Automatische Strukturen

WS 20/21

Prof. Dr. Dietrich Kuske

12. Oktober 2020

Generelle Frage

Es gibt verschiedene Datenstrukturen für endliche Graphen.

Gibt es auch die Möglichkeit, in Programmen unendliche Graphen oder allgemeiner „Strukturen“ zu verwenden?

Beispiel

1. Sei $\mathcal{M}$ eine Turingmaschine.

   Knotenmenge: Alle Konfigurationen

   Kanten (u, v): falls $u \vdash v$

   Dieser Graph heißt *Konfigurationsgraph von* $\mathcal{M}$.

2. $(\mathbb{N}, \leq) =: \omega$ ist unendlicher Graph

3. Sei $n \in \mathbb{N}$

   $\bar{a} = (a_1, \ldots, a_n)$ und $\bar{b} = (b_1, \ldots, b_n)$ Tupel aus $\mathbb{N}^n$.
   $\bar{a} <_{\mathrm{lex}} \bar{b}$, falls es $1 \leq i \leq n$ gibt mit $(a_1, \ldots, a_{i-1}) = (b_1, \ldots, b_{i-1})$ und $a_i < b_i$.

   $\omega^n := (\mathbb{N}^n, \leq_{\mathrm{lex}})$ ist unendlicher Graph.

4. $k, \ell \in \mathbb{N}$, $\bar{a} \in \mathbb{N}^k$, $\bar{b} \in \mathbb{N}^\ell$.

   $\bar{a} \leq_{\mathrm{llex}} \bar{b}$, falls $k < \ell$ oder $k = \ell$ und $\bar{a} \leq_{\mathrm{lex}} \bar{b}$

   $\omega^\omega := (\bigcup_{1 \leq n} \mathbb{N}^n, \leq_{\mathrm{llex}})$ ist unendlicher Graph.

5. $(\mathbb{Q}, \leq)$ ist unendlicher Graph.

6. $u \in \Sigma^*$ ist *Teilwort* von $v \in \Sigma^*$, wenn es Wörter $u_1, u_2, \ldots, u_n, v_0, v_1, \ldots, v_n \in \Sigma^*$ gibt mit $u = u_1u_2 \ldots u_n$ und $v = v_0u_1v_1u_2v_2 \ldots v_{n-1}u_nv_n$.

   Hierfür schreiben wir $u \preceq v$. Dann ist $(\Sigma^*, \preceq)$ unendlicher Graph.

7. $(\mathcal{P}(\mathbb{N}), \subseteq)$ ist ein unendlicher Graph.

   Sei $\mathcal{P}_{\text{fin}}(\mathbb{N})$ die Menge der endlichen Teilmengen von $\mathbb{N}$. Dann ist $(\mathcal{P}_{\text{fin}}(\mathbb{N}) \cup \{A \subseteq \mathbb{N} \mid \mathbb{N} \setminus A \in \mathcal{P}_{\text{fin}}(\mathbb{N})), \subseteq)$ ein unendlicher Graph, die *Boolesche Algebra der endlichen und co-endlichen Teilmengen von* $\mathbb{N}$

8. $\mathcal{N} = (\mathbb{N}, +, \cdot)$ und $(\mathbb{Q}, +, \cdot)$ sind unendliche „Strukturen“

9. $\mathcal{T}_\Sigma = (\Sigma^*, \leq_{\text{pref}}, (S_a)_{a \in \Sigma}, \text{el})$ ist unendliche „Struktur“ mit $\leq_{\text{pref}} = \{(u, uv) \mid u, v \in \Sigma^*\}$, $S_a = \{(u, ua) \mid u \in \Sigma^*\}$ für $a \in \Sigma$ und $\text{el} = \{(u, v) \mid u, v \in \Sigma^*, |u| = |v|\}$.

