Relationaler Tupelkalkül

Datenbanken I (Systemorientierte Informatik IV)
Sommersemester 2007

Gunar Fiedler (fiedler@is.informatik.uni-kiel.de)
Institut für Informatik
Arbeitsgruppe „Technologie der Informationssysteme“
Christian-Albrechts-Universität zu Kiel

1 Syntax des relationalen Tupelkalküls

Der relationale Tupelkalkül (engl. *tuple relational calculus*, TRC) ist eine Anfragesprache für relationale Datenbanken, die Ausdrücke der Prädikatenlogik erster Stufe benutzt, um das gewünschte Ergebnis einer Anfrage zu beschreiben. Eine mögliche Anfrage an das Vorlesungs- und Studentenbeispiel aus dem Übungsskript zur relationalen Algebra könnte z.B. sein:

Gib die Namen und Matrikelnummern der Studenten, die eine Vorlesung hören, die von einem Dozenten namens Meyer gehalten wird.

Wer die relationale Algebra benutzt, muss eine Operationsfolge angeben, die das Ergebnis aus den Relationen der Datenbank konstruiert. Die Anfrage ist aber eigentlich anders formuliert: die gesuchten Studenten werden durch die sie charakterisierenden Eigenschaften beschrieben, nämlich die Eigenschaft, dass sie bestimmte Vorlesungen hören. Diese Vorlesungen sind dadurch charakterisiert, dass sie von einem bestimmten Dozenten (mit dem Namen *Meyer*) gehalten werden. Der TRC versucht, diese Charakterisierung auf einem formalen Wege durchzuführen.

Sei ein Datenbankschema $\mathcal{D}$ gegeben. Wir definieren eine Menge von Variablen. In den folgenden Ausführungen werden wir Variablen mit kleinen lateinischen Buchstaben schreiben. Die Variablen stellen Platzhalter für Tupel dar. Jeder Variablen ist ein Typ zugeordnet: seien $A_1, ..., A_n$ die Attribute über denen die Tupel, die der Variable später zugeordnet werden sollen, definiert sind. Dann ist die Menge $\{A_1, ..., A_n\}$ der Typ der Variable. Den Typ einer Variablen, z.B. der Variablen x, bezeichnen wir mit $type(x)$.

In unserem Beispiel suchen wir z.B. die Namen und die Matrikelnummern der Studenten. Deshalb können wir z.B. einen Typ $type(x) = \{MatrikelNr, Name\}$ einführen. Die Variable x ist jetzt Platzhalter für Tupel über MatrikelNr und Name.

Eine Anfrage des TRC ist eine prädikatenlogische Formel (erster Stufe) der folgenden Form:

- Wenn R ein Relationenschema in $\mathcal{D}$ und x eine Variable mit $type(x) = attr(R)$ ist, dann ist $R(x)$ eine Formel im Sinne des TRC mit x als freier Variablen.

  Beispiele:
  $STUDENT(y)$ mit $type(y) = \{MatrikelNr, Name, Wohnort\}$
  $RAUM(z)$ mit $type(z) = \{RaumNr, Bezeichnung\}$

- Wenn x und y Variablen, $A \in type(x)$ und $B \in type(y)$ Attribute und $\odot$ ein Prädikat über den Typen von A und B ist, dann ist $x.A \odot y.B$ eine Formel im Sinne des TRC mit den freien Variablen x und y. Analog wird der Vergleich mit Konstanten definiert.

  Beispiele:
  $y.Wohnort =' Kiel'$
  $x.Name = y.Name$

- Wenn φ und ψ Formeln im Sinne des TRC sind, dann sind auch die aussagenlogischen Verknüpfungen von φ und ψ Formeln im Sinne des TRC. Die freien Variablen entsprechen denen der Formeln φ und ψ.

  Beispiele:
  $y.Wohnort =' Kiel' \wedge x.Name = y.Name$
  $x.Name =' Petersen' \implies y.Name =' Schmidt'$

- Wenn φ eine Formel im Sinne des TRC und x eine freie Variable in φ ist, dann sind auch

  $$(\forall x)(\varphi)$$

  $$(\exists x)(\varphi)$$

  Formeln des TRC. Die freien Variablen dieser Formeln entsprechen den freien Variablen von φ ohne x.

  Beispiel:
  $(\forall v)(VORLESUNG(v) \implies (\exists f)(FINDETSTATT(f) \wedge f.VorlesungsNr = v.VorlesungsNr))$

2 Erste Semantikdefinition des TRC

Im Übungsskript „Grundlagen der Logik“ wurde der Begriff des Modells für eine Formel φ der Prädikatenlogik erster Stufe eingeführt. Wir gehen in dieser Diskussion von gegebenen Wertebereichen und einer festen (und trivialen) Belegung der Konstantensymbole aus, deshalb sind wir an Paaren I, ϱ interessiert, die unsere Formel φ wahr werden lassen.
Die Interpretation I ordnet den Prädikaten eine Ausprägung zu, also die Menge von Tupeln, für die das Prädikat zu wahr ausgewertet wird. Im TRC betrachten wir 2 verschiedene Arten von Prädikaten:

1. $R(x)$ für ein Relationenschema R,
2. $\odot$, definiert auf Ebene der Datentypen.

Es liegt nahe, die Ausprägung der Prädikate der ersten Art an die Relationen unserer Datenbank zu binden. Für die Prädikate der zweiten Art benutzen wir die Definitionen der Prädikate in den Datentypen. Da wir diese Definition als konstant ansehen, werden wir in der weiteren Diskussion nur noch die Prädikate der ersten Art betrachten.

Als Interpretation benutzen wir demnach einen („den aktuellen“) Zustand $\sigma(\mathcal{D})$ zu unserem Datenbankschema $\mathcal{D}$. Als Ergebnis einer TRC-Anfrage bezeichnen wir die Menge der Variablenbelegungen ϱ, so dass

$$\sigma(\mathcal{D}), \varrho \models \varphi$$

Da i.d.R. nicht die Belegungen aller Variablen interessant sind, wird pro Antwort auf die Anfrage φ nur die Belegung der freien Variablen in φ angegeben. Um auch dies weiter einzuschränken, kann man vor die Formel φ eine *Projektionsliste* mit den gewünschten Attributen schreiben. Seien $x_1, ..., x_k$ die freien Variablen in φ und $A_1, ..., A_l$ Attribute in den Typen von $x_1, ..., x_k$, dann kann man schreiben:

$$x_1.A_1, ..., x_i.A_j \ | \ \varphi$$

Diese Schreibweise konstruiert Tupel über $\{A_1, ..., A_j\}$ als Ergebnis der Anfrage.

Beispiel: *„Gib die Namen und Matrikelnummern aller Kieler Studenten zusammen mit den Veranstaltungsnummern, so dass dieser Student diese Veranstaltung mindestens in der ersten Wiederholung hört.“*

$s.Name, s.MatrikelNr, h.VorlesungsNr \ | \ STUDENT(s) \wedge HOERT(h) \wedge$
$h.MatrikelNr = s.MatrikelNr \wedge h.Wiederholung \geq 1$

Wenn die Menge der freien Variablen der Formel φ leer ist (d.h. alle Variablen in φ sind an Quantoren gebunden), dann ist die Antwort entweder *ja* (d.h. $\sigma(\mathcal{D}) \models \varphi$) oder *nein* (d.h. $\sigma(\mathcal{D}) \not\models \varphi$).

Selbststudium Führen Sie sich anhand der folgenden Semantikdefinition vor Augen, dass die Variablenbelegung ϱ in diesem Falle für die Erfüllbarkeit keine Rolle spielt. Führen Sie sich weiterhin vor Augen, dass die Projektion auf die gewünschten Attribute nur „syntaktischer Zucker" ist.

Nachdem wir festgelegt haben, welche Attribute als Antwort auf die Anfrage auszugeben sind, können wir eine erste Semantikdefinition des TRC angeben. Sei ϱ eine Variablenbelegung:

1. Sei R ein Relationenschema und x eine Variable, dann gilt $\sigma(\mathcal{D}), \varrho \models R(x)$ gdw. $\varrho(x) \in \sigma(R)$, d.h. das Tupel, mit dem x belegt ist, ist in der zu R gehörenden Relation enthalten.

2. $\sigma(\mathcal{D}), \varrho \models x.A \odot y.B$ gdw. $\odot(\varrho(x)(A), \varrho(y)(B))$, d.h. wir betrachten die Tupel, mit denen die Variablen x und y belegt sind. Wenn das Prädikat $\odot$ auf die Werte der Attribute A und B dieser Tupel angewendet wird, muss es *wahr* ergeben, damit der Datenbankzustand und die Variablenbelegung ein Modell bilden. Vergleiche mit Konstanten werden analog behandelt.

3. $\sigma(\mathcal{D}), \varrho \models \varphi \wedge \psi$ für zwei Formeln φ und ψ gdw. $\sigma(\mathcal{D}), \varrho \models \varphi$ und $\sigma(\mathcal{D}), \varrho \models \psi$. Die anderen aussagenlogischen Verknüpfungen werden analog behandelt.

4. $\sigma(\mathcal{D}), \varrho \models (\exists x)(\varphi)$, gdw. es eine Variablenbelegung ϱ' gibt, so dass sich ϱ' von ϱ höchstens in der Belegung von x unterscheidet und $\sigma(\mathcal{D}), \varrho' \models \varphi$ gilt.

5. $\sigma(\mathcal{D}), \varrho \models (\forall x)(\varphi)$, gdw. für alle Variablenbelegungen ϱ', die sich von ϱ höchstens in der Belegung von x unterscheiden, gilt, dass $\sigma(\mathcal{D}), \varrho' \models \varphi$.

3 Auswertung einer Anfrage

Es sei der folgende Zustand gegeben (es sind nur die relevanten Relationen angegeben):

STUDENT

MatrikelNr	Name	Wohnort
190245	Max Müller	Kiel
327641	Tina Petersen	Flensburg
612491	Tobias Schulze	Kiel
762198	Uwe Schmidt	Rendsburg

VORLESUNG

VorlesungsNr	Bezeichnung
080104	SysInf IV
080016	Info IV
080127	Info II
080176	SysInf II

HOERT

MatrikelNr	VorlesungsNr	Wiederholung
190245	080104	0
190245	080016	0
327641	080127	0
327641	080176	0
612491	080104	0
612491	080016	0
612491	080127	1
612491	080176	1
762198	080104	2

Betrachten wir obige Anfrage

$s.Name, s.MatrikelNr, h.VorlesungsNr \mid STUDENT(s) \wedge HOERT(h) \wedge$
$h.MatrikelNr = s.MatrikelNr \wedge h.Wiederholung \geq 1$

Wir müssen nun für alle beliebigen Variablenbelegungen prüfen, ob der Datenbankzustand und die Belegung ein Modell der Anfrage bilden. Die möglichen Variablenbelegungen ergeben sich aus den Wertebereichen der Attribute:

$$\mathit{type}(s) = \{MatrikelNr, Name, Wohnort\}$$

$$\mathit{type}(h) = \{MatrikelNr, VorlesungsNr, Wiederholung\}$$

Die Belegungen der Variablen s bestehen demnach aus allen Kombinationen möglicher Matrikelnummern, Namen und Wohnorte. Wenn wir davon ausgehen, dass alle drei Attribute über dem Datentyp „Zeichenkette“ definiert sind, müssen wir alle Kombinationen aus drei beliebigen und beliebig langen Zeichenketten betrachten. Damit ergeben sich z.B. folgende Belegungen:

$\rho(s)$	**MatrikelNr**	**Name**	**Wohnort**
	000000	a	a
	000001	a	a
	000002	a	a
	...	...	...
	190245	Max Müller	Kiel
	190245	Max Müller	Lübeck
	190245	Max Müller	Flensburg
	...	...	...
	327641	Tina Petersen	Flensburg
	327641	Max Müller	Flensburg
	327641	Max Müller	aW349(3!
	...	...	...
	zsafhgsadz	HTjhsbva,!	OiEwiuq43D
	...	...	...

Es ist leicht ersichtlich, dass es unendlich viele Kombinationen dreier (beliebig langer) Zeichenketten gibt. Eine analoge Diskussion kann für die Belegung der Variablen h erfolgen. Durch „scharfes Hinsehen“[1] stellen wir aber fest, dass für alle Belegungen, die s kein Tupel aus der Relation $STUDENT$ zuweisen, die gesamte Formel niemals erfüllt werden kann. Analog wird die Formel nur erfüllt, wenn der Variablen h ein Tupel aus der Relation $HOERT$ zugeordnet wird. Die letzten beiden Bedingungen schränken dies weiter ein: es werden alle Kombinationen aus s und h verworfen, die die beiden Vergleiche nicht erfüllen. Es bleiben die folgenden Belegungen übrig:

	s			h		
	MatrikelNr	**Name**	**Wohnort**	**MatrikelNr**	**VorlesungsNr**	**Wiederholung**
ϱ_1	612491	Tobias Schulze	Kiel	612491	080127	1
	MatrikelNr	**Name**	**Wohnort**	**MatrikelNr**	**VorlesungsNr**	**Wiederholung**
ϱ_2	612491	Tobias Schulze	Kiel	612491	080176	1
	MatrikelNr	**Name**	**Wohnort**	**MatrikelNr**	**VorlesungsNr**	**Wiederholung**
ϱ_3	762198	Uwe Schmidt	Rendsburg	762198	080104	2

Diese projizieren wir auf die gegebenen Attribute und erhalten unsere Antwort:

Name	**MatrikelNr**	**VorlesungsNr**
Tobias Schulze	612491	080127
Tobias Schulze	612491	080176
Uwe Schmidt	762198	080104

Betrachten wir weiterhin folgende Anfrage ($type(x) = attr(VORLESUNG)$):

$$\neg VORLESUNG(x)$$

[1] Systematischere Verfahren lernen Sie in der Veranstaltung „Datenbanktheorie“ oder in der Logikprogrammierung kennen.

Auch hier müssen wir alle Tupel betrachten, die sich aus zwei beliebig langen Zeichenketten (VorlesungsNr und Bezeichnung) bilden lassen. Durch „scharfes Hinsehen“ stellen wir fest, dass sich die vier im Zustand der Datenbank aufgeschriebenen Tupel nicht für die Antwort qualifizieren. Jede andere Kombination aus einer Zeichenkette für die Vorlesungsnummer und die Bezeichnung — auch alle unsinnigen Kombinationen — sind Teil der Antwortmenge. Da es unendlich viele solche Kombinationen gibt, dauert es unendlich lange, bis die Antwort berechnet ist. Mit anderen Worten ausgedrückt: unser Algorithmus terminiert nicht, das Ergebnis der Anfrage ist nicht berechenbar. Ausdrücke dieser Form nennen wir in Zukunft „unsichere Ausdrücke“ (da man sich, vereinfacht gesprochen, nicht sicher sein kann, ein Ergebnis zu erhalten.) Ausdrücke, die stets eine endliche Menge von Ergebnistupeln liefern, nennen wir „sichere Ausdrücke“. „Stets“ bedeutet in diesem Zusammenhang, dass die Endlichkeit des Ergebnisses für alle gültigen Datenbankzustände garantiert ist.

4 Sicherheit und Wertebereichsunabhängigkeit

Die Menge der sicheren TRC-Ausdrücke ist unentscheidbar, d.h. es gibt keinen Algorithmus, der für einen beliebigen gegebenen TRC-Ausdruck bestimmt, ob dieser sicher oder unsicher ist. Aus diesem Grunde betrachtet man eine weitere Eigenschaft: die Wertebereichsabhängigkeit bzw. -unabhängigkeit eines TRC-Ausdrucks.
Das Problem der unsicheren TRC-Ausdrücke liegt in der Tatsache begründet, dass über einem unendlichen Wertebereich gearbeitet wird und sich prinzipiell jedes Tupel dieser unendlichen Menge für das Ergebnis qualifizieren kann. Unsinnige Wertekombinationen in Tupeln der Antwort (z.B. die Vorlesungsbezeichnung $lhiGf!r5W$) interessieren uns aber eigentlich nicht. Deshalb können wir untersuchen, was passiert, wenn wir nicht mehr alle beliebigen Belegungen unserer Variablen betrachten, sondern nur noch die Belegungen, deren (Attribut-)Werte auf irgendeine Art und Weise im Datenbankzustand oder in der Anfrage vorkommen, die wir also als „sinnvoll“ erachten. Wenn eine Anfrage jetzt eine andere Antwort liefert als im vorher diskutierten Fall, dann hängt das Ergebnis nicht nur vom Datenbankzustand ab, sondern von den Werten der Wertebereiche der Attribute. In diesem Fall sprechen wir von einem *wertebereichsabhängigen* TRC-Ausdruck. Wenn sich das Ergebnis nicht verändert, dann ist der TRC-Ausdruck *wertebereichsunabhängig*. Die erste Anfrage im Abschnitt 3 ist eine wertebereichsunabhängige Anfrage: Egal welche Wertebereiche wir betrachten[2], die Anfrage liefert bzgl. eines fest gewählten Datenbankzustands immer die gleiche Antwort. Die zweite Anfrage ($\neg VORLESUNG(x)$) ist wertebereichsabhängig, denn z.B. je nach den möglichen Vorlesungsbezeichnungen entsteht jedesmal eine andere Antwort.

Es gilt: Jeder wertebereichsunabhängige TRC-Ausdruck ist sicher. Die Umkehrung muss nicht zwangsläufig gelten[3]. Leider ist die Menge der wertebereichsunabhängigen

[2] Die Werte des Datenbankzustands müssen natürlich in den Wertebereichen enthalten sein, sonst widerspricht dies unseren Definitionen des relationalen Modells!

[3] Wenn man z.B. nur endliche Wertebereiche betrachtet, dann ist jeder Ausdruck sicher. Er kann

Ausdrücke immer noch unentscheidbar. Um eine Entscheidbarkeit zu erzwingen, werden die vom System zugelassenen Anfragen syntaktisch eingeschränkt, d.h. wir erlauben nicht mehr beliebige Anfragen des TRC. Dies führt uns zur Definition der *erlaubten Ausdrücke*.

Erlaubter TRC-Ausdruck In einem erlaubten TRC-Ausdruck wird jede Variable an den Datenbankzustand „gebunden". Um dies festzustellen, prüfen wir für jedes Attribut A jeder Variablen x, ob das Paar (x, A) in einem Ausdruck φ positiv oder negativ beschränkt ist:

1. (x, A) ist positiv beschränkt in $R(x)$
2. Sei c eine Konstante. Dann ist (x, A) positiv beschränkt in $x.A = c$ und $c = x.A$. Wir setzen die übliche Definition des Gleichheitsprädikats voraus.
3. (x, A) ist positiv beschränkt in $x.A = y.B$ bzw. $y.B = x.A$, wenn der Term Teil einer Konjunktion ist ($F_1 \wedge ... \wedge x.A = y.B \wedge ... \wedge F_n$), in der $y.B$ positiv beschränkt ist.
4. (x, A) ist positiv beschränkt in $\neg\varphi$, wenn (x, A) negativ beschränkt ist in φ.
5. (x, A) ist positiv beschränkt in $\varphi \wedge \psi$, falls (x, A) positiv beschränkt ist in φ oder in ψ.
6. (x, A) ist positiv beschränkt in $\varphi \vee \psi$, falls (x, A) positiv beschränkt ist in φ und in ψ.
7. (x, A) ist positiv beschränkt in $\varphi \implies \psi$, falls (x, A) negativ beschränkt ist in φ und positiv beschränkt in ψ.
8. (x, A) ist positiv beschränkt in $(\exists y)(\varphi)$ oder in $(\forall y)(\varphi)$, falls (x, A) positiv beschränkt ist in φ.
9. (x, A) ist negativ beschränkt in $\neg\varphi$, falls (x, A) positiv beschränkt ist in φ.
10. (x, A) ist negativ beschränkt in $\varphi \wedge \psi$, falls (x, A) negativ beschränkt ist in φ und in ψ.
11. (x, A) ist negativ beschränkt in $\varphi \vee \psi$, falls (x, A) negativ beschränkt ist in φ oder in ψ.
12. (x, A) ist negativ beschränkt in $\varphi \implies \psi$, falls (x, A) positiv beschränkt ist in φ oder negativ beschränkt in ψ.
13. (x, A) ist negativ beschränkt in $(\exists y)(\varphi)$ oder in $(\forall y)(\varphi)$, falls (x, A) negativ beschränkt ist in φ.

aber sehr wohl wertebereichsabhängig sein.

Eine Variable x ist positiv (negativ) beschränkt in einem Ausdruck φ, falls die positive (negative) Beschränkung für alle Attribute dieser Variablen gilt.

Ein TRC-Ausdruck φ heißt erlaubt, wenn folgendes gilt:

1. Jede freie Variable in φ ist positiv beschränkt.
2. Für jeden Teilausdruck $(\exists x)(\psi)$ ist die Variable x positiv beschränkt in ψ.
3. Für jeden Teilausdruck $(\forall x)(\psi)$ ist die Variable x negativ beschränkt in ψ.

Jeder erlaubte Ausdruck ist wertebereichsunabhängig und demnach sicher. Damit lässt sich ein Algorithmus angeben, der überprüft, ob ein gegebener Ausdruck wertebereichsunabhängig ist:

1. Ist der Ausdruck nach obiger Definition erlaubt? Wenn ja: der Ausdruck ist wertebereichsunabhängig.
2. Sonst: Lässt sich ein Gegenbeispiel angeben? Man konstruiert sich ein Universum, dass genau die Konstanten aus dem Datenbankzustand und der Anfrage enthält und führt die Anfrage aus. Anschließend fügt man weitere Werte zum Universum hinzu und führt die Anfrage erneut aus. Wenn sich unterschiedliche Ergebnisse erzeugen lassen, ist die Anfrage wertebereichsabhängig.
3. Falls kein Gegenbeispiel gefunden wurde: lässt sich die Formel umstellen (De Morgansche Gesetze, Quantorumformung, etc.)? Falls ja, gehe zu 1.
4. Falls nicht: keine Entscheidung möglich.

5 Bereichsbeschränkte TRC-Ausdrücke

Es gibt eine weitere Möglichkeit, nur sichere und wertebereichsunabhängige Ausdrücke zu formulieren: Variablen müssen strikt an Relationenschemata gebunden werden:

- freie Variablen stehen in einem Term der Form $R(x)$
- Quantifizierte Variablen werden nur mit Tupeln aus einer Relation belegt:

$$(\exists x \in R)(\varphi)$$

$$(\forall x \in R)(\varphi)$$

TRC-Ausdrücke dieser Form heißen beschränkte Ausdrücke bzw. R-beschränkte Ausdrücke. Bei R-beschränkten Ausdrücken ist eine Projektion auf Attribute in der Zielliste nicht länger syntaktischer Zucker. Beispiele sind:

„Welche Studenten hören welche Vorlesung?“

$s.MatrikelNr, h.VorlesungsNr \mid STUDENT(s) \wedge HOERT(h) \wedge h.MatrikelNr = s.MatrikelNr$

„Welche Studenten hören alle Vorlesungen?“

$s.Name \mid$
$STUDENT(s) \wedge (\forall v \in VORLESUNG)($
$(\exists h \in HOERT)(h.MatrikelNr = s.MatrikelNr \wedge h.VorlesungsNr = v.VorlesungsNr)$
$)$

Integritätsbedingung: „Jeder Student hört mindestens zwei verschiedene Vorlesungen.“

$(\forall s \in STUDENT)($
$(\exists h1 \in HOERT)($
$(\exists h2 \in HOERT)($
$h1.MatrikelNr = s.MatrikelNr \wedge h2.MatrikelNr = s.MatrikelNr \wedge$
$\neg(h1.VorlesungsNr = h2.VorlesungsNr)$
$)))$

Gib die Namen und Matrikelnummern der Studenten, die eine Vorlesung hören, die von einem Dozenten namens Meyer gehalten wird.

$s.Name, s.MatrikelNr \mid$
$STUDENT(s) \wedge (\exists h \in HOERT)(h.MatrikelNr = s.MatrikelNr \wedge$
$(\exists d \in DOZENT)(d.VorlesungsNr = h.VorlesungsNr \wedge$
$(\exists m \in MITARBEITER)($
$m.BearbeiterNr = d.BearbeiterNr \wedge m.PersonalNr = d.PersonalNr \wedge$
$m.Name =' Meyer'$
$)))$