ 10 ms	$<$ 10 ms	9	11	$<$ 10 ms	432 ms
2	3	$<$ 10 ms	$<$ 10 ms	21	24	$<$ 10 ms	16375 ms
2	4	$<$ 10 ms	$<$ 10 ms	32	36	$<$ 10 ms	52345 ms
3	2	$<$ 10 ms	$<$ 10 ms	5	8	$<$ 10 ms	22 ms
3	3	$<$ 10 ms	$<$ 10 ms	11	15	$<$ 10 ms	23450 ms
4	3	$<$ 10 ms	-	32	-	20 ms	-
4	4	67 ms	-	139	-	2243 ms	-

Tabelle 4.1: Existenz von zulässigen Lösungen

Offensichtlich liefert das in dieser Arbeit vorgestellte Quantoreneliminationsverfahren Bounds, deren Erfüllungsmengen deutlich kleiner sind als die der vorherigen Implementierung. Dies äußert sich deutlich bei der Auswertung der quantorenfreien Ergebnisformeln. Für $m \geq 5$ dauert die Auswertung der Ergebnisse des Quantoreneliminationsverfahrens in dieser Arbeit im Schnitt länger als 10 Minuten. Die Laufzeiten der Quantorenelimination wachsen allerdings nicht so schnell an. Für die Auswertung der Ergebnisse reicht dabei der Speicher nicht mehr aus um die Ergebnisformeln, welche zum Beispiel bereits für $m = 5$ und $n = 4$ eine sechsstellige Anzahl atomarer Formeln besitzen, aufzunehmen.

Als nächstes werden parametrische uniforme lineare Optimierungsprobleme untersucht. Dabei wird neben der Untersuchung der Ergebnislängen und der Laufzeiten die Fragestellung untersucht, wie stark sich die Abschätzung der Kongruenzenmoduli durch positiv definite Terme auf die Erfüllungsmengen der Bounds der Ergebnisformel auswirkt. Dazu werden für eine feste Belegung der Parameter die Auswertungszeit der Ergebnisformel für das parametrische Problem und die Auswertungszeit des entsprechenden nicht parametrischen Problems, welches durch das Einsetzen der

Parameterwerte in die Eingabeformel der Quantorenelimination entsteht, verglichen. Der Vergleich mit der alten Implementierung entfällt.

Beispiel 4.2.2 Minimiere die Funktion $x + y$ bezüglich der Zusicherungen $\{y \geq -ax + 4, 2y \geq x + 2b\}$ mit den Parametern a und b. Nach Definition 1.2.4 ist die Menge der zulässigen Lösungen für jede Stelle $\mathbf{z} \in \mathbb{Z}^2$ die Erfüllungsmenge

$$S_{\mathbf{z}}^{(a,b)}((y \geq -ax + 4 \wedge 2y \geq x + 2b)(a, b, x, y)).$$

Nach Satz 1.2.7 wird das parametrische uniforme Problem durch die Formel

$$\exists x \exists y (y \geq -ax + 4 \wedge 2y \geq x + 2b \wedge z \geq x + y)$$

modelliert. Die Quantorenelimination liefert nach 20 ms eine schwach quantorenfreie Formel mit 90 atomaren Formeln. Das Einsetzen der Parameterwerte $a = 3$ und $b = 1$ und anschließende Expansion liefert nach 170 ms die mit der aus Beispiel 4.2.1 identische Formel

$$z - 2 > 0 \vee (z + 2 \cong_3 0 \wedge z - 2 = 0).$$

Der Unterschied zwischen den Auswertungszeiten in Beispiel 4.2.1 und 4.2.2 entsteht offensichtlich durch die Abschätzung der Kongruenzenmoduli und ist *nicht vernachlässigbar*. Für die folgende Untersuchung werden wieder die Anzahl m der Variablen und die Anzahl n der Zusicherungen in den Eingabeformeln der Form

$$\exists x_1 ... \exists x_m \left(\bigwedge_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_j \leq 0 \wedge z \geq \sum_{j=1}^{m} c_j x_j \right)$$

als Testreihen-Parameter verwendet. Im Gegensatz zur letzten Testreihe wird nun die Problemstellung auf die Suche nach dem minimalen Wert der Zielfunktion in Abhängigkeit der Parameter der linearen Probleme in den Koeffizienten a_{ij} erweitert. Für alle Testläufe werden zwei Parameter a und b verwendet, die additiv und multiplikativ in die Eingabeformel einfließen. Die Ergebnisse der Testreihen sind in Tabelle 4.2 zusammengefaßt. Die Abkürzungen (P) bzw. (NP) symbolisieren parametrische bzw. nicht parametrische Testreihen, wie in Beispiel 4.2.2.

