P114
Definovatelnost

Manipulace s HIT-atributy

9

Témata

- Identita a semiidentita HIT-atributů
- Definovatelnost
- Informační ekvivalence
- Informační schopnost
- Rotace, triviální odvození

HIT-atributy

- každá taková funkce **A** je dána konstrukcí
  $\lambda wt\ \lambda x_1 \ldots x_n\ \square y\ (C)$
  kde w::Wrd, t::Tim, x_i::T_i, y::S, T_i, S jsou základní typy splňující podmínky definice HIT-atributu, nebo
  $\lambda wt\ \lambda x\ \square y\ (C)$
  kde x :: $(T_1,\ldots,T_n)$
  C je otevřená Bool-konstrukce neobsahující jiné volné proměnné než w, t, x_i, y
  $\mathbf{A}/(Wrd \rightarrow (Tim \rightarrow ((T_1,\ldots,T_n) \rightarrow S)))$ nebo
  $\mathbf{A}/(Wrd \rightarrow (Tim \rightarrow ((T_1,\ldots,T_n) \rightarrow (S \rightarrow Bool))))$,

Základní předpoklad o HIT-atributech:

- **extenze HIT-atributů jsou konečné tabulky**.

- Předpoklad je oprávněný, poněvadž při modelování bereme v úvahu vždy
  - konečný diskrétní časový interval „života“ IS
  - konečnou populaci sort (E-typů a/nebo D-typů)

základní předpoklad o analytických funkcích

- Analytické funkce které budeme dále uvažovat jsou vždy surjekce a jsou to totální funkce.
- ZOPAKOVÁNÍ:
  $f / (T \rightarrow \Re)$, kde $\Re = S$ nebo $\Re = (S \rightarrow Bool)$
  f je injekce (zobrazení prosté), když $\forall x,y \in T([f\ x] = [f\ y] \ \Rightarrow\ x = y)$
  f je surjekce (zobrazení na), když $\forall y \in \Re\ (\exists x \in T([f\ x] = \ y))$
  f je bijekce, když je injekce a surjekce
- často používané budou logické funkce, tj. funkce jejichž oborem hodnot je Bool
- základní předpoklad vylučuje možnost, že by mezi logickými funkcemi, které budeme používat, byly tautologie nebo kontradikce

Identita HIT-atributů

- konstrukce C_1 je ekvivalentní s konstrukcí C_2, jestliže pro libovolnou valuaci *v* a libovolný objekt <u>A</u> platí C_1 v-konstruuje <u>A</u> právě tehdy, když C_2 v-konstruuje <u>A</u> .
- Nechť A_1, A_2 jsou HIT-atributy pořadě konstruované konstrukcemi C_1 a C_2.
- Jestliže C_1 je ekvivalentní s C_2, pak HIT-atributy A_1 a A_2 jsou identické a píšeme $A_1 = A_2$.
- To, že konstrukce C konstruuje HIT-atribut A budeme pro jednoduchost rovněž zapisovat A = C

Problém s parcialitou HIT-atributů

- Budeme konstruovat atributy pomocí jiných atributů
- atribut může být na některých argumentech nedefinován: konstrukce jej používající pak je pro příslušnou valuaci v-nevlastní
- jiná konstrukce, která konstruuje prakticky totéž, však na těchže argumentech vrací { }.
- Tento rozdíl nemá vliv na množinu z atributu generovaných propozic
- Proto se zavádí tzv. semiidentita

Semiidentita

Nechť $A = \lambda wt\, \lambda x\, \square y\, (C)$, $B = \lambda wt\, \lambda x\, \square y\, (C_1)$

Atributy A, B nazveme semiidentické, jestliže

- buďto jsou identické
- nebo se liší následujícím způsobem:
  $\forall wtx\, ([A_{wