4.5 Ausnutzung der Redundanz

Sei $\mathcal{K}$ ein korrektes Inferenzsystem auf Klauseln.

Beachte

Ausnutzung des Redundanz-Begriffs ist von zentraler Bedeutung für die Effizienz von Saturierungsprozeduren.

Es gibt zwei Redundanzbegriffe

- Redundanz einer Klausel
- Redundanz einer Inferenz

Redundante Klauseln kann man beliebig einfügen und löschen.

Redundante Inferenzen kann man, muss man aber nicht ziehen.

Redundante Klauseln dienen zum Säubern der Datenbasis S.

Also

Wir betrachten Paare (S, T) mit der Eigenschaft: Jedes $C \in T$ ist bezüglich S redundant, d. h., $T \subseteq R(S)$.

Beachte

$\mathcal{K}$-Inferenzen, bei der eine Prämisse in T liegt, sind redundant.

Inferenzen mit expliziter Redundanz

Definition 4.35.

Eine **Klauselmenge mit Redundanz** ist ein Paar (S, T) mit $T \subseteq R(S)$.

Eine **Inferenz mit expliziter Redundanz** ist $\Pi = \dfrac{(S,T)}{(S',T')}$ mit $T \subseteq R(S)$ impliziert $T' \subseteq R(S')$. Π heißt **korrekt**, falls $S \models S'$ und $S - S' \subseteq R(S')$ gilt.

Beachte

Wir betrachten nur Paare (S, T) mit $T \subseteq R(S)$.

Aus $S \cup T \models S'$ folgt also $S \models S'$.

Bemerkung

Sei $\Pi = \dfrac{(S,T)}{(S',T')}$ eine Inferenz mit expliziter Redundanz.

a) Ist $T_0 \subseteq R(S)$, $S' = S - T_0$ und $T' = T \cup T_0$, so ist Π korrekt.

b) Ist $S \subseteq S'$, $S \models S'$ und $T' \subseteq T$, so ist Π korrekt.

c) Sei π eine $\mathcal{K}$-Inferenz mit Prämissen aus $S \cup T$ und Konklusion D. Ist $S' = S \cup \{D\}$ und $T' = T$, so ist Π eine **$\mathcal{K}$-Inferenz mit expliziter Redundanz**. Sie ist korrekt, da $\mathcal{K}$ als korrekt vorausgesetzt ist.

Kontrahierende Inferenzen Π

Ziel

Entwicklung von Kriterien zur „Säuberung“ von S.

Definition 4.36.

Eine Inferenz $\Pi = \dfrac{(S,T)}{(S',T')}$ heißt **kontrahierend**, falls $S \ggcurly_c S'$ gilt.

Lemma 4.29

Folgende Inferenzen Π sind korrekt und kontrahierend:

a) Tautologie-Elimination

$$\frac{(S \cup \{C\}, T)}{(S,T)} \quad \text{falls } C \text{ eine Tautologie ist}$$

b) Faktorisierung / Reflexiv-Resolution

$$\frac{(S \cup \{C \vee l \vee \cdots \vee l\}, T)}{(S \cup \{C \vee l\}, T)} \qquad \frac{(S \cup \{C \vee \neg s = s\}, T)}{(S \cup \{C\}, T)}$$

c) Subsumption

$$\frac{(S \cup \{C, D\}, T)}{(S \cup \{C\}, T)} \quad \text{falls es ein } \sigma \text{ gibt mit } \sigma(C) \subseteq D \text{ oder}$$

$C \equiv s = t$ und
$D \equiv D' \vee u[\sigma(s)]_p = u[\sigma(t)]_p$

Bemerkung zu den kontrahierenden Inferenzen Π

Tautologien und Faktorisierung

$C \vee s = s$ und $C \vee l \vee \neg l$ sind Tautologien.

Diese Tests auf Anwendbarkeit der Tautologie-Elimination, der Faktorisierung und der Reflexiv-Resolution sind rein syntaktisch. Sie sind effizient programmierbar, also werden diese Kriterien für Säuberungsschritte mit Vorrang abgeprüft.

Subsumption

a) C subsumiert $C \vee D$.
$s = t$ subsumiert $C \cup u[\sigma(s)]_p = u[\sigma(t)]_p$.
Diese Kriterien sind leicht abprüfbar.

b) C subsumiert $\sigma(C)$, falls $\sigma(C)$ eine echte Instanz von C ist, also $\sigma(C) \rhd C$ gilt. Dieses Kriterium ist auch relativ leicht abprüfbar.

c) Generell ist der Test, ob C D subsumiert schwer. Also wird der volle Test nur selten ausgeführt.

d) Sei $C \in T$ und C subsumiert D. Dann ist auch
$\Pi = \dfrac{(S \cup \{D\}, T)}{(S,T)}$ korrekt und kontrahierend.

Sichere $\mathcal{K}$-Inferenzen

Definition 4.37.
Eine $\mathcal{K}$-Inferenz $\pi = \dfrac{C_1 \ldots C_n C}{D}$ heißt **sicher**, falls C redundant in $\{C_1, \ldots, C_n, D\}$ ist.

Beispiele für sichere $\mathcal{K}$-Inferenzen

a) Sei $\mathcal{K} = \mathcal{R}_\succ$ der Resolutionskalkül
$\dfrac{C \vee A \vee A}{C \vee A}$ ist sicher.

b) Sei $\mathcal{K} = \mathcal{H}_\succ$ der Superpositionskalkül für Horn-Klauseln. Folgende Superposition-Inferenzen (Rewriting) sind sicher.

$$\frac{\Rightarrow l \doteq r \quad \Gamma \Rightarrow s[\sigma(l)]_p \doteq t}{\Gamma \Rightarrow s[\sigma(r)]_p = t}$$

$$\frac{\Rightarrow l \doteq r \quad \Gamma, s[\sigma(l)]_p \doteq t \Rightarrow \Delta}{\Gamma, s[\sigma(r)]_p = t \Rightarrow \Delta}$$

Lemma 4.30

Sei π eine sichere und korrekte $\mathcal{K}$-Inferenz mit Prämissen aus $S \cup T$, Hauptprämisse C, Konklusion D und $C \succ_c D$. Dann ist Π korrekt und kontrahierend:

$$\Pi = \frac{(S \cup \{C\}, T)}{(S \cup \{D\}, T)}$$

Das Hauptresultat zu $\mathcal{K}$

Definition 4.38.

Eine **Ableitung mit Redundanz** ist eine Folge $(S_i, T_i)_{i \in \mathbb{N}}$, so dass $\dfrac{(S_i, T_i)}{(S_{i+1}, T_{i+1})}$ für alle $i \in \mathbb{N}$ eine korrekte Inferenz ist.
Sie heißt $\mathcal{K}$-**fair**, falls $(S_i)_{i \in \mathbb{N}}$ $\mathcal{K}$-fair ist.

Lemma 4.31

Eine Ableitung $(S_i, T_i)_{i \in \mathbb{N}}$ ist $\mathcal{K}$-fair, falls $\mathcal{K}$ korrekt und monoton ist und con$(\mathcal{K}(S_\infty - R(S_\infty)) - \mathcal{K}_R(S_\infty - R(S_\infty))) \subseteq S^\infty \cup T^\infty \cup R(S^\infty \cup T^\infty)$ gilt. Dann ist S_∞ $\mathcal{K}$-saturiert bis auf Redundanz.

Satz 4.21.

Sei $\mathcal{K}$ monoton, korrekt und liftbar. $\mathcal{K}$ habe die Verkleinerungseigenschaft für Gegenbeispiele und $(S_i, T_i)_{i \in \mathbb{N}}$ sei eine $\mathcal{K}$-faire Ableitung. Dann gilt

S_0 ist H-unerfüllbar gdw $\Box \in S_n$ für ein $n \in \mathbb{N}$.

Sichere $\mathcal{S}$-Inferenzen

Lemma 4.32

Folgende $\mathcal{S}$-Inferenzen sind korrekt und sicher:

Contextual Rewriting

$$\frac{C \vee s \doteq t \quad D \vee l[\sigma(s)]_p}{D \vee l[\sigma(t)]_p} \quad \begin{array}{l}\text{falls } \sigma(s) \succ \sigma(t) \text{ und} \\ \sigma(C) \models D\end{array}$$

Constant Rewriting

$$\frac{C \vee \neg s \doteq t \vee l[s]_p}{C \vee \neg s = t \vee l[t]_p} \quad \text{falls } s \succ t$$

Beachte

In der Regel Contextual Rewriting gilt $\sigma(C) \models D$, falls $\sigma(C) \subseteq D$ gilt. Dies ist leicht zu prüfen. Contextual Rewriting ist für den Spezialfall $C = \Box$ besonders einfach zu implementieren. Contextual Rewriting und Constant Rewriting mit Klauseln aus $S \cup T$ wirken säubernd auf (S, T).

Die $\mathcal{S}$-Inferenzen Reflexiv-Resolution und Ordnungsfaktorisierung mit $\sigma = \text{id}$ sind auch sicher. Sie fallen unter die Faktorisierung in Lemma 4.29.

Grundzusammenführbarkeit (1)

Ziel

Übertragung des Redundanzkriteriums Grundzusammenführbarkeit aus Abschnitt 3.2 auf $\mathcal{S}$.

Bezeichnung

Sei $E = \{u_1 = v_1, \ldots, u_n = v_n\}$ ein Gleichungssystem und s, t Grundterme.

$s \rightarrow^0 t$ in $E^\succ$, falls es ein σ und ein $u_i \doteq v_i$ aus E gibt mit $\sigma(u_i) \succ \sigma(v_i)$, $s = t \succ_l \sigma(u_i) = \sigma(v_i)$ und $s \rightarrow t$ mit der Regel $\sigma(u_i) \rightarrow \sigma(v_i)$.

Definition 4.39.

$s = t$ heißt **in $E^\succ$ stark grundzusammenführbar**, falls für jede Grundinstanz $\sigma(s) = \sigma(t)$ gilt

- $\sigma(s) \equiv \sigma(t)$ oder
- $\sigma(s) \succ \sigma(t)$ und $\sigma(s) \rightarrow^0 s_0 \downarrow \sigma(t)$ in $E^\succ$ oder
- $\sigma(t) \succ \sigma(s)$ und $\sigma(t) \rightarrow^0 t_0 \downarrow \sigma(s)$ in $E^\succ$.

$s = t$ heißt **in S stark grundzusammenführbar**, falls es Unit-Klauseln $u_i = v_i$ in S gibt, so dass $s = t$ in $E^\succ = \{u_1 = v_1, \ldots, u_n = v_n\}^\succ$ stark grundzusammenführbar ist.

Grundzusammenführbarkeit (2)

Beachte

Sei $E^{\succ} = \{\sigma(u) \to \sigma(v) \mid u \doteq v \text{ in } E, \sigma(u) \succ \sigma(v)\}$. Gilt $s \to s_0 \downarrow t_0 \leftarrow t$ in $E^{\succ}$ und $s \to s_0$ und $t \to t_0$ jeweils bei Position $p \neq \lambda$, so ist $s = t$ in $E^{\succ}$ stark grundzusammenführbar.

Lemma 4.33

Ist $s = t$ stark grundzusammenführbar in S, so ist $s = t$ redundant in S.

Lemma 4.34

Ist $s = t$ in $S \cup T$ stark grundzusammenführbar, so ist die Inferenz

$$\frac{(S \cup \{C \vee s = t\}, T)}{(S, T \cup \{s = t\})} \qquad \text{korrekt und kontrahierend.}$$

Hauptergebnis für $\mathcal{S}$

Sei S_0 eine Menge von Klauseln.

Bezeichnung

Ein **Saturierungsprozess** von S_0 besteht aus der Berechnung einer $\mathcal{S}$-fairen korrekten Ableitung $(S_i, T_i)_{i\in\mathbb{N}}$.

Ein Ableitungsschritt (S_i, T_i) nach (S_{i+1}, T_{i+1}) besteht dabei aus (1) oder (2):

(1) dem Ziehen einer $\mathcal{S}$-Inferenz π zu S_i und
$S_{i+1} = S_i \cup \{\mathrm{con}(\pi)\}$,

(2) dem Ziehen einer korrekten und kontrahierenden Inferenz

$$\frac{(S_i, T_i)}{(S_{i+1}, T_{i+1})}$$

Beachte

Schritte nach (1) sind nötig, um Fairness zu erreichen. Schritte nach (2) sind wichtig für die Säuberung der ersten Komponente in den (S, T), also für die Effizienz des Saturierungsprozesses.

Satz 4.22.

Sei $\mathcal{A}$ ein Saturierungsprozess mit Eingabe $(S_0, \emptyset)$.

Es ist S_0 genau dann H-unerfüllbar, wenn $\mathcal{A}$ ein (S_n, T_n) liefert mit $\square \in S_n$.

Literatur

Buch

Chang, C.-L. und Lee, R.:

Symbolic Logic and Mechanical Theorem Proving, Academic Press 1990.

Robinson, A. und Voronkov, A. (Eds.):

Handbook of Automated Reasoning, Volume I, Elsevier 2001

Übersichtsartikel

Ganzinger, H. und Bachmair, L.:

Equational Reasoning in Saturation-Based Theorem Proving, in B. Bibel, D. Schmitt (Eds.) Automated Deduction - A Basis for Applications Vol. I, Kluwer 1998, pp. 353-397.

Ganzinger, H. und Niewenhuis, R.:

Constraints and Theorem Proving, in H. Comon, C. Marche, R. Treinen (Eds.): Constraints in Computational Logics: Theory and Applications, Springer 2000, LNCS 2002, pp. 159-201.

Nonnengart, A. und Weidenbach, Chr.:

Computing Small Clause Normal Forms, in A. Robinson, A. Voronkov (Eds.): Handbook of Automated Reasoning, Elsevier 2001.

Zeitschriftenartikel

Avenhaus, J., Hillenbrand, Th., Löchner, B.:

On Using Ground Joinable Equations in Equational Theorem Proving, erscheint in J. Symb. Comput.

Bachmair, L., Deshowitz, N., Plaisted, D.:

Completion without Failure, in H. Ait Kaci, M. Nivat (Eds.): Resolution of Equations in Algebraic Structures, Vol. 2, Academic Press 1989, pp. 1-30.

Bachmair, L., Ganzinger, H.:

Rewrite-Based Equational Theorem Proving with Selection and Simplification, J. of Logic and Comput., 1994, pp. 217-247.

Weitere Bücher

Bibel, W.:

Deduktion: Automatisierung der Logik, Oldenbourg 1992.

Loveland, D.:

Automated Theorem Proving: A Logical Basis, North Holland 1978.