Ziel Datenstruktur für solche unendlichen Strukturen, die es erlaubt

- Operationen auf den Strukturen auszuführen (z.B. Kanten einzufügen oder zu löschen bzw. aus zwei geg. Strukturen eine neue - wie das direkte Produkt - zu berechnen)
- Aussagen über die Strukturen zu machen (z.B. in `if`-Anweisungen)

1 Strukturen, Logik 1. Stufe und effektive Strukturen

Signatur $\tau = (\mathcal{R}, \mathcal{C}, \mathrm{ar})$ mit

- $\mathcal{R}$ endlicher Menge von *Relationssymbolen*
- $\mathcal{C}$ endlicher Menge von *Konstantensymbolen*
- $\mathrm{ar}\colon \mathcal{R} \to \mathbb{N}_{\geq 1}$ *Stelligkeitsabbildung*

Beispiele

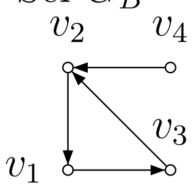
- $\tau_{\mathrm{graph}} = (\{E\}, \emptyset, \mathrm{ar})$ mit $\mathrm{ar}(E) = 2$
- $\tau_{\mathcal{N}} = (\{\mathrm{add}, \mathrm{mult}\}, \emptyset, \mathrm{ar})$ mit $\mathrm{ar}(\mathrm{add}) = \mathrm{ar}(\mathrm{mult}) = 3$

τ-Struktur $\mathcal{A} = (A, (R^{\mathcal{A}})_{R\in\mathcal{R}}, (c^{\mathcal{A}})_{c\in\mathcal{C}})$ mit

- $A = ||\mathcal{A}||$ nichtleere Menge (*Grundmenge*)
- für $R \in \mathcal{R}$ mit $\mathrm{ar}(R) = n$: $R^{\mathcal{A}} \subseteq ||\mathcal{A}||^n$
- für $c \in \mathcal{C}$: $c^{\mathcal{A}} \in ||\mathcal{A}||$

Beispiele

- Sei $G_B = (V, E^{G_B})$ der folgende Graph



d.h., $V = \{v_1, v_2, v_3, v_4\}$ und $E^{G_B} = \{(v_1, v_3), (v_3, v_2), (v_2, v_1), (v_4, v_2)\}$. Dann ist G_B eine endliche τ_{graph}-Struktur.

$\omega = (\mathbb{N}, \leq)$ ist τ_{graph}-Struktur mit $E^{\omega} = \{(m, n) \in \mathbb{N} \mid m \leq n\}$

- $\mathcal{N} = (\mathbb{N}, \mathrm{add}^{\mathcal{N}}, \mathrm{mult}^{\mathcal{N}})$ ist $\tau_{\mathcal{N}}$-Struktur mit $\mathrm{add}^{\mathcal{N}} = \{(m, n, m+n) \mid m, n \in \mathbb{N}\}$ und $\mathrm{mult}^{\mathcal{N}} = \{(m, n, m \cdot n) \mid m, n \in \mathbb{N}\}$
  analog: rationale Zahlen mit Addition und Multiplikation etc.

Variablen $\mathcal{V} = \{v_1, v_2, \dots\}$ abzählbare Menge

τ-Terme $\mathcal{V} \cup \mathcal{C}$

τ-Formeln, freie Variable und Quantorenrang *τ-Formeln* sind diejenigen Zeichenketten, die sich wie folgt erzeugen lassen

1. t_1, t_2 τ-Terme
   $\Rightarrow \varphi = (t_1 = t_2)$ τ-Formel mit *freien Variablen* $\mathrm{frei}(\varphi) = \{t_1, t_2\} \cap \mathcal{V}$ und *Quantorenrang* $\mathrm{qr}(\varphi) = 0$

