Constraint based program analysis

- syntaxgesteuert
- abstrakte Interpretation
- finde lokale Constraints
- kombiniere, löse
- beweist Programm-Eigenschaften

  (Das dritte Argument von ... ist nie leer)
- hilft beim Kompilieren

  (wir können automatisch spezialisieren)

Set constraints: Definition

- Signatur Σ, Term-Algebra $\mathsf{Term}(\Sigma)$
- Signatur $\Omega = \{\top, \bot, \neg, \cup, \cap\}$
- set constraint: $x \subseteq y, x \not\subseteq y$ mit $x, y \in \mathsf{Term}(\Sigma \cup \Omega, V)$
- set-constraint-system: Konjunktion von set constraints
- Lösung: Belegung $B : V \to 2^{\mathsf{Term}(\Sigma)}$

Set constraints: Eigenschaften

- existiert Lösung?
- Lösung ist eindeutig?
- existiert einfache (reguläre) Lösung?

regulär = Tupel von regulären Baumsprachen

ordnen Belegungen B komponentenweise per Inklusion

- minimale, maximale Lösung?
- sind eindeutig?

Set constraints: Resultate

- Lösbarkeit ist entscheidbar

  aber Vorsicht: ist NEXPTIME-vollständig
- jede minimale, jede maximale Lösung ist regulär
- konstruierbar (GTSA, verallgemeinerte Baum-Mengen-Automaten)

Baum-Automaten ⇒ Set Constraints

- Automat $A = (\Sigma, Q, Q_F, \Delta)$
- Regel $f(q_1, \ldots, q_k) \to q_0$
- Constraint $f(V_1, \ldots, V_k) \subseteq V_0$
- hat minimale Lösung B
- mit $\mathsf{Lang}(A) = B(Q_F)$

Set Constraints ⇒ Baum-Automat

TSA (GTSA): (generalized) tree set automaton.

Eingabe: Funktion $S : \mathsf{Term}(\Sigma) \to E$

hier: $E = \{0,1\}^n$, kodiert eine Belegung B:

$B(V_k) = \{t \mid t \in \mathsf{Term}(\Sigma), S(t)|_k = 1\}$

Konstruieren $G \in$ GTSA mit
$\mathsf{Lang}(G) = \{S \mid B \models \mathsf{SC}\}$

Entscheidbar, ob $\emptyset \neq \mathsf{Lang}(G)$

ein reguläres Element in $\mathsf{Lang}(G)$ konstruierbar

GTSA

G ist E-wertiger GTSA:

- $G = (\Sigma, E, Q, Q_F, \Delta)$
- $\Delta \subseteq \bigcup_k \Sigma_k \times Q^k \times E \times Q$
- $Q_F \subseteq 2^Q$

r ist Lauf von $G \in$ GTSA auf $S : \mathsf{Term}(\Sigma) \to E$:

- $r : \mathsf{Term}(\Sigma) \to E$
- $\forall k, f \in \Sigma_k, t = f(t_1, \ldots, t_k)$:

  $(f, r(t_1), \ldots, r(t_k), S(t), r(t)) \in \Delta$

$S \in \mathsf{Lang}(G) \iff$
$\exists$Lauf r von G auf S mit range $r \in Q_F$

Set constraints ⇒ GTSA: Zustände

Beispiel: $f(g(X), Y) \subseteq f(Z, Z)$

Variablen $V_1, \ldots, V_n$, Bsp: X, Y, Z.

Bewertung nach $E = \{0,1\}^n$, Bsp: $\{0,1\}^3$.

Teilterme $T_1, \ldots, T_m$, Bsp:
$X, Y, Z, g(X), f(g(X), Y), f(Z, Z)$,
$f(g(X), Y) \subseteq f(Z, Z)$.

Zustände $Q = \{0,1\}^m$, Bsp: $\{0,1\}^7$.

Set constraints ⇒ GTSA: Transitionen

$f(q_1, \dots, q_k) \xrightarrow{l} q \in \Delta$

- (Interpretation der Variablen:)

  $\forall 1 \leq j \leq m$: wenn $T_j = V_i$,

  dann $q(T_j) = l_i$
- (Interpretation der Nicht-Variablen:)

  $\forall 1 \leq j \leq m$: wenn $T_j = e \notin V$,

  dann $q(T_j) = 1 \iff$

  - $e = f(e_1, \dots, e_k)$ und
  - $\forall 1 \leq i \leq k : q_i(e_i) = 1$

Set constraints ⇒ GTSA: Akzeptanz

- wenn SC $= x \subseteq y$,

  dann $Q_F = \{h \in 2^Q \mid \forall x \in h : x(SC) = 1\}$
- wenn SC $= x \not\subseteq y$,

  dann $Q_F = \{h \in 2^Q \mid \exists x \in h : x(SC) = 0\}$

Wenn SC positiv ist, reichen Zustände $Q' = \{q \mid q(SC) = 1\}$ und $Q'_F = 2^{Q'}$.

Teilklassen: Einschränkungen

- positiv: keine negativen constraints:

  nur $x \subseteq y$, kein $x \not\subseteq y$.

  Erfüllbarkeit ist NEXPTIME-vollständig.
- definit: links keine Komplemente, rechts überhaupt keine Operationen:

  $x \subseteq y$ mit $x \in \mathsf{Term}(\Sigma \cup \{\top, \bot, \cup, \cap\}, V)$, $y \in \mathsf{Term}(\Sigma, V)$

  Erfüllbarkeit ist DEXPTIME-vollständig.

  Bereich der Lösungen ist complete lower semilattice.
- atomic: keine Mengenoperationen, aber Tests auf Leere erlaubt

  $x \subseteq y, x \neq \bot, x \cap y \neq \bot$

  Erfüllbarkeit in kubischer Zeit entscheidbar.

Teilklassen: Erweiterungen

Für definite und atomische:

- Durchschnitte rechts:

  $X \subseteq Y \cap Z \iff X \subseteq Y \wedge X \subseteq Z$
- Vereinigungen links:

  $X \subseteq Y \cap Z \iff X \subseteq Y \wedge X \subseteq Z$

Membership expressions für definite Constraints: auf linker Seite auch $\{x \mid \Phi(x)\}$, wobei Φ beliebige Formel erster Ordnung (mit Quantoren für Grundterme).

Projektionen:
$f_i^-(X) = \{t \mid t = f(t_1, \dots, t_k), t_i \in X\}$

$f_i^-(X) \subseteq Y$ ist einfach (z. b. bleibt definit), aber $X \subseteq f_i^-(Y)$ ist schwer. Trotzdem entscheidbar.