

Wiederholung: Tupel- und Bereichskalkül

Relationenkalkül

- was wird berechnet = deklarative Sprache (\leftrightarrow Relationalen Algebra = prozedurale Sprache)
- Kalkül = logischer Formalismus zur Ableitung von Ergebnissen
- Kalkül besteht immer aus Syntax (Wie sind Ausdrücke aufgebaut?) und Semantik (Was bedeuten Ausdrücke?).
- zwei Ansätze: Tupelkalkül und Bereichskalkül
- Tupelkalkül: Variablen werden an Tupel einer Relation gebunden
- Bereichskalkül: Variablen werden an Wertebereiche von Attributen gebunden

Syntax Tupelkalkül:

- Tupelvariablen t bzgl. Schema $S = \text{Schema}(t)$
- Atome: $R(t)$, $t.A\theta s.B$, $t.A\theta c$ ($\theta \in \{<, \leq, >, \geq, =, \neq\}$)
- Induktive Definition von Formeln:
 - Jedes Atom ist Formel
 - φ_1 und φ_2 Formel $\Rightarrow \neg\varphi_1, \varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2$ auch Formel
 - φ Formel und t frei in $\varphi \Rightarrow \exists t\varphi$ und $\forall t\varphi$ auch Formel
- Ausdruck (Alternative 1): $\{t|\varphi(t)\}$, wobei t einzige freie Tupelvariable in φ ist. Das Schema von t muss explizit angegeben werden.
- Ausdruck (Alternative 2): $\{[t_1.A_1, \dots, t_n.A_n]|\varphi(t_1, \dots, t_n)\}$, wobei t_1, \dots, t_n die einzigen freien Tupelvariablen in φ sind. Bei Ausdrücken dieser Form ist das Schema der Ergebnisrelation implizit und muss beim Anfragen nicht explizit angegeben werden.

Syntax Bereichskalkül:

- Bereichsvariablen $x_1 : D_1, \dots, x_k : D_k$ für einzelne Attribute
- Atome: $R(x_1, \dots, x_k)$, $x\theta y$, ($\theta \in \{<, \leq, >, \geq, =, \neq\}$, x, y Bereichsvariablen oder Konstanten)
- Induktive Definition von Formeln: analog zum Tupelkalkül
- Ausdruck: $\{x_1, \dots, x_k|\varphi(x_1, \dots, x_k)\}$, wobei x_1, \dots, x_k die einzig freien Variablen in φ ist