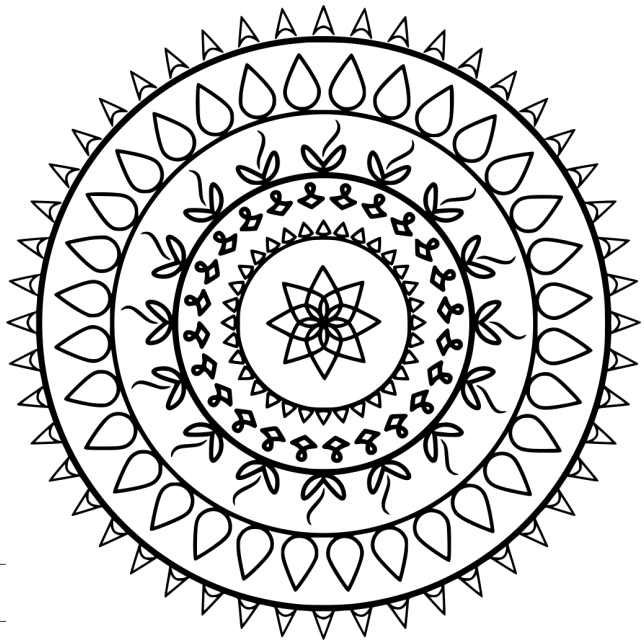
Prof. Dr. Thomas Schneider WS 2018/19

Logik

Fragebogen 15 vom 7. 1.

1. Wie kann man Relationsvariablen X in SO-Formeln verwenden? Man kann ...

- ☐ mit ihnen Terme bilden – z. B. $f(X)$
- ☐ mit ihnen Atome bilden – z. B. $X(f(x))$
- ☐ über sie quantifizieren – z. B. $\exists X.X(f(x))$
- ☐ mit drei davon unterschreiben – z. B. XXX



2. Wie wird eine Relationsvariable interpretiert?

- ◯ als ein Element des Universums
- ◯ als eine Menge von Elementen
- ◯ als eine Relation über dem Universum
- ◯ egal – in SO ist alles erlaubt

3. Welche Eigenschaften sind in SO ausdrückbar?

- Erreichbarkeit
- _____
- _____
- _____
- _____
- _____

4. Was bedeutet „monadisch“ in „MSO“?

_____

Welche der Formeln in den Beispielen auf den Folien (siehe Rückseite) sind in MSO?

_____

5. Sei $\mathfrak{A}$ eine Struktur mit Universum A und $|A| = n$, β eine Zuweisung in $\mathfrak{A}$ sowie φ eine **MSO-Formel** mit $|\varphi| = k$. Betrachte den Rekursionsbaum des Aufrufs $\texttt{ausw}(\mathfrak{A}, \beta, \varphi)$. Wie groß ist dessen ...

a) ... Tiefe?
- ◯ $\mathcal{O}(n)$
- ◯ $\mathcal{O}(k)$ (FO)
- ◯ $\mathcal{O}(2^k)$
- ◯ $\mathcal{O}(k^n)$

b) ... Verzweigungsgrad?
- ◯ $\mathcal{O}(n)$ (FO)
- ◯ $\mathcal{O}(k)$
- ◯ $\mathcal{O}(2^n)$
- ◯ $\mathcal{O}(2^k)$

c) ... Knotenzahl?
- ◯ $\mathcal{O}(nk)$
- ◯ $\mathcal{O}(n^k)$ (FO)
- ◯ $\mathcal{O}(k^n)$
- ◯ $\mathcal{O}(2^{nk})$

Zur Erinnerung ist die Antwort für den FO-Fall markiert.

Bitte wenden.

Beispiele

Erreichbarkeit:

$$\varphi(x,y) = \forall X \Big(\big(X(\underline{x}) \wedge \forall z \forall z' \, (X(z) \wedge E(z,z') \rightarrow X(z')) \big) \rightarrow X(\underline{y}) \Big)$$

„Jede Knotenmenge, die x enthält und unter Nachfolgern abgeschlossen ist, enthält auch y."

T4.1

EVEN $= \{ \mathfrak{A} \mid |A| \text{ geradzahlig} \}$

$$\text{Func}(R) = \forall x \forall y \forall y' \, (\, R(x,y) \wedge R(x,y') \rightarrow y = y' \,)$$
$$\text{Func}^{-}(R) = \forall x \forall y \forall y' \, (\, R(y,x) \wedge R(y',x) \rightarrow y = y' \,)$$

$$\varphi = \exists R \Big(\forall x \exists y \, R(x,y) \vee R(y,x) \wedge \forall x \, (\, \exists y \, R(x,y) \rightarrow \neg \exists y \, R(y,x) \,) \wedge \text{Func}(R) \wedge \text{Func}^{-}(R) \Big)$$

„A kann ohne Überlappungen mit $R \in \text{VAR}^2$ überdeckt werden."

Beispiele

Unendliche Strukturen:

$$\varphi_\infty = \exists R \Big(\exists x \exists y \, R(x,y) \wedge \forall x \forall y \, \big(R(x,y) \rightarrow \exists z \, R(y,z) \big) \wedge \forall x \neg R(x,x) \wedge \forall x \forall y \forall z \, \big((R(x,y) \wedge R(y,z)) \rightarrow R(x,z) \big) \Big)$$

„Man kann eine Teilmenge des Universums in einer irreflexiven, transitiven Ordnung ohne größtes Element anordnen."

Endliche Strukturen:

$\neg \varphi_\infty$

Auch Abzählbarkeit und Überabzählbarkeit sind definierbar.

Löwenheim-Skolem gilt also nicht (weder aufwärts noch abwärts).

Zur Erinnerung: der Algorithmus für FO

Function $\texttt{ausw}(\mathfrak{A}, \beta, \varphi)$**:**
- **input** : endl. Interpretation $(\mathfrak{A}, \beta)$, FO-Formel φ
- **output**: Wahrheitswert: 1 für $\mathfrak{A}, \beta \models \varphi$, 0 für $\mathfrak{A}, \beta \not\models \varphi$
- **switch** φ **do**
  - **case** $\varphi = (t = t')$: **if** $\beta(t) = \beta(t')$ **then return** 1 **else return** 0
  - **case** $\varphi = P(t_1, \ldots, t_k)$:
    - **if** $(\beta(t_1), \ldots, \beta(t_k)) \in P^{\mathfrak{A}}$ **then return** 1 **else return** 0
  - **case** $\varphi = \neg\psi$: **return** $1 - \texttt{ausw}(\mathfrak{A}, \beta, \psi)$
  - **case** $\varphi = \psi \wedge \vartheta$: **return** $\min\{\texttt{ausw}(\mathfrak{A}, \beta, \psi), \texttt{ausw}(\mathfrak{A}, \beta, \vartheta)\}$
  - **case** $\varphi = \psi \vee \vartheta$: **return** $\max\{\texttt{ausw}(\mathfrak{A}, \beta, \psi), \texttt{ausw}(\mathfrak{A}, \beta, \vartheta)\}$
  - **case** $\varphi = \exists x \, \psi$:
    - rufe $\texttt{ausw}\ (\mathfrak{A}, \beta[x/a], \psi)$ für alle $a \in A$
    - **if** *ein Ruf erfolgreich* **then return** 1 **else return** 0
  - **case** $\varphi = \forall x \, \psi$:
    - rufe $\texttt{ausw}\ (\mathfrak{A}, \beta[x/a], \psi)$ für alle $a \in A$
    - **if** *alle Rufe erfolgreich* **then return** 1 **else return** 0

Erweiterung des Algorithmus auf SO

Fälle für die SO-Teilformeln:

case $\varphi = X(t_1, \ldots, t_\ell)$*:* **do**
- **if** $(\beta(t_1), \ldots, \beta(t_\ell)) \in \beta(X)$ **then return** 1 **else return** 0

case $\varphi = \exists X \, \psi$*, wobei* X *eine* ℓ*-stellige Relationsvariable:* **do**
- rufe $\texttt{ausw}\ (\mathfrak{A}, \beta[X/B], \psi)$ für alle $B \subseteq A^\ell$
- **if** *ein Ruf erfolgreich* **then return** 1 **else return** 0

case $\varphi = \forall X \, \psi$*, wobei* X *eine* ℓ*-stellige Relationsvariable:* **do**
- rufe $\texttt{ausw}\ (\mathfrak{A}, \beta[X/B], \psi)$ für alle $B \subseteq A^\ell$
- **if** *alle Rufe erfolgreich* **then return** 1 **else return** 0

Platzkomplexität von Auswertung

Lemma 4.5

Der Algorithmus benötigt

1. polynomiell viel Platz, wenn die Eingabe eine MSO-Formel ist
2. exponentiell viel Platz im Allgemeinen

T4.2

Das Auswertungsproblem ist damit für MSO PSpace-vollständig, genauso wie für FO.

Warum beruht SQL dann auf FO und nicht auf der deutlich ausdrucksstärkeren MSO?

Zur Antwort betrachten wir die *Zeit*komplexität des Auswertungsproblems.

Zeitkomplexität von Auswertung

Lemma 4.6

Bei Eingabe einer Struktur $\mathfrak{A}$ der Größe n und einer Formel φ der Größe k benötigt der Algorithmus

1. Zeit $\mathcal{O}(n^k)$, wenn φ eine FO-Formel ist
2. Zeit $2^{\mathcal{O}(nk)}$, wenn φ eine MSO-Formel ist
3. Zeit $2^{\mathcal{O}(n^{2k})}$ im Allgemeinen.

T4.3

Diese Komplexitäten kann man (wahrscheinlich) nicht wesentlich verbessern.

Beachte: in Anwendungen ist …

- n meist sehr groß (Datenbank, zu verifizierendes System, …)
- k meist eher klein (Datenbankanfrage, zu verifizierende Eigenschaft, …)

Diese Beobachtungen führen zur Idee der *Datenkomplexität*.