		Laufzeit der QE		Ergebnislänge der QE		Laufzeit der Auswertung	
m	n	SB (P)	SB (NP)	SB (P)	SB (NP)	S (P)	S (NP)
2	2	$<$ 10 ms	$<$ 10 ms	21	0	24 ms	$<$ 10 ms
2	3	$<$ 10 ms	$<$ 10 ms	53	0	46 ms	$<$ 10 ms
2	4	23 ms	$<$ 10 ms	118	0	116 ms	$<$ 10 ms
2	5	41 ms	$<$ 10 ms	234	0	173 ms	$<$ 10 ms
3	2	100 ms	$<$ 10 ms	436	0	$>$ 10 min	$<$ 10 ms

Tabelle 4.2: Minimaler Wert der Zielfunktion in Abhängigkeit von Parametern

Die Auswertung uniformer schwach quantorenfreier Formeln dauert also deutlich länger, als die Auswertung der Ergebnisformel der Quantorenelimination mit den in die Eingabe eingesetzten Parameterwerten. Bereits für Optimierungsprobleme mit $m = 3$ und $n = 2$ dauert die Auswertung eines parametrischen Problems durchschnittlich länger als 10 min.

4.2.2 Untersuchung der Gauss-Elimination

In diesem Abschnitt werden die Laufzeiten und die Ergebnislängen für Probleme vorgestellt, deren Formalisierung als eine Presburger-Formel Gauss-Elimination ermöglicht. Darunter fallen parametrische Gleichungssysteme, das n-Damen Problem und die Beispiele aus dem Kapitel „Programmverifikation und Abhängigkeitsanalyse". Die Elimination wird durch Zuschalten der Simplifikation (S) und des Gauss-Condensing (Cg) parametrisiert.

Als nächstes wird die Lösung des n-Damen Problems für verschiedene Werte der Parameter „Brettgröße" (k) und „Anzahl der Damen" (n) vorgestellt. Dabei wird für die Untersuchung $k = n$ gewählt. Zusätzlich wird das Problem für den Spezialfall $k = n$ geschickter formuliert durch

$$\begin{aligned} \varphi_x &= \bigwedge_{1\leq i\leq n} \bigvee_{1\leq i\leq k} x_i = i, \\ \psi &= \bigwedge_{1\leq i\leq n} \bigwedge_{1\leq j\leq n, j\neq i} (x_i \neq x_j \wedge i \neq j \wedge x_i - x_j \neq i - j \wedge x_i - x_j \neq j - i) \end{aligned}$$

Vergleiche dazu die Formulierung aus Abschnitt 1.2.2. Es ist vorteilhafter die Intervallbereiche, die für diese Testreihe relativ klein ausfallen, durch eine Disjunktion von Gleichungen zu kodieren. Dies ermöglicht die Anwendung der Gauss-Elimination. Da für $n = k$ jede der Damen in einer eigenen Spalte stehen muß, ist zusätzlich eine der Koordinaten stets redundant und kann durch den Spalten bzw. Zeilenindex kodiert werden. Somit halbiert sich in der Eingabeformel

$$\gamma = \exists x_1 ... \exists x_n (\varphi_x \wedge \psi)$$

die Anzahl der Quantoren. Die Ergebnisse der Untersuchung sind in Tabellen 4.3 und 4.4 zusammengefaßt.

Brett	Damen	-	S	SG	SGCg	GCg
1×1	1	$<$ 10 ms	$<$ 10 ms	$<$ 10 ms	$<$ 10 ms	$<$ 10ms
2×2	2	30 ms	30 ms	$<$ 10 ms	$<$ 10 ms	$<$ 10ms
3×3	3	1290 ms	1060 ms	$<$ 10 ms	$<$ 10 ms	$<$ 10 ms
4×4	4	-	-	10 ms	10 ms	30 ms
5×5	5	-	-	20 ms	20 ms	5390 ms
6×6	6	-	-	110 ms	100 ms	-
7×7	7	-	-	470 ms	470 ms	-
8×8	8	-	-	2560 ms	2670 ms	-

Tabelle 4.3: Laufzeiten für das n-Damen Problem auf einem $k \times k$ Brett

Zusätzlich zur Ausgabe der Ergebnislänge für die einzelnen Werte der Parameter wird auch das Ergebnis, das stets entweder zu true oder zu false äquivalent ist, angegeben. Es ist bemerkenswert, daß nach der Zuschaltung aller Techniken für alle Parameterwerte die Ausgabe stets entweder true oder false ist.

Brett	Damen	-	S	SG	SGCg	GCg	Ergebnis
1×1	1	0	0	0	0	0	true
2×2	2	84	82	0	0	0	false
3×3	3	8464	7960	0	0	0	false
4×4	4	-	-	0	0	0	true
5×5	5	-	-	0	0	0	true
6×6	6	-	-	0	0	-	true
7×7	7	-	-	0	0	-	true
8×8	8	-	-	0	0	-	true

Tabelle 4.4: Ergebnislängen für das n-Damen Problem auf einem $k \times k$ Brett

Für die Laufzeit und für die Ergebnislänge der Quantorenelimination spielt im Falle des n-Damen Problems die Gauss-Elimination eine maßgebliche Rolle. Dies läßt sich dadurch begründen, daß in

```
10: rlqea(f);

{{true,{x4 = 2,x3 = 4,x2 = 1,x1 = 3}},

 {true,{x4 = 3,x3 = 1,x2 = 4,x1 = 2}}}

Time: 10 ms
```

Abbildung 4.4: Antwort der Quantorenelimination für das n-Damen Problem für $k = n = 4$

der Eingabeformel jede der quantifizierten Variablen in einer separaten Gleichung vorkommt. In diesem Fall kann die Gauss-Elimination als ein intelligentes Probieren angesehen werden. Dabei entstehen *keine* gebundenen Quantoren. Es ist also vom Vorteil auch Teilformeln der Form $t > k \wedge t < l$ für $t \in \mathcal{T}$ und $l, k \in \mathbb{Z}$ als Gauss-Formeln zu betrachten. Solche Formeln können geschickt mit Hilfe der Simplifikation mit Theorie auch an *verschiedenen* booleschen Ebenen der Formel entdeckt werden. Der Betrag der Differenz $|l - k|$ spielt dabei nur dann eine Rolle, wenn die Entstehung gebundener Quantoren verhindert werden soll.

Der geringe Unterschied zwischen den Testreihen mit und ohne der Anwendung von Condensing (Reihen SG und SGCg) ist darauf zurückzuführen, daß die Eingabeformel eine Gauss-Formel ist. Dies setzt sich unter Anwendung der blockweisen Elimination auf die Zwischenergebnisse fort. Ein weiterer Grund dafür ist, daß die Teilformel ψ, die das eigentliche Problem beschreibt, ausschließlich konjunktiv verknüpfte Teilformeln enthält. Gauss-Elimination ist aber als alleiniges Mittel unzureichend. Vielmehr wird Gauss-Elimination in Kombination mit Simplifikation erst leistungsfähig. Dies belegt der Vergleich der jeweils letzten Testreihen (SGCg und GCg) aus Tabellen 4.3 bzw. 4.4.

Erweiterte Quantorenelimination liefert zusätzlich zur Lösung des n-Damen Problems auch Beispielbelegungen für die Damen. Für den Fall $k = n = 4$ ist die Ausgabe in Abbildung 4.4 und die entsprechende Belegung in Abbildung 4.5 abgebildet.

		X	
X			
			X
	X		

Abbildung 4.5: Beispielbelegung für das n-Damen Problem für $k = n = 4$

Als nächstes werden die in Abschnitt 1.2.4 angegebenen Beispiele für Abhängigkeitsanalyse und Programmverifikation mit Hilfe des Quantoreneliminationsverfahrens gelöst. Für das Beispiel 1.2.8 sind die Ergebnisse der Elimination der linear quantifizierten Formel

$$\begin{aligned} \varphi \quad &= \quad \exists i \exists j (1 \leq i \wedge i \leq n \wedge 1 \leq j \wedge j \leq i \\ &\wedge \quad 1 \leq i_1 \wedge i_1 \leq n \wedge 1 \leq j_1 \wedge j_1 \leq i_1 \\ &\wedge \quad i = i_1 \wedge j = n - j_1) \end{aligned}$$

in Tabelle 4.5 zusammengefaßt. Im Testfall SGCg liefert dabei das Verfahren die quantorenfreie Formel

$$i_1 + j_1 - n \geq 0 \wedge i_1 - j_1 \geq 0 \wedge i_1 - n \leq 0 \wedge i_1 > 0 \wedge j_1 - n < 0 \wedge j_1 > 0.$$

Kriterium	-	S	G	SG	SGCg
Laufzeit	20 ms	30 ms	< 10 ms	< 10 ms	< 10 ms
Ausgabelänge	126	126	6	6	6

Tabelle 4.5: Datenabhängigkeiten aus Beispiel 1.2.8

Quantifiziert man zusätzlich i_1 und j_1, so liefert das Verfahren im Testfall SGCgB eine schwach quantorenfreie Formel φ' mit $|\mathrm{at}(\varphi')| = 129$, welche die Existenz der Datenabhängigkeiten in Abhängigkeit vom Parameter n beschreibt. Diese kann durch Expansion der gebundenen Quantoren in 20 ms zu $n > 1$ vereinfacht werden. Die Ergebnisse der Testreihen für die Elimination von

$$\begin{aligned}\varphi &= \exists i_1 \exists j_1 \exists i \exists j (1 \leq i \wedge i \leq n \wedge 1 \leq j \wedge j \leq i \\ &\wedge \quad 1 \leq i_1 \wedge i_1 \leq n \wedge 1 \leq j_1 \wedge j_1 \leq i_1 \\ &\wedge \quad i = i_1 \wedge j = n - j_1)\end{aligned}$$

sind in Tabelle 4.6 zusammengefaßt.

Kriterium	-	S	G	SG	SGCg	SGCgB
Laufzeit	350 ms	360 ms	30 ms	40 ms	30 ms	30 ms
Ausgabelänge	2145	2076	103	103	103	75

Tabelle 4.6: Existenz der Abhängigkeiten aus Beispiel 1.2.8

Das Programm aus Beispiel 1.2.9 wird durch die uniforme Presburger-Formel

$$\begin{aligned}\varphi &= \exists i \exists j (1 \leq i \wedge i \leq n \wedge 1 \leq j \wedge j \leq n \\ &\wedge \quad 1 \leq i_1 \wedge i_1 \leq n \wedge 1 \leq j_1 \wedge j_1 \leq n \\ &\wedge \quad ((j_1 < i_1 \wedge in + j = j_1 n + i_1) \\ &\vee \quad (j_1 \geq i_1 \wedge in + j = j_1 n - i_1)))\end{aligned}$$

modelliert. Die Ergebnisse der Testreihen für diesen Fall sind in Tabelle 4.7 zusammengefaßt. Quantifiziert man zusätzlich auch die Variablen i und j, so erhält man analog zum Beispiel 1.2.8 eine Formel die die Existenz von Datenabhängigkeiten in Abhängigkeit des Parameters n beschreibt. Die Ergebnisse der Testreihen für diesen Fall sind in Tabelle 4.8 zusammengefaßt.

Kriterium	-	S	G	SG	SGCg
Laufzeit	40 ms	60 ms	30 ms	50 ms	50 ms
Ausgabelänge	367	367	316	304	300

Tabelle 4.7: Datenabhängigkeiten aus Beispiel 1.2.9

Zur exemplarischen Darstellung des Verhaltens der Gauss-Elimination bei Programmverifikation wird das in Abschnitt 1.2.4 vorgestellte Programm herangezogen. Die Ergebnisse der Testreihen sind in Tabelle 4.9 zusammengefaßt. Im Testfall SGCg liefert die Quantorenelimination die Formel

Kriterium	-	S	G	SG	SGCg
Laufzeit	3530 ms	3530 ms	950 ms	1580 ms	1580 ms
Ausgabelänge	34112	34122	9857	9854	9854

Tabelle 4.8: Existenz der Abhängigkeiten aus Beispiel 1.2.9

$$\begin{aligned}(a+b-2c-1=0 \wedge a-b>=0 \wedge b-c-1<=0) \quad &\vee \\ (a+b-2c=0 \wedge a-b>=0 \wedge b-c<=0) \quad &\vee \\ (a+b-2c-1=0 \wedge a-b<0 \wedge b-c>=0) \quad &\vee \\ (a+b-2c=0 \wedge a-b<0 \wedge b-c>=0).&\end{aligned}$$

Wertet man die Formel zum Beispiel für die Werte $a = 4$ und $b = 17$ aus, so erhält man als Ergebnis die Formel $c = 10$. Das entspricht auch der Semantik des Programms, welches das abgerundete arithmetische Mittel von a und b berechnet.

Kriterium	-	S	G	SG	SGC	SGTC
Laufzeit	-	2080	10 ms	10 ms	10 ms	10 ms
Ausgabelänge	-	21181	16	14	14	14

Tabelle 4.9: Laufzeit und Ergebnislänge für das Programm aus Beispiel 1.2.10

Als letztes werden die Ergebnisse einer systematischen Auswertung von zufällig erzeugten nicht uniformen Eingabeformeln vorgestellt. Jeder Testsatz beinhaltet 100 nicht uniforme Eingabeformeln gegebener maximaler Tiefe n mit jeweils 3 existentiellen Quantoren. Die ganzzahligen Koeffizienten sind für alle folgenden Testsätze vom Betrag nicht größer als 10. In Tabelle 4.10 sind die Durchschnittslaufzeiten und Durchschnittsergebnislängen der durchgeführten Testreihen vorgestellt.

	Laufzeit			Ergebnislänge		
Maximale Tiefe n	S	SG	SGCg	S	SG	SGCg
2	12 ms	7 ms	7 ms	58	38	36
3	15 ms	11 ms	10 ms	73	44	37
4	28 ms	12 ms	11 ms	134	46	38
5	42 ms	35 ms	32 ms	209	177	174
6	136 ms	72 ms	67 ms	671	355	315
7	326 ms	89 ms	74 ms	1174	656	593
8	1830 ms	925 ms	1026 ms	1694	1015	950

Tabelle 4.10: Durchschnittliche Laufzeit und Ergebnislänge der Testreihen

Deutlich erkennbar ist der Rückgang der Ausgabelänge um ca. 50% gegenüber der Elimination mit dem Kernverfahren[5].

[5] Eine genauere Angabe des Gewinns ist aufgrund der Länge der Testreihen nicht sinnvoll.

4.2.3 Untersuchung der strukturellen Eliminationsmengen

In diesem Abschnitt wird die Bildung von strukturellen Eliminationsmengen und Condensing untersucht. Dabei wird die Anwendung des Lemmas 3.2.31 im Vordergrund stehen, da dieses die in Abschnitt 3.2 beschriebenen Techniken sinnvoll vereinigt. Als nächstes folgt ein Beispiel, welches die Wirkung von Condensing demonstriert.

Beispiel 4.2.3 Betrachte die Elimination von $\exists x$ in $\exists x\varphi$ für

$$\varphi = x < a + b \wedge ((x < 2 + y \wedge (x \geq c \vee 2x \cong_m 0)) \vee x < c).$$

Das Quantoreneliminationsverfahren nach Satz 3.1.12 liefert nach Simplifikation des Ergebnisses eine schwachquantorenfreie Formel mit 39 atomaren Formeln. Schaltet man zusätzlich Condensing hinzu, verkleinert sich die Ausgabelänge der Formel auf 30 atomare Formeln. Schaltet man Konflation ab, so ist nach weniger als 10 ms das Ergebnis der Elimination true. Dies ist darauf zurückzuführen, daß $\exists x \Gamma_{\overline{L}((1,1))}(\varphi)$ geschickt zu true eliminiert werden kann. Diese Wirkung ist mit bisherigen Mitteln ohne Bildung einer disjunktiven Normalform der Eingabeformel *nicht erzielbar*.

Praktische Probleme, die im Rahmen dieser Arbeit betrachtet wurden, sind so gegeben, daß deren Formulierung als Presburger-Formel wenige Disjunktionen enthält. Die Stärke des Konzeptes der strukturellen Eliminationsmengen kann daher für solche Probleme nicht ausgelotet werden. Es hat sich herausgestellt, daß Condensing besonders gut zum Vergleich der Semantik zweier Presburger-Formeln verwendet werden kann. Dies wird nun an einem Beispiel demonstriert.

Beispiel 4.2.4 Man stelle fest ob die Formeln φ und φ' mit

$$\varphi = c > a + b \wedge 0 < c - d \wedge c \geq d + 1 \wedge a \leq b$$

$$\varphi' = c > a + d \wedge 0 < a - d \wedge c \geq a + 1 \wedge a \leq b$$

äquivalent sind. Das Problem ist offenbar äquivalent zu

$$\forall a ... \forall d(\varphi \longleftrightarrow \varphi').$$

Sowohl die Elimination der Quantoren mit als auch ohne Condensing benötigt die gleiche Zeit von 680 ms. Die Expansion der Ergebnisse mit dem erwünschten Ergebnis false dauert für die Ergebnisformel mit strukturellen Eliminationsmengen, Condensing und Konflation (EKB) lediglich 100 ms. Die Auswertung im anderen Fall dauert hingegen 340 ms.

Als letztes werden einige Testreihen an zufälligen Formeln vorgestellt. Die Testreihen werden durch die maximale Tiefe n der Eingabeformeln parametrisiert. Jede Formel enthält dabei 3 existentielle Quantoren. Die nicht uniformen Formeln enthalten *keine* Gleichungen, sodaß Gauss-Elimination nur selten, etwa durch Simplifikation der Zwischenergebnisse, möglich ist. Neben den Laufzeiten und Ergebnislängen der Quantorenelimination wird auch die Laufzeit der Auswertungsphase untersucht. Die Ergebnisse sind in Tabelle 4.11 zusammengefaßt.

	Laufzeit der QE		Ergebnislänge der QE		Laufzeit der Auswertung	
n	SB	SEKB	SB	SEKB	SB	SEKB
2	33 ms	18 ms	124	40	100 ms	26 ms
3	245 ms	126 ms	1947	299	1115 ms	289 ms
4	640 ms	438 ms	5291	3537	3837 ms	1574 ms

Tabelle 4.11: Strukturelle Eliminationsmengen und Condensing

Die Wirkung von strukturellen Eliminationsmengen und Condensing ist in vor allem während der Auswertungsphase der Ergebnisformeln deutlich sichtbar. Dabei ändern sich die Laufzeiten der Quantorenelimination nur geringfügig.

4.3 Zusammenfassung

In diesem Kapitel sind einige Testreihen und deren Ergebnisse angegeben worden, an denen die Wirkung der vorgestellten Techniken im Rahmen dieser Arbeit untersucht wurde. Dabei belegen die Ergebnisse, daß sowohl das vorgestellte Quantoreneliminationsverfahren als auch die Erweiterungen deutliche Verbesserungen zum Laufzeitverhalten der Implementierung beitragen.

Simplifikation der Zwischenergebnisse ist sowohl für das Kernverfahren als auch für die Bildung von strukturellen Eliminationsmengen eines der wichtigsten Werkzeuge. Besonders deutlich ist das an der Untersuchung des n-Damen Problems zu erkennen. Alle oben vorgestellten Testreihen wurden auch mit abgeschalteter Simplifikation durchgeführt. Ist dabei Gauss-Elimination nicht angewandt worden, so ist es nur für Eingabeformeln sehr geringer Länge überhaupt gelungen ein Ergebnis der Quantorenelimination zu erhalten. Dies ist aufgrund der hohen Komplexität der Quantorenelimination kein unerwartetes Ergebnis. Aus diesem Grund wurde auf die Angabe der Ergebnisse ohne Simplifikation für ganzzahlige Optimierungsprobleme und für Condensing weitgehend verzichtet.

Lineare Optimierungsprobleme mit ausschließlich Ungleichungen liefern eine gute Quelle von *repräsentativen* Problemstellungen. Daher ist es sinnvoll lineare Optimierungsprobleme als Vergleichskriterien für zukünftige Weiterentwicklungen des Verfahrens zu verwenden. Für praktische Anwendungen ist allerdings das Verfahren aufgrund zu langer Laufzeit während der Auswertungsphase in dieser Form wenig geeignet. Sollten die Ergebnisse des Verfahrens in einer Anwendung benutzt werden, so muß man zwangsläufig eine effiziente Auswertungsmethode entwickeln. Dieses Ziel wurde in dieser Arbeit allerdings nicht verfolgt. Parametrische lineare Optimierungsprobleme leiden sehr stark an der großzügigen Abschätzung der Kongruenzenmoduli durch positiv definite Terme. Für eine praktische Anwendung, in der eine Ergebnisformel etwa zur Laufzeit eines Programms ausgewertet werden muß, benötigt man bessere Abschätzungen. Dies stellt ein weiteres offenes Forschungsthema dar.

Gauss-Elimination in Verbindung mit Simplifikation liefert sehr gute Ergebnisse, wie die Testreihen aus diesem Kapitel belegen. Sowohl für Probleme der automatischen Programmverifikation als auch für die Analyse von Datenabhängigkeiten liefert Gauss-Elimination ein konkurrenzfähiges Mittel zu üblich verwendeten Bibliotheken, wie etwa PIP von Featurier [Fea88]. Testreihen für parametrische Gleichungssysteme ergaben keine neuen Einsichten. Auf die Dokumentation dieser wurde daher verzichtet.

Strukturelle Eliminationsmengen und Condensing tragen zur Verbesserung des Laufzeitverhaltens, wenn in der Eingabeformel Disjunktionen vorkommen. Der Gewinn ist vor allem während der Auswertungsphase der Ergebnisformeln bemerkbar. Es ist denkbar, daß eine alternative Formulierung bekannter Probleme mit Verwendung von Disjunktionen zu deutlich besseren Auswertungszeiten führt.

Kapitel 5

Zusammenfassung

In dieser Arbeit wurde ein effektives Quantoreneliminationsverfahren für linear quantifizierte Formeln der uniformen Presburger-Arithmetik vorgestellt. Weiterhin wurde neben einem Simplifikationsalgorithmus für Presburger-Formeln das Konzept der strukturellen Eliminationsmengen und des Condensing ausführlich behandelt. Alle vorgestellten Konzepte tragen zur Verbesserung des Laufzeitverhaltens des Gesamtverfahrens bei. Dies kann anhand der Untersuchung der Testergebnisse der Implementierung des Verfahrens im Computeralgebra-System REDUCE belegt werden.

Im ersten Kapitel ist die Presburger-Arithmetik als eine Struktur über der Sprache der Ringe, erweitert um dreistellige Relationszeichen für Kongruenzen und Inkongruenzen, eingeführt worden. Eine besondere Aufmerksamkeit verdient dabei die Einführung von gebundenen Quantoren. Gebundene Quantoren ermöglichen erst durch eine syntaktische Kodierung der parametrischen Suchbereiche für Testpunkte die gebundene uniforme Quantorenelimination in Presburger-Arithmetik mit Hilfe des vorgestellten Verfahrens. Weiterhin wurden im ersten Kapitel einige praktische ganzzahlige parametrische Probleme vorgestellt, die als linear quantifizierte Presburger-Formeln modelliert und mit Hilfe des vorgestellten Quantoreneliminationsverfahrens gelöst werden können. Mit Hilfe der uniformen Presburger-Arithmetik lassen sich unter anderem lineare ganzzahlige Optimierungsprobleme, parametrische Gleichungssysteme und kombinatorische Suchprobleme modellieren. Weiterhin ist es möglich bestimmte Probleme der Suche nach Datenabhängigkeiten in Schleifenprogrammen und der automatischen Programmverifikation als Presburger-Formeln zu kodieren. Speziell im Falle der Suche nach parametrischen Datenabhängigkeiten können Probleme modelliert werden, die mit bisherigen Mitteln nicht gelöst werden konnten.

Im zweiten Kapitel sind Simplifikationsalgorithmen für Presburger-Formeln vorgestellt worden. Neben zahlreichen Regeln zur Simplifikation von atomaren Formeln ist auch die, bereits in [DS97b] für angeordnete Körper beschriebene, Theoriesimplifikation von stark quantorenfreien Formeln der Presburger-Arithmetik umgesetzt worden.

Im dritten Kapitel wurde das Quantoreneliminationsverfahren als Elimination durch virtuelle Substitution von Testpunkten dargestellt. Dies zeigte Parallelen zwischen der Elimination in Presburger-Arithmetik und der sehr gut untersuchten reellen Quantorenelimination durch virtuelle Substitution von Testpunkten auf [Dol00]. Diese Ähnlichkeit ermöglichte eine Übertragung von Erkenntnissen, welche aus der reellen Elimination gewonnen wurden, auf die Elimination in Presburger-Arithmetik. So ist das Konzept der Gauss-Elimination und des Condensing bereits in [Dol00] für reelle Quantorenelimination beschrieben worden. Es ist zu erwarten, daß auch andere Techniken, wie zum Beispiel lokale und generische Elimination, sich effektiv auf diese Weise übertragen lassen. Das vorgestellte Quantoreneliminationsverfahren ist sowohl syntaktisch als auch semantisch von dem in [Wei97] verschieden. Jeder Suchbereich der schwach quantorenfreien Ergebnisformel wird in dieser Arbeit syntaktisch durch mehrere gebundene Quantoren kodiert. Auf der semantischen Seite liefert das eine präzisere Formulierung der Testpunkte in Abhängigkeit

von Parametern, die keine Abschätzungen der Erfüllungsmengen der Bounds in der Eingabeformel benötigt. Neben dieser Tatsache ist es auch möglich auf die Bildung des kleinsten gemeinsamen Vielfachen aller Koeffizienten der Eingabeformel zu verzichten. Daraus ergeben sich deutlich kleinere Mengen von Testtermen, die während der Auswertung der schwach quantorenfreien Ergebnisformel in die Bereiche der gebundenen Quantoren eingesetzt werden müssen. Die Bildung von strukturellen Eliminationsmengen ermöglicht das Einbeziehen der booleschen Struktur der Eingabeformel in den Eliminationsvorgang. Dies ergibt schnellere Auswertungszeiten und liefert flachere Ergebnisformeln, als das Kernverfahren. Bei der Anwendung von strukturellen Eliminationsmengen können weiterhin durch Condensing bestimmte Teilformeln während der Substitution durch false ersetzt werden.

Im vierten Kapitel wurden die vorgestellten Ansätze anhand einer Implementierung evaluiert. Der Fortschritt des hier vorgestellten Verfahrens zeigte sich deutlich anhand eines Vergleiches der Auswertungszeiten des neuen Verfahrens mit der alten Implementierung. Weiterhin stellte man fest, daß vor allem Gauss-Elimination und Gauss-Condensing erhebliche Vorteile zur Verkürzung der Laufzeit und der Länge der Ausgabeformeln für das vorgestellte Quantorenelimination beitragen. Die Bildung von strukturellen Eliminationsmengen trägt zusätzlich allgemein zu einer signifikanten Verbesserung der Auswertungszeiten der quantorenfreien Ergebnisformeln bei. Alle beschriebenen Techniken werden allerdings erst in Kombination mit der Simplifikation der Zwischenergebnisse aus dem zweiten Kapitel zu leistungsfähigen Werkzeugen.

Bei der Formulierung der in dieser Arbeit vorgestellten Konzepte wurde besonders darauf Wert gelegt, daß die Ergebnisse möglichst unverändert auch auf andere Strukturen übertragbar sind. So ist die Ausweitung beschriebener Techniken für reelle und gemischte reelle und ganzzahlige Quantorenelimination vom besonderen Interesse. Diese Arbeit kann im obigen Sinne als ein weiterer Schritt zur erfolgreichen Umsetzung applikativer Strategien für Quantorenelimination durch virtuelle Substitution von Testpunkten angesehen werden.

Index

Literaturverzeichnis

[Dol00] Dolzmann, Andreas: *Algorithmic Strategies for Applicable Real Quantifier Elimination.* University of Passau, 2000. – Dissertation

[DS97a] Dolzmann, A. ; Sturm, T.: REDLOG: Computer algebra meets computer logic. In: *SIGSAM Bulletin (ACM Special Interest Group on Symbolic and Algebraic Manipulation)* 31 (1997), Nr. 2, S. 2–9

[DS97b] Dolzmann, A. ; Sturm, T.: Simplification of Quantifier-Free Formulae over Ordered Fields. In: *JSC* 24 (1997), S. 209–231

[Fea88] Feautrier, P.: Parametric Integer Programming. In: *Operationnelle/Operations Research* 22 (1988), Nr. 3, S. 243–268

[Fea91] Feautrier, P.: Dataflow Analysis of Array and Scalar References. In: *Int. J. Parallel Programming* 20 (1991), Februar, Nr. 1, S. 23–53

[FR74] Fischer ; Rabin: Super-Exponential Complexity of Presburger Arithmetic. In: *SIAMAMS: Complexity of Computation: Proceedings of a Symposium in Applied Mathematics of the American Mathematical Society and the Society for Industrial and Applied Mathematics* Bd. 7, 1974, S. 27–41

[Gri04] Griebl, Martin: *Automatic Parallelization of Loop Programs for Distributed Memory Architectures.* University of Passau, 2004. – habilitation thesis

[Koe91] Koeppl, C.: Eine REDUCE-Implementierung eines Quantoreneliminationsverfahrens für die Presburger Arithmetik. (1991). – Diplomarbeit

[Pre29] Presburger, M.: Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In: *Comptes Rendus du I congrès de Mathématiciens des Pays Slaves* (1929), S. 92–101

[Wei88] Weispfenning, V.: The Complexity of Almost Linear Diophantine Problems. In: *JSC* 5 (1988), Februar–April, Nr. 1–2, S. 3–28

[Wei97] Weispfenning, V.: Complexity and uniformity of elimination in Presburger arithmetic. In: *ISSAC '97: Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, ACM Press, 1997, S. 48–53

Eidesstattliche Erklärung

Diese Diplomarbeit wurde selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet. Diese Diplomarbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Passau, den 28.April.2005

Aless Lasaruk