Datenbanktheorie
Sommersemester 2012
Prof. Dr. W. May

1. Übungsblatt: Kalkül I

Besprechung voraussichtlich am 10./11.5.2012

Aufgabe 1 (FOL: Beweise) a) Beweisen Sie die Äquivalenz der Ausdrücke $\forall x : F(x)$ und $\nexists y : \neg F(y)$ indem Sie die Definition der Semantik der Formeln benutzen.

b) Zeigen Sie: Wenn in einer Anwendung ein kausaler Zusammenhang zwischen der Gültigkeit einer Formel φ und einer Formel ψ besteht (d.h., wenn in einem beliebigen Zustand φ erfüllt ist, auch immer ψ in diesem Zustand φ erfüllt ist), dass dann dass dann auch die *materiale Implikation* $\varphi \rightarrow \psi$ in jedem der Spezifikation entsprechenden Zustand erfüllt ist.

c) Geben Sie ein Beispiel dafür, dass die Formel $\varphi \rightarrow \psi$ in einem Zustand erfüllt sein kann, ohne dass ein kausaler Zusammenhang besteht.

d) Geben Sie Beispiele an, wie die materiale Implikation $\varphi \rightarrow \psi$ sinnvoll in einer Formel vorkommt.

---

a) Bei der sehr expliziten Lösung geht es darum, die drei Ebenen
   - innerhalb der FOL-Syntax: $\neg$, $\forall$, $\exists$,
   - auf der mathematischen Ebene über die Formel (bzw. die Beziehung zwischen einer Interpretation $\mathcal{I}$ und einer Formel) sprechend: $\models$, $\not\models$,
   - auf der "natürlichen" Metaebene: "es gibt", "für alle", "nicht"

   zu unterscheiden.

   Für alle Interpretationen $\mathcal{I} = (I, \mathcal{D})$ und alle Variablenbelegungen β gilt:

	$\mathcal{I} \models_\beta \forall x : F(x)$	
$\Leftrightarrow$	für alle $d \in \mathcal{D}$, $\mathcal{I} \models_{\beta_x^d} F(x)$	per Definition von $\forall$
$\Leftrightarrow$	es gibt kein $d \in \mathcal{D}$, so dass nicht $\mathcal{I} \models_{\beta_x^d} F(x)$	Schritt auf der Metaebene (über Erfülltsein sprechend)
$\Leftrightarrow$	es gibt kein $d \in \mathcal{D}$, so dass $\mathcal{I} \not\models_{\beta_x^d} F(x)$	Schritt auf die math. Ebene (über die Formel sprechend)
$\Leftrightarrow$	es gibt kein $d \in \mathcal{D}$, so dass $\mathcal{I} \models_{\beta_x^d} \neg F(x)$	Schritt in die FOL-Formel per Definition von $\neg$
$\Leftrightarrow$	nicht (es gibt ein $d \in \mathcal{D}$, so dass $\mathcal{I} \models_{\beta_x^d} F(x)$)	Schritt auf Metaebene
$\Leftrightarrow$	nicht $\mathcal{I} \models_\beta \exists x : F(x)$	per Definition von $\exists$
$\Leftrightarrow$	$\mathcal{I} \not\models_\beta \exists x : F(x)$	Schritt auf die math. Ebene (über die Formel sprechend)
$\Leftrightarrow$	$\mathcal{I} \models_\beta \neg\exists x : F(x)$	per Definition von $\neg$
$\Leftrightarrow$	$\mathcal{I} \models_\beta \neg\exists y : F(y)$	Variable noch umbenennen

   Was hat man jetzt gewonnen?
   - Beweis, dass man die eine Formel in die andere umformen darf und die Semantik dieselbe ist,
   - insbesondere dass diese Umformung in Algorithmen verwendet werden kann.
   - Beweis, dass die Semantik der FOL-Syntax so definiert ist, wie man es sich vorstellt.

b) Sei $\mathcal{I}$ eine Interpretation, die einen solchen Zustand beschreibt. Man weiss also, dass, wenn $\mathcal{I} \models \varphi$ erfüllt ist, auch $\mathcal{I} \models \psi$ erfüllt ist.

$$\begin{aligned} & \mathcal{I} \models \varphi \rightarrow \psi \\ \Leftrightarrow\ & \mathcal{I} \models \neg\varphi \vee \psi \\ \Leftrightarrow\ & \mathcal{I} \models \neg\varphi \text{ oder } \mathcal{I} \models \psi \end{aligned}$$

Fallunterscheidung: Wenn $\mathcal{I} \models \neg\varphi$ der Fall ist, "ja"; sonst ist also $\mathcal{I} \models \varphi$ der Fall, und damit aufgrund des angenommenen kausalen Zusammenhanges auch sonst ist also $\mathcal{I} \models \psi$, und auch "ja".
$\mathcal{I} \models \varphi \rightarrow \psi$ gilt also in jeder solchen Interpretation.

c) Sei $\mathcal{I}$ die Interpretation einer vereinfachten Signatur von Mondial, die die gegenwärtigen geografischen Fakten enthält. Es gilt

$$\mathcal{I} \models \mathsf{encompassed(``Germany",``Europe")} \rightarrow \mathsf{borders(``BR",``RA")}$$

("falls in $\mathcal{I}$ das Faktum (i), dass Deutschland zu Europa gehört zutrifft, trifft auch das Faktum (ii), dass Brasilien und Argentinien Nachbarn sind, zu". (i): Ja. (ii): Ja. – Formel ist erfüllt.)
Aber auch

$$\mathcal{I} \models \mathsf{encompassed(4,\ ``North\ Sea")} \rightarrow \mathsf{borders(``BR",``RA")}$$

("falls in $\mathcal{I}$ das Tupel (i) (4, North Sea) in der Relation "encompassed" enthalten ist (was nicht der Fall ist, weil das Tupel Blödsinn ist), dann ist auch das Tupel (ii) (BR,RA) in der Relation "borders" enthalten." (i) Nein. (ii) Ja - aber das ist schon egal.)
Es gilt auch

$$\mathcal{I} \models \mathsf{encompassed(4,\ ``North\ Sea")} \rightarrow \mathsf{borders(``Berlin",\ ``04.05.2012")}$$

$\neg A \vee B$ – wenn $\neg A$ erfüllt ist, interessiert B keinen mehr.

- Die wichtigste Anwendung, solange man nur einen einzigen Zustand betrachtet, und nicht allgemeines Reasoning betreibt, ist, eine Bedingung ψ nur für eine eingeschränkte Menge von Werten, die durch φ beschrieben wird, zu fordern: "falls x φ erfüllt, soll es auch ψ erfüllen".
  - Man kann Integritätsbedingungen durch allquantifizierte solche Formeln ausdrücken, z.B. $\forall x : (\exists y : \text{encompassed}(x, y)) \rightarrow (\exists n, p, a, cap, cp : \text{country}(n, x, p, a, cap, cp))$.
    "Falls x in encompassed (als Landescode) erscheint, wird gefordert, dass es ein entsprechendes Land in country gibt."
    Wenn ein Zustand diese Formel nicht erfüllt, gibt es in "encompassed" einen Eintrag, der einen Landescode enthält, der in "country" nicht als Schlüsselwert existiert.
    Für die Anfrage

    $$F(x) = (\exists y : \text{encompassed}(x, y)) \wedge \neg(\exists n, p, a, cap, cp : \text{country}(n, x, p, a, cap, cp))$$

    darf es keine Antwortbindungen für X geben.
    Vor allem wird die negierte Form als *Denials* benutzt: Die Formel

    $$\neg\exists x : (\exists y : \text{encompassed}(x, y)) \wedge \neg(\exists n, p, a, cap, cp : \text{country}(n, x, p, a, cap, cp))$$

    muss immer erfüllt sein.
  - Die relationale Division nutzt die materiale Implikation ebenfalls:

    $$F(Y) = (\exists X : R(X, Y)) \wedge \forall X : (S(X) \rightarrow R(X, Y))$$

    $R(X, Y)$ wird nur für diejenigen X gefordert, die in $S(X)$ vorkommen.

Aufgabe 2 (Kalkül: Sichere, Wertebereichsunabhängige Anfragen) Prüfen Sie für die folgenden Anfragen, ob sie in SRNF sind (geben Sie dazu $rr(G)$ für jede ihrer Teilformeln an).

Prüfen Sie auch, ob die Formeln in RANF sind. Wenn nicht, geben Sie eine äquivalente Formel in RANF an.

Geben Sie einen äquivalenten Ausdruck in der relationalen Algebra und in SQL an (entwickeln Sie dabei SQL-Ausdrücke sowohl aus der Originalformel, als auch aus der RANF-Formel).

a) $F(X) = p(X,Y) \land (q(Y) \lor r(Z))$,

b) $F(X) = p(X,Y) \land (q(Y) \lor r(X))$,

c) $F(X,Y) = p(X,Y) \land \neg\exists Z : r(Y,Z)$,

d) $F(X) = p(X) \land \exists Y : (q(Y) \land \neg r(X,Y))$,

e) $F(X) = p(X) \land \neg\exists Y : (q(Y) \land \neg r(X,Y))$

f) $F(X,Y) = \exists V : (r(V,X) \land \neg s(X,Y,V)) \land \exists W : (r(W,Y) \land \neg s(Y,X,W))$

---

a) $p(X,Y) \land (q(Y) \lor r(Z))$:

G	$rr(G)$
$p(x,y)$	X,Y
$q(Y)$	Y
$r(Z)$	Z
$q(Y) \lor r(Z)$	$\{Y\} \cap \{Z\} = \emptyset$
$p(X,Y) \land (q(Y) \lor r(Z))$	$\{X,Y\} \cup \emptyset = \{X,Y\}$

Da $free(F) = \{X,Y,Z\} \neq \{X,Y\} = rr(F)$ ist, ist F nicht in SRNF.
Sie ist auch nicht wertebereichsunabhängig: für $\mathcal{S}$ mit $\mathcal{S}(p) = \{1,a\}$ und $\mathcal{S}(q) = \{(a)\}$ und $\mathcal{S}(r) = \emptyset$ und domain $\mathcal{D}$ ist die Antwortmenge $\{X \mapsto 1, Y \mapsto a, Z \mapsto d | d \in \mathcal{D}\}$.

b) $p(X,Y) \land (q(Y) \lor r(X))$:

G	$rr(G)$
$p(x,y)$	X,Y
$q(Y)$	Y
$r(X)$	X
$q(Y) \lor r(X)$	$\{Y\} \cap \{X\} = \emptyset$
$p(X,Y) \land (q(Y) \lor r(X))$	$\{X,Y\} \cup \emptyset = \{X,Y\}$

Da $free(F) = \{X,Y\} = rr(F)$ ist, ist F in SRNF.
Sie ist nicht in RANF, da die Disjunktion $q(Y) \lor r(X)$ nicht self-contained ist.
In SQL läßt F sich problemlos ausdrücken (P(P1,P2), Q(Q1), R(R1)):

```
SELECT P1,P2
FROM P
WHERE P2 in (SELECT Q1 FROM Q)
  AND P1 in (SELECT R1 FROM R)
```

Etwas außergewöhnlicher wäre die Umsetzung als Outer Join:

```
SELECT P1, P2
FROM P OUTER JOIN Q ON P2=Q1
       OUTER JOIN R ON P1=R1
```

Letzteres geht natürlich auch in der relationalen Algebra.
Der äquivalente Ausdruck aus den Grundoperatoren der Algebra ist
$(P \bowtie [P2 = Q1]Q) \cup (P \bowtie [P1 = R1]R)$.
Letzteren erhält man auch wenn man erst von SRNF nach RANF mit “push-into-or” umformt:
$(p(X,Y) \land q(Y)) \lor (p(X,Y) \land r(Z))$
und dann nach Kochrezept in die relationale Algebra übersetzt.

c) $F(X,Y) = p(X,Y) \wedge \neg\exists Z : r(Y,Z)$,

G	$rr(G)$
$p(X,Y)$	X,Y
$r(Y,Z)$	Y,Z
$\exists Z : r(Y,Z)$	Y
$\neg\exists Z : r(Y,Z)$	$\emptyset$
$p(X,Y) \wedge \neg\exists Z : r(Y,Z)$	X,Y

Da $free(F) = \{X\} = rr(F)$ ist, ist F in SRNF.
Alle Teilformeln sind self-contained.
In SQL läßt F sich problemlos ausdrücken (P(P1,P2), R(R1,R2)):

```
SELECT P1,P2
FROM P
WHERE P2 NOT IN (SELECT R2 FROM R)
```

Der äquivalente Ausdruck aus den Grundoperatoren der Algebra ist
$P \bowtie (\pi[P2](P) - \pi[R1](R))$.
Verwendet man nach Kochrezept die Formel um alle Werte des aktiven Domains (hier: die in P und R vorkommen) zu berücksichtigen:
$P \bowtie ((\pi[P1](P) \cup \pi[P2](P) \cup \pi[R1](R) \cup \pi[R2](R)) - \pi[R1](R))$.

d) $F(X) = p(X) \wedge \exists Y : (q(Y) \wedge \neg r(X,Y))$,

G	$rr(G)$
$p(X)$	X
$q(Y)$	Y
$r(X,Y)$	X,Y
$\neg r(X,Y)$	$\emptyset$
$q(Y) \wedge \neg r(X,Y)$	Y
$\exists Y : q(Y) \wedge \neg r(X,Y)$	$\emptyset$
$p(X) \wedge \exists Y : q(Y) \wedge \neg r(X,Y)$	X

Da $free(F) = \{X\} = rr(F)$ ist, ist F in SRNF.
Sie ist nicht in RANF, da die Teilformel $G = \exists Y : q(Y) \wedge \neg r(X,Y)$ nicht self-contained ist: für den Body $H = q(Y) \wedge \neg r(X,Y)$ ist $free(H) = \{X,Y\} \supsetneq \{Y\} = rr(H)$ (hier sieht man, dass das Problem eigentlich schon im Body $q(Y) \wedge \neg r(X,Y)$ liegt; das Kriterium SAFE würde diese Teilformel als Problem feststellen).
In SQL läßt F sich problemlos ausdrücken (P(P1), Q(Q1), R(R1,R2)):

```
SELECT P1
FROM P
WHERE EXISTS (SELECT Q1
              FROM Q
              WHERE (P1,Q1) IN SELECT R1,R2 FROM R)
```

Der äquivalente Ausdruck aus den Grundoperatoren der Algebra ist
$(P \times Q) \bowtie [P1 = R1, Q1 = R2]R$.
Letzteren erhält man auch wenn man erst von SRNF nach RANF mit "push-into-exist" umformt:
$F(X) = \exists Y : (p(X) \wedge q(Y) \wedge \neg r(X,Y))$,
und dann nach Kochrezept in die relationale Algebra übersetzt.

e) Dieses Beispiel ist das Pattern für die relationale Division:

$F(X) = p(X) \wedge \neg\exists Y : (q(Y) \wedge \neg r(X,Y))$,

G	$rr(G)$
$p(X)$	X
$q(Y)$	Y
$r(X,Y)$	X,Y
$\neg r(X,Y)$	$\emptyset$
$q(Y) \wedge \neg r(X,Y)$	Y
$\exists Y : q(Y) \wedge \neg r(X,Y)$	$\emptyset$
$\neg\exists Y : q(Y) \wedge \neg r(X,Y)$	$\emptyset$
$p(X) \wedge \neg\exists Y : q(Y) \wedge \neg r(X,Y)$	X

Da $free(F) = \{X\} = rr(F)$ ist, ist F in SRNF.
Sie wie in (c) ist nicht in RANF, da die Teilformel $G = \exists Y : q(Y) \wedge \neg r(X,Y)$ schon nicht self-contained ist (und dieselbe Anmerkung wie in (c) gilt).
In SQL läßt F sich problemlos ausdrücken (P(P1), Q(Q1), R(R1,R2)):

```
SELECT P1
FROM P
WHERE NOT EXISTS (SELECT Q1
                  FROM Q
                  WHERE (P1,Q1) NOT IN SELECT R1,R2 FROM R)
```

Der äquivalente Ausdruck aus den Grundoperatoren der Algebra ist
$P - \pi[P1]((P \times Q) - \rho[R1 \to P1, R2 \to P2](R))$.
Letzteren erhält man auch wenn man erst von SRNF nach RANF mit "push-into-not-exist" umformt:
$F(X) = p(X) \wedge \neg\exists Y : (p(X) \wedge q(Y) \wedge \neg r(X,Y))$,
und dann nach Kochrezept in die relationale Algebra übersetzt.

f) Dies ist ein Beispiel für eine Konjunktion, in der keine der beiden Teilformeln self-contained (nach RANF-Definition) ist:
$F(X,Y) = \exists V : (r(V,X) \wedge \neg s(X,Y,V)) \wedge \exists W : (r(W,Y) \wedge \neg s(Y,X,W))$

G	$rr(G)$
$r(V,X)$	X,V
$s(X,Y,V)$	X,Y,V
$\neg s(X,Y,V)$	$\emptyset$
$r(V,X) \wedge \neg s(X,Y,V)$	X,V
$\exists V : (r(V,X) \wedge \neg s(X,Y,V))$	X
$r(W,Y)$	W,Y
$s(Y,X,W)$	X,Y,W
$\neg s(Y,X,W)$	$\emptyset$
$r(W,Y) \wedge \neg s(Y,X,W)$	W,Y
$\exists W : (r(W,Y) \wedge \neg s(Y,X,W))$	Y
$(\ldots) \wedge (\ldots)$	X,Y

Da $free(F) = \{X,Y\} = rr(F)$ ist, ist F in SRNF.
Sie ist nicht in RANF, da die Teilformeln $\exists V : (r(V,X) \wedge \neg s(X,Y,V))$ und $\exists W : (r(W,Y) \wedge \neg s(Y,X,W))$ nicht self-contained sind (wieder liegt das Problem eigentlich in deren konjunktiven Bodies).
In SQL läßt F sich problemlos ausdrücken (R(R1,R2), S(S1,S2,S3)):

```
SELECT rv.R2, r2.R2
FROM R rv, R rw
```

```
WHERE NOT EXISTS (SELECT * FROM S
                  WHERE S1=rv.R2 and S2=rw.R2 and S3=rv.R1)
  AND NOT EXISTS (SELECT * FROM S
                  WHERE S1=rw.R2 and S2=rv.R2 and S3=rw.R1)
```

oder

```
SELECT rv.R2, r2.R2
FROM R rv, R rw
WHERE NOT (rv.R2, rw.R2, rv.R1 IN (SELECT * FROM S))
  AND NOT (rw.R2, rv.R2, rw.R1 IN (SELECT * FROM S))
```

Der äquivalente Ausdruck aus den Grundoperatoren der Algebra ist ... aus dem Kopf nicht so einfach.

Also formt man F von SRNF nach RANF um, indem man das erste Konjunkt mit "push-into-exists" in das zweite verlagert (oder umgekehrt, schlussendlich kommt dasselbe raus):

$$\exists W : \exists V : (r(V,X) \wedge \neg s(X,Y,V)) \wedge (r(W,Y) \wedge \neg s(Y,X,W))$$

Existenzquantoren plattmachen, Konjunktion plattmachen:

$$\exists V, W : B(X,Y,V,W)$$

mit $B = (r(V,X) \wedge r(W,Y) \wedge \neg s(X,Y,V) \wedge \neg s(Y,X,W))$

ist jetzt self-contained mit $free(B) = \{V,W,X,Y\} = rr(B)$

Nach dem Umsetzungsalgorithmus der Vorlesung muss man

- (XYV)-Komponente von B bilden, und davon s abziehen,
- parallel die (XYW)-Komponente von B bilden, davon auch s abziehen,
- das sind jeweils die Tripel, die überleben,
- und diese joinen,
- und dann auf X, Y projizieren:

$$\begin{aligned}\pi[X,Y](\quad & (\pi[X,Y,V](\rho[R1 \to V, R2 \to X](r) \times \rho[R1 \to W, R2 \to Y](r)) \\ & -\rho[S1 \to X, S2 \to Y, R2 \to V](s)) \\ \bowtie\ & (\pi[X,Y,W](\rho[R1 \to V, R2 \to X](r) \times \rho[R1 \to W, R2 \to Y](r)) \\ & -\rho[S1 \to Y, S2 \to X, R2 \to W](s)))\end{aligned}$$

Aufgabe 3 (Relationale Anfragen an Mondial: Schweizer Sprachen) Geben Sie Ausdrücke im relationalen Kalkül für die folgenden Anfragen an die Mondial-Datenbank an. Vergleichen Sie mit denselben Anfragen in der Algebra und in SQL.

a) Alle Landescodes von Ländern, in denen eine Sprache gesprochen wird, die auch in der Schweiz gesprochen wird.

b) Alle Landescodes von Ländern, in denen ausschliesslich Sprachen gesprochen werden, die in der Schweiz nicht gesprochen werden.

c) Alle Landescodes von Ländern, in denen nur Sprachen gesprochen werden, die auch in der Schweiz gesprochen werden.

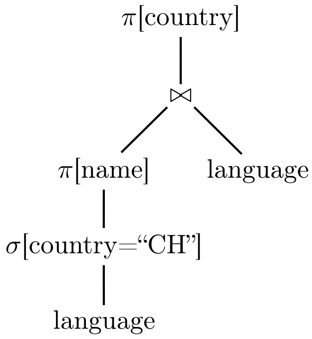
d) Alle Landescodes von Ländern, in denen alle Sprachen gesprochen werden, die in der Schweiz gesprochen werden.

---

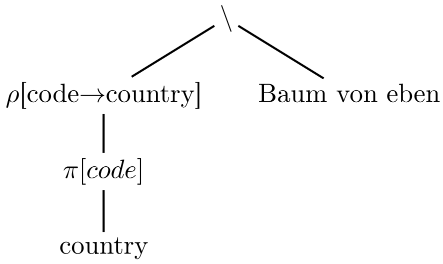
a) $F(C) = \exists L, Perc1, Perc2 : (\text{language}('CH', L, Perc1) \wedge \text{language}(C, L, Perc2))$

b) $F(C) = \exists CN, A, Pop, Cap, CapP :$
$\quad (\text{country}(CN, C, A, Pop, Cap, CapP) \wedge$
$\quad \neg \exists L, Perc1, Perc2 : (\text{language}('CH', L, Perc1) \wedge \text{language}(C, L, Perc2)))$

Algebra: a)



und (b)



c) $F(C) = (\exists CN, A, Pop, Cap, CapP : \text{country}(CN, C, A, Pop, Cap, CapP)) \wedge$
$\neg\exists L, Perc1 : (\text{language}(C, L, Perc1) \wedge \neg\exists Perc1 : \text{language}('CH', L, Perc2))$

d) $F(C) = (\exists CN, A, Pop, Cap, CapP : \text{country}(CN, C, A, Pop, Cap, CapP)) \wedge$
$\forall L : ((\exists Perc1 : \text{language}('CH', L, Perc1)) \rightarrow (\exists Perc2 : \text{language}(C, L, Perc2)))$