BLATT 10

DANIEL SCHMIDT & PAMELA FLEISCHMANN

Aufgabe 1. Um zu zeigen, dass sich für jede TRC-Anfrage zu einem DB-Schema σ eine äquivalente Anfrage des DRC zu σ finden lässt definieren wir uns einen Algorithmus, welcher TRC-Anfragen zu DRC-Anfragen umformt. Sei also eine allgemeine TRC-Anfrage $(x)/\theta(x)$, so lässt sich der Algorithmus wie folgt beschreiben:
Sei zunächst für jede Variable k in x mit dem Typen $\tau_1, \cdots, \tau_n$ neue Variablen $k_1, \cdots, k_n$ mit den entsprechenden Typen eingeführt. Nun gilt es die Variablen zu ersetzen um DRC-Anfragen zu erhalten, dies geschieht nach den folgenden Regeln:
Wenn $RT_i(k)$ gegeben ist, so muss dies durch $RT_i(k_1, \cdots, k_n)$ ersetzt werden. Falls $k.B_j\theta c(c\theta k.B_j)$ gegeben ist, so muss dies durch $k_j\theta c(c\theta k_j)$ ersetzt werden. Wenn $k.B_j\theta z.C_h$ gegeben ist, so muss dies ersetzt werden durch $k_j\theta z_h$.
Falls $\exists k$ gegeben ist, so muss dies falls k gebunden ist durch $(\exists k_1), \cdots, (\exists k_n)$ ersetzt werden. Falls k ungebunden ist, so ist dies nicht nötig, da das Ergbnis ohnehin nicht weiterverwendet wird. Analog lässt sich $\forall k$ umformen. Zuguterletzt muss die Zielfunktion noch angepasst werden, entprechend also $(x)/\cdots$ zu $(x_1, \cdots, x_n)/\cdots$ umgeformt werden.

Aufgabe 2. Die Idee für rank(e) ist in einer while-Schleife eine Variable hochzuzählen während von der Originalrelation die Stellen sukzessive abgeschnitten werden:

$$y = E \downarrow\downarrow\uparrow \text{ // Darstellung von 1}$$
$$x = e$$
$$\text{while } x \text{ do } (x = x \downarrow; y = y \uparrow)$$

Für die Projektion auf die i-te Komponente einer Relation r_i sei e die Darstellung von rank$(r_i) - i$ und e'' die Darstellud von $i - 1$.

$$x = e;$$
$$y = r_i;$$
$$\text{while } x \neq \emptyset \text{ do } (y = y' \downarrow; x = x \downarrow)$$
$$x = e'';$$
$$\text{while } x \neq \emptyset \text{ do } (y = y \circlearrowleft; y = y \downarrow; x = x \downarrow;)$$

Die Idee ist die hinteren Stellen abzuschneiden und dann immer zu permutieren und abzuschneiden, bis nur noch die richtige Stelle übrig ist.

Definiere $\uparrow_d$: $\mathcal{R}_s \to \mathcal{R}_s; e \mapsto \{(d_1, \ldots, d_s, d)\}|(d_1, \ldots, d_s) \in e\}$ (wir haben diesen Operator leider nicht modelliert bekommen). Das kartesische Produkt ist gegeben durch

$$\begin{aligned} &x = \mathrm{rank}(e); \\ &\text{while } x \neq \emptyset \text{ do } (e = e \uparrow_{\mathrm{pr}_x(e_2)}; x = x \downarrow) \end{aligned}$$

Ist r_2 eine Relation, die auf die zu projizierenden Indizes vorhält, so ergibt sich die allgemeine Projektion durch

$y = r_2$

$x = \mathrm{rank}(r_1)$

while $y \neq \emptyset$ do (while $x \neq \emptyset$ do (if $(x, y) \in E$ then $e_1 = e_1 \times \mathrm{pr}_y(e))))$

Das Anfügen von Elementen von links ist definiert durch

$$E \downarrow \times e;$$