Zusammenhänge zwischen Term-Ersetzungs-Systemen und regulären Baum-Sprachen

Johannes Waldmann
Institut für Informatik
Universität Leipzig

Nachfolgermengen

Term-Ersetzungs-System R
gegeben durch Regelmenge
Relation $\to_R$

Reguläre Baum-Sprache L
gegeben durch endlichen Automaten A

Nachfolgermenge:
$R(L) = \{t' \mid t \in L, t \to_R t'\}$
$R^*(L) = \{t' \mid t \in L, t \to_R^* t'\}$

gesucht: Automat für
(Approximation von) $R(L), R^*(L)$,

Anwendung (Genet, 1998): Programmanalyse
R: Programm
L: Eingabe
$R^*(L)$: Zwischenresultate
$R^*(L) \cap NF$: Ausgabe

Vorsicht!

$R'(L)$ muß nicht regulär sein.

Beispiel (Kopieren)

$R = \{f(x) \to g(x,x)\}, L = f(h^*(0))$

$R(L) = \{g(h^n(0), h^n(0)) \mid n \geq 0\}$

Beispiel (Kopieren ohne Kopieren)

$R = \{a \to bc, cb \to bc\}$
(linear, terminierend, konfluent)

$L = a^*$

$aaa \to^* bcbcbc \to^* bbcbcc \to bbbccc$

$R^*(L) \cap NF = \{b^n c^n \mid n \geq 0\}$ nicht regulär

Ausdrucksstärken

TRS im Allgemeinen:
Turing-vollständig, unentscheidbar
endlicher Automat: alles effektiv entscheidbar

Endlichen Automaten als Grund-TRS
Signatur erweitern (um Automatenzustände)
Ersetzungsrelation $\to_A$.

$R^*(L) \in Reg$ ist im Allg. nicht entscheidbar

$R^*(L)$ ist nur konstruierbar,
wenn R mit A kooperiert.

verschiedene Beispiele.

gesucht: möglichst umfassende Beschreibung.

Konstruktion für $R^*(L)$

Signatur F, Automat A mit Zuständen Q.

Sei $X \in T(F, Q)$, und $X \to_A^* q$.

Wenn $X \to_R Y$,
dann $Y \to_A^* q$ als Regel für A hinzufügen.

Beispiel:

Signatur:
$e(\cdot), o(\cdot), n$ (Listen),
$t(\cdot, \cdot), l$ (binäre Bäume),
$f(\cdot, \cdot)$ inorder-Durchlauf mit Akkumulator.

TRS $R = \left\{ \begin{array}{c} f(l, z) \to o(z), \\ f(t(x, y), z) \to f(x, e(f(y, z))) \end{array} \right\}$

Sprache $B \leftarrow l, B \leftarrow t(B, B)$ (alle Bäume).

Suchen $C = R^*(f(B, n))$ oder $C \cap NF$

Kommentar zur Konstruktion

Evtl. neue Zustände erforderlich,
dabei evtl. neue Redexe produziert,
also wieder vervollständigen.

Termination nicht garantiert.

(Genet, 1998):
Durch Identifikation von Zuständen
Verfahren abbrechen.
liefert Approximation für L' (von oben).

Selbst wenn endlicher A' für L' existiert,
ist nicht garantiert, daß er gefunden wird.

Ground Tree Transducer (GTT)

Endliche Automaten $\sim$ reguläre *Mengen*.

Welche *Relationen* $\sim$ *regulär*?

$G = (A, B)$ mit
A, B Automaten über gleicher Signatur.

$$\to_G \quad := \quad \to_A^* \circ \leftarrow_B^*$$

Satz (Dauchet, Tison, Coquide, Gilleron, 1985):
GTT-Relationen sind
abgeschl. unter Komposition, Iteration.
Sei $\to_G$ per GTT,
dann $\{t \mid s \in L, s \to_G t\}$ regulär.

Satz: Für Grund-TRS:
Konfluenz entscheidbar.

Eingeschränkte TRS

GTT $\sim$ grund-TRS

Variablen, trotzdem R^* konstruierbar.

flach (shallow): in $l \to r$,
wenn $v \in Var(l) \cap Var(r)$,
dann in Tiefe 1 in l und 0 oder 1 in r.

Beispiel: $g(x, h(z)) \to f(0, x), h(y) \to y$.

Satz (Comon, 1995):
R linear flach: $R^{-*}(L)$ konstruierbar.

wachsend (growing): ... Tiefe 1 in l.

Beispiel: $g(x, h(z)) \to f(0, h(h(x)))$.

Satz (Jacquemard, 1996):
R linear wachsend: $R^{-*}(L)$ konstruierbar.

Reduktions-Strategien

Def: Redex R ist *notwendig*:
jede Reduktion zu Normalform
reduziert R oder eines seiner Residuen.

Satz (Huet, Levy, 1991):

In orthogonalen
(linkslinear, nichtüberlappend) TRS:

a) jede Nicht-Normalform
enthält notwendigen Redex;

b) Reduktion von notwendigen Redexen
ist optimale normalisierende Strategie

notwendig ist im Allg. nicht entscheidbar.

Approximation von Reduktions-Strategien

(Durand, Middeldorp, 1995)

a) entscheidbare Approximation
für *Notwendigkeit*

b) entscheidbares Kriterium für TRS,
so daß jeder reduzierbare Term
einen (approx.) notwendigen Redex enthält

benutzen GTT

vgl. mit *strictness analysis*
in funktionalen Programmen.

(head/root normal form statt normal form,
root-neededness)

Beweis induktiver Theoreme

Beispiel: Signatur $\{0, s(\cdot)\}$ (Zahlen),

System $\{s(x) + y \to s(x + y)\}$ (Addition)

Aussage $x + y = y + x$ ist induktives Theorem

(gilt nur in initialen Modellen = Grund-Termen).

Grund-Konfluenz, Grund-Vervollständigung.

Mengen von Grund-Instanzen linearer Terme sind regulär.

Nichtlinear: Automaten mit Constraints.

(Bogaert, Tison)

Linearisierung

Wenn TRS R links-linear,
dann Normalformen regulär.

Für NF: rechte Seiten egal.

Nicht alle regulären Sprachen
als Normalformen darstellbar
(nur nicht-zählende)

Wenn nicht links-linear?

Satz (Hofbauer, Kucherov, . . . , 1992):
Entscheidbar, ob NF regulär.
Wenn ja, dann auch konstruierbar.

D. h. die linken Seiten
könnnen linearisiert werden.

Wie weiter?

Für $R = CL(S)$ sollte $R^{-*}(N)$
(normalisierende Terme) konstruierbar sein.

Habe starke Vermutung für das Resultat.
(Det. Aut., etwa 40 Zustände, 1000 Regeln.)

Zum Beweis helfen vielleicht
Reduktionsordnungen auf regulären Mengen.

Konstruktion von $R^*(L)$ auffassen
als Vervollständigung (analog Knuth-Bendix)

Kooperation mit LIFL (Lab. Fund. Inf. Lille)
(Tison, Dauchet ...),

weitere Algorithmen auf Baumautomaten/GTTs
implementieren (System `RX` erweitern)