2. $t_1, \dots, t_n$ τ-Terme, $R \in \mathcal{R}$ mit $\mathrm{ar}(r) = n \Rightarrow$
   $\varphi = R(t_1, \dots, t_n)$ τ-Formel mit $\mathrm{frei}(\varphi) = \{t_1, \dots, t_n\} \cap \mathcal{V}$ und $\mathrm{qr}(\varphi) = 0$

3. φ, ψ τ-Formeln, $x \in \mathcal{V} \Rightarrow$

   - $\xi = \neg\varphi$ τ-Formel, $\mathrm{frei}(\xi) = \mathrm{frei}(\varphi)$, $\mathrm{qr}(\xi) = \mathrm{qr}(\varphi)$
   - $\xi = \varphi \wedge \psi$ τ-Formel, $\mathrm{frei}(\xi) = \mathrm{frei}(\varphi) \cup \mathrm{frei}(\psi)$, $\mathrm{qr}(\xi) = \max(\mathrm{qr}(\varphi), \mathrm{qr}(\psi))$
   - $\xi = \exists x : \varphi$ τ-Formel, $\mathrm{frei}(\xi) = \mathrm{frei}(\varphi) \setminus \{x\}$, $\mathrm{qr}(\xi) = \mathrm{qr}(\varphi) + 1$

$\mathrm{FO}[\tau]$ ist die Menge aller τ-Formeln, ein *τ-Satz* ist eine τ-Formel ohne freie Variable

„$\varphi = \varphi(x_1, \dots, x_n)$“ bedeutet: „$\varphi \in \mathrm{FO}[\tau]$ und $\mathrm{frei}(\varphi) \subseteq \{x_1, \dots, x_n\}$“

Beispiele

- $\varphi_\Delta = \exists y \exists z : (E(x,y) \wedge E(y,z) \wedge E(z,x))$ ist τ_{graph}-Formel mit $\mathrm{frei}(\varphi_\Delta) = \{x\}$
- $\psi = \neg\exists x \neg\exists y \exists z : (E(x,y) \wedge E(y,z) \wedge E(z,x))$ (d.h. $\neg\exists x \neg\varphi_\Delta$) ist τ_{graph}-Satz

Belegung in τ-Struktur $\mathcal{A}$ Abbildung $\alpha : \mathcal{V} \cup \mathcal{C} \to ||\mathcal{A}||$ mit $\alpha(c) = c^{\mathcal{A}}$ f.a. $c \in \mathcal{C}$

Gültigkeit von τ-Formeln Für Tupel $\overline{x} = (x_1, \ldots, x_n)$ von Variablen, Formel φ mit frei$(\varphi) \subseteq \{x_1, x_2, \ldots, x_n\}$ und Struktur $\mathcal{A}$ definieren wir induktiv die Menge $\varphi(\overline{x})^{\mathcal{A}}$ (oder kürzer $\varphi^{\mathcal{A}}$ der φ erfüllenden Tupel:

1. $\varphi = (t_1 = t_2)$: $\varphi(\overline{x})^{\mathcal{A}} = \{(\alpha(x_1), \ldots, \alpha(x_n)) \mid \alpha$ Belegung mit $\alpha(t_1) = \alpha(t_2)\}$
2. $\varphi = (R(t_1, \ldots, t_k))$: $\varphi(\overline{x})^{\mathcal{A}} = \{(\alpha(x_1), \ldots, \alpha(x_n)) \mid \alpha$ Belegung mit $(\alpha(t_1), \ldots, \alpha(t_n)) \in R^{\mathcal{A}}\}$
3. φ, ψ Formeln mit frei$(\varphi) \cup$ frei$(\psi) \subseteq \{x_1, \ldots, x_n\}$:
   - $(\neg\varphi)(\overline{x})^{\mathcal{A}} = ||\mathcal{A}||^n \setminus \varphi(\overline{x})^{\mathcal{A}}$
   - $(\varphi \wedge \psi)(\overline{x})^{\mathcal{A}} = \varphi(\overline{x})^{\mathcal{A}} \cap \psi(\overline{x})^{\mathcal{A}}$
4. φ Formel mit frei$(\varphi) \subseteq \{x_1, x_2, \ldots, x_n, x\}$ (also frei$(\exists x\, \varphi) \subseteq \{x_1, \ldots, x_n\}$):

   $(\exists x\, \varphi)(\overline{x})^{\mathcal{A}} = \{(a_1, a_2, \ldots, a_n) \mid$ es gibt $(a_1, \ldots, a_n, a) \in \varphi(\overline{x}, x)^{\mathcal{A}}\}$

Für $\overline{a} \in \varphi(\overline{x})^{\mathcal{A}}$ schreiben wir auch $(\mathcal{A}, \overline{a}) \models \varphi$ oder $\mathcal{A} \models \varphi[\overline{a}]$.

Beispiele

- $\varphi_{\Delta}^{G_B} = \{v_1, v_2, v_3\}$ (siehe Beispiel oben)
- $\psi^{G_B} = \emptyset$, $(\neg\psi)^{G_B} = \{()\}$ wegen $\text{frei}(\psi) = \emptyset$

Abkürzungen

$$\begin{aligned} (\varphi \vee \psi) &= \neg(\neg\varphi \wedge \neg\psi) & \qquad (\varphi \to \psi) &= (\neg\varphi \vee \psi) \\ (\varphi \leftrightarrow \psi) &= (\varphi \to \psi) \wedge (\psi \to \varphi) & \qquad (\forall x : \varphi) &= (\neg\exists x : \neg\varphi) \end{aligned}$$

dann gilt z.B. $(\mathcal{A}, \overline{a}) \models \varphi \vee \psi$ gdw. $(\mathcal{A}, \overline{a}) \models \varphi$ oder $(\mathcal{A}, \overline{a}) \models \psi$, also $(\varphi \vee \psi)^{\mathcal{A}} = \varphi^{\mathcal{A}} \cup \psi^{\mathcal{A}}$ usw.

Ziel Datenstruktur für unendliche τ-Strukturen, so daß das folgende Problem entscheidbar ist:

Eingabe: τ-Struktur $\mathcal{A}$ (geg. durch die Datenstruktur) und Satz φ
Frage: Gilt $\mathcal{A} \models \varphi$?

Etwas formaler:

Definition. Sei D eine formale Sprache und, für $d \in D$, sei $\mathcal{S}(d)$ eine τ-Struktur. Das *uniforme Auswertungsproblem* für $(D, \mathcal{S})$ ist das folgende Problem:

Eingabe: $d \in D$ und Satz φ
Frage: Gilt $\mathcal{S}(d) \models \varphi$?

1.1 Effektive Strukturen

Definition. Eine Struktur $\mathcal{A}$ ist *effektiv* oder *berechenbar*, wenn die Mengen $A \subseteq \Sigma^*$ und $R^{\mathcal{A}} \subseteq (\Sigma^*)^{\mathrm{ar}(R)}$ für $R \in \mathcal{R}$ entscheidbar sind.

Eine Struktur $\mathcal{A}$ ist *effektiv darstellbar*, wenn es eine isomorphe effektive Struktur gibt.

Beispiel.
- $\mathcal{N} = (\mathbb{N}, +, \cdot)$ ist effektiv darstellbar. Betrachte dazu die folgende Struktur $\mathcal{A}$:
  - $A = \{0\} \cup 1\{0,1\}^*$ mit $[a_n a_{n-1} \dots a_0]_{\mathcal{N}} = \sum_{0 \le i \le n} a_i 2^i$ für $a_0, \dots, a_n \in \{0,1\}$ und $a_n \dots a_0 \in A$
  - $\mathrm{add}^{\mathcal{A}} = \{(u,v,w) \in A^3 \mid [u]_{\mathcal{N}} + [v]_{\mathcal{N}} = [w]_{\mathcal{N}}\}$ - diese Menge ist entscheidbar
  - $\mathrm{mult}^{\mathcal{A}} = \{(u,v,w) \in A^3 \mid [u]_{\mathcal{N}} \cdot [v]_{\mathcal{N}} = [w]_{\mathcal{N}}\}$ - diese Menge ist entscheidbar

  Also ist $\mathcal{A} = (A, \mathrm{add}^{\mathcal{A}}, \mathrm{mult}^{\mathcal{A}})$ eine effektive Struktur. Wegen $\mathcal{A} \cong \mathcal{N}$ ist $\mathcal{N}$ effektiv darstellbar.

- $\mathcal{Q} = (\mathbb{Q}_{\geq 0}, +, \cdot)$ ist effektiv darstellbar. Betrachte dazu die folgende Struktur $\mathcal{B}$:
  - $B = \{(u,v) \in A \times A \mid v \neq 0, \mathrm{ggT}([u],[v]) = 1\}$ mit $[(u,v)]_{\mathcal{Q}} = \frac{[u]_{\mathcal{N}}}{[v]_{\mathcal{N}}}$ für $(u,v) \in B$
  - $\mathrm{add}^{\mathcal{B}} = \{((u_1,u_2),(v_1,v_2),(w_1,w_2)) \in B^3 \mid [(u_1,u_2)]_{\mathcal{Q}} + [(v_1,v_2)]_{\mathcal{Q}} = [(w_1,w_2)]_{\mathcal{Q}}\}$ - diese Menge ist entscheidbar
  - $\mathrm{add}^{\mathcal{B}} = \{((u_1,u_2),(v_1,v_2),(w_1,w_2)) \in B^3 \mid [(u_1,u_2)]_{\mathcal{Q}} \cdot [(v_1,v_2)]_{\mathcal{Q}} = [(w_1,w_2)]_{\mathcal{Q}}\}$ - diese Menge ist entscheidbar

  Also ist $\mathcal{B} = (B, \mathrm{add}^{\mathcal{B}}, \mathrm{mult}^{\mathcal{B}})$ eine effektive Struktur. Wegen $\mathcal{B} \cong \mathcal{Q}$ ist $\mathcal{Q}$ effektiv darstellbar.

- $\mathcal{S} = (\Sigma^*, H^{\mathcal{S}})$ mit $u \in R^{\mathcal{S}}$ gdw. Turingmaschine M mit Namen u hält bei leerer Eingabe.

  Da das Halteproblem nicht entscheidbar ist, ist $\mathcal{S}$ keine effektive Struktur.

  Betrachte die Struktur $\mathcal{B} = (B, R^{\mathcal{B}})$ mit $B = a^*$ und $R^{\mathcal{A}} = (aa)^*$.

  Offensichtlich ist $\mathcal{A}$ effektiv. Da unendlich viele Turingmaschinen bei der leeren Eingabe anhalten und ebensoviele nicht anhalten, gilt $\mathcal{S} \cong \mathcal{A}$, d.h. $\mathcal{S}$ ist effektiv darstellbar.

- $(\mathbb{R}, +)$ ist nicht effektiv darstellbar, da nur abzählbare Strukturen effektiv sein können.

Jede effektive Struktur ist endlich beschreibbar: man nehme z.B. Turingmaschinen, die die Grundmenge bzw. die Relationen entscheiden.

Also: D ist die Menge der Tupel von Turingmaschinen, die τ-Strukturen beschreiben. Für ein solches Tupel d ist $\mathcal{S}(d)$ die beschriebene effektive Struktur.

Da z.B. die Theorie von $\mathcal{N}$ nicht semi-entscheidbar (und damit nicht entscheidbar) ist, ist das uniforme Auswertungsproblem für $(D, \mathcal{S})$, d.h. für effektive Strukturen, unentscheidbar.

Ausweg: beschränke die Berechnungskraft der Turingmaschinen