Lehr- und Forschungsgebiet
Mathematische Grundlagen der Informatik
RWTH Aachen
Prof. Dr. E. Grädel, F. Reinhardt

SS 2014

10. Übung Algorithmische Modelltheorie II

Abgabe: bis Montag, 7. Juni um 15:00 Uhr am Lehrstuhl.

Aufgabe 1

Sei $\tau_{EDB} = \{Metro, Bus\}$ die Signatur einer Datenbank, welche die Verbindungsdaten für das Pariser Metro- und Busverbindungsnetz enthält. Die Datenbank enthalte genau dann den Eintrag $Metro(s_1, s_2)$ bzw. $Bus(s_1, s_2)$, falls man mit der Metro bzw. mit dem Bus von der Haltestelle s_1 ohne Zwischenhalt zur Haltestelle s_2 kommt. Schreiben Sie S-Datalog Programme für die folgenden Datenbankabfragen.

(a) Alle Paare (a, b), so dass man zwar mit der Metro von a nach b kommt, aber nicht mit dem Bus.

(b) Alle Paare (a, b), so dass es eine Busfahrt von a nach b gibt, ohne dass man an zwei direkt aufeinanderfolgenden Haltestellen c, d dieser Fahrt von c auch mit der Metro nach d fahren könnte.

(c) Alle Paare (a, b), so dass es eine Route von a nach b gibt, auf der man mindestens einmal mit dem Bus und mindestens einmal mit der Metro fahren kann, aber keine Route von a nach b gibt, auf der man nur mit dem Bus oder nur mit der Metro fahren kann.

(d) Alle Paare (a, b), so dass man von keiner Haltestelle c einer Busfahrt von a nach b aus mit der Metro eine der auf c folgenden Haltestellen dieser Busfahrt erreichen kann.

Aufgabe 2

Datalog mit Zählen, (Datalog+C), erweitert Datalog um Zählterme und zweisortige IDB - Prädikate (siehe hierzu auch AMT1-Skript Kapitel 6.1 und 6.2). Die zweisortigen IDB-Prädikate haben die Form $R(\overline{x}, \overline{\mu})$, wobei das Variablentupel $\overline{x}$ Werte der ersten Sorte annehmen kann, und $\overline{\mu}$ Werte der zweiten Sorte, also Zahlen. Für jedes Atom $R(x, \overline{y}, \overline{\mu})$ haben wir den Zählterm $\#_x[R(x, \overline{y}, \overline{\mu})]$. Datalog+C - Programme werden über zweisortigen Strukturen $\mathfrak{A}^* = \mathfrak{A} \cup (\{0, 1, \dots, |A|\}, \leq, 0, e, +1)$ ausgewertet. Ein Term der zweiten Sorte wird *arithmetischer Term* genannt. Ein arithmetischer Term ist entweder 0,e, ein Zählterm t oder $t + 1$ für einen arithmetischen Term t. Ein Datalog+C - Programm Π ist eine endliche Menge von Klauseln der Form $H \leftarrow B_1 \wedge \ldots \wedge B_m$, wobei H eine atomare Formel $R(\overline{x}, \overline{\mu})$ ist und $B_1, \dots, B_m$ entweder atomare Formeln der Art $R(\overline{x}, \overline{\mu})$ oder Gleichungen $t_1 = t_2$ von Termen der ersten oder zweiten Sorte sind. Im Unterschied zu Datalog wird die Semantik jedoch nicht über den kleinsten Fixpunkt, sondern über den *inflationären* Fixpunkt definiert:
Seien $Q_1, \dots, Q_n$ die in Π vorkommenden Kopfprädikate. Jeder Körper $B_1 \wedge \ldots \wedge B_m$ einer Regel $R(\overline{x}, \overline{\mu}) \leftarrow B_1 \wedge \ldots B_m$ hat die Form $\beta(\overline{x}, \overline{\mu}, \overline{y}, \overline{\nu})$. Der inflationäre Updateoperator F ist definiert durch $F(Q_1, \dots, Q_n) = (Q_1' \cup Q_1, \dots, Q_n' \cup Q_n)$ mit

$$Q_i' := \{(\overline{a}, \overline{n}) : \text{es gibt eine Klausel } Q_i(\overline{x}, \overline{\mu}) \leftarrow \beta(\overline{x}, \overline{\mu}, \overline{y}, \overline{\nu}) \text{ in } \Pi, \text{ so dass } \mathfrak{A}^* \models \exists \overline{y} \exists \overline{\nu} \beta(\overline{a}, \overline{n}, \overline{y}, \overline{\nu})\}$$

Sei $(Q_1^\infty, \dots, Q_n^\infty)$ der induktive Fixpunkt von F, dann ist die Auswertung von Π in $\mathfrak{A}$ definiert als $\Pi(\mathfrak{A}) := (\mathfrak{A}, Q_1^\infty, \dots, Q_n^\infty)$. Außerdem setzen wir $\Pi^{Q_i}(\mathfrak{A}) := Q_i^\infty$.

(a) Warum würde es hier keinen Sinn machen die Semantik wie bei Datalog über den kleinsten Fixpunkt zu definieren?

(b) Gegeben sei das folgende (Datalog+C)-Programm Π_x:

$$\begin{array}{rcl} Ax\mu & \leftarrow & \#_y[Exy] = \mu \\ B\mu\nu & \leftarrow & \#_x[Ax\mu] = \nu \end{array}$$

(i) Was wird durch die Funktion

$$f(G) := \sum_{(\mu,\nu)\in\Pi_x^B(G)} \mu \cdot \nu$$

berechnet für $G \in \mathrm{Fin}(\{E\})$?

(ii) Erweitern Sie Π_x zu einem (Datalog+C)-Programm das $f(G)$ berechnet.

(c) Schreiben Sie ein (Datalog+C)-Programm, welches die Gewinnregion $W_0 = [\mathrm{ifp} Wx.Tx \vee \exists y \forall z(Exy \wedge (Eyz \rightarrow Wz))]$ von Spieler 0 in Erreichbarkeitsspielen $\mathcal{G} = (V, E, T)$ definiert, wobei (V, E) ein gerichteter Graph ist, $T \subseteq V$ die Menge der Positionen, die Spieler 0 erreichen muß und Spieler 0 und 1 jeweils abwechselnd entlang der E-Kanten ziehen. *Hinweis:* Definieren Sie ein Hilfsprädikat $Wx\mu$ mit der Bedeutung, dass Spieler 0 von x aus in μ Zügen gewinnen kann.