Erzeugen von Antworten

Aufgabe: berechne $2 + 2$ (anspruchsvoll!)
Wir benutzen die rekursive Definition von '+':

$$0 + n = n \quad \text{und} \quad (n + 1) + m = (n + m) + 1$$

Wir führen ein dreistelliges Prädikatensymbol $Sum(x, y, z)$ ein, welches ausdrucken soll: 'die Summe von x und y ergibt z'.

$$\begin{aligned} F_1 &= \forall x\, Sum(0, x, x) \\ F_2 &= \forall x \forall y \forall z\, (Sum(x, y, z) \rightarrow Sum(s(x), y, s(z))) \end{aligned}$$

Gilt $\{F_1, F_2\} \models F$ mit $F = \exists x\, Sum(s^2(0), s^2(0), x)$?

$\{F_1, F_2\} \models F$ gdw. $F_1 \wedge F_2 \wedge \neg F$ unerfüllbar.

Klauselform von $F_1 \wedge F_2 \wedge \neg F$:

$$\{ \quad \{Sum(0, x, x)\}, \{\neg Sum(x, y, z), Sum(s(x), y, s(z))\}, \{\neg Sum(s^2(0), s^2(0)\} \quad \}$$

Da die leere Klausel sich ableiten lässt gilt $\{F_1, F_2\} \models F$.

Der 'Wert' von x für den $sum(s^2(0), s^2(0), x)$ lässt sich aus den Substitutionen ablesen (siehe Tafel).

Die Turme von Hanoi

Am Fusse des Himalaya, nicht weit von der Mündung des roten Flusses, liegt die vietnamesische Stadt Hanoi. Eine Legende erzählt, dass einmal drei goldene Säulen in einem Tempel in Hanoi standen. Auf einer der Säulen befanden sich 100 Scheiben, jedesmal eine kleinere auf einer größeren Scheibe. Ein alter Mönch bekam den Auftrag, den Scheibenturm von der ersten auf die dritte Säule zu versetzen, wobei er jedesmal nur die oberste Scheibe von einem Turm nehmen und nie eine größere Scheibe auf eine Kleinere legen durfte. Wenn er alle Scheiben auf die dritte Säule gelegt hat kommt das Ende der Welt.

Aufgabe: berechne eine Sequenz von Zügen, die n Scheiben von der ersten in die zweite Säule versetzen.

Idee: für $n = 1$ trivial, für $n > 1$ setze die oberen $n - 1$ Scheiben d Turmes auf die dritte Säule, setze die untere Scheibe auf die zwei Säule und setze die $n - 1$ auf der dritten Säule zwischengelagerte Scheiben auf die dritte Säule.

Die Formel

Wir führen ein vierstelliges Prädikatensymbol $Vers(n, x, y, z)$ ein, welches ausdrucken soll: 'es ist möglich, n Scheiben von x nach y mit z als Zwischenlager zu versetzen'

$$\begin{aligned} F_1 &= \forall x, y, z : Vers(0, x, y, z) \\ F_2 &= \forall n, x, y, z : \ ((Vers(n, x, z, y) \wedge Vers(n, z, y, x)) \\ &\qquad \longrightarrow \\ &\qquad Vers(s(n), x, y, z) \ ) \end{aligned}$$

Gilt $\{F_1, F_2\} \models F$ mit $F = Vers(s^4(0)), erste, zweite, dritte)$?

Hornklausel

- Tatsachenklauseln sind einelementige positive Klauseln

$$\{Sum(0, x, x)\}, \{Vers(0, x, y, z)\}$$

- Prozedurklauseln haben die Form $\{P, \neg Q_1, \ldots, \neg Q_n\}$. P ist der Prozedurkopf und $Q_1, \ldots, Q_n$ der Prozedurkörper.

$$\{\neg Sum(x, y, z), Sum(s(x), y, s(z)\},$$
$$\{\neg Vers(n, x, z, y), \neg Vers(n, z, y, x)), Vers(s(n), x, y, z)\}$$

- Zielklauseln sind einelementige negative Klauseln

$$\{Sum(s^2(0), s^2(0), x)\}, \{Vers(s^4(0), erste, zweite, dritte)\}$$

Es handelt sich stets um (prädikatenlogische) Hornklauseln

LUSH-resolution

Die LUSH-Resolution (linear resolution with unrestricted selection for Horn clauses) führt die folgenden Restriktionen ein:

- Bei jedem Resolutionsschritt müssen eine Zielklausel und eine Tatsachen- oder Prozedurklausel resolviert werden.
- Die Tatsachen- oder Prozedurklausel muss eine Ausgangsklausel sein.

Satz: Die LUSH-Resolution ist vollständig für Hornklauseln

Prolog

Prolog-Programm = Menge von Tatsachen- und Prozedurklauseln.

Eine Zielklausel stellt die zu lösende Aufgabe dar.

Der Prolog-Interpreter versucht mit SLD-Resolution, eine Verfeinerunng der LUSH-Resolution, die Aufgabe zu lösen.

Die primitiven Datenstruktur sind Terme, mit denen sich Listen und Bäume leicht darstellen lassen.

Viele Abweichungen von einer 'reinen' prädikatenlogischen Lösung.

Deklarative Programmierung nach Wikipedia

Die deklarative Programmierung ist ein Programmierparadigma. Es ist ursprünglich als Gegensatz zur imperativen Programmierung hervorgegangen. Die Grundidee besteht darin, nicht mehr, wie beim imperativen Paradigma, zu beschreiben, wie etwas zu geschehen hat bzw. was zu tun ist, sondern nur noch welches Ergebnis gewünscht ist. Es sollte also nicht mehr der Lösungsweg programmiert werden, sondern nur noch angegeben werden, welches Ergebnis gewünscht wird. Den richtigen Lsungsweg zu beschreiten ist dann Aufgabe des Rechners, der dazu eine hinreichend mächtige Inferenzmaschine einsetzen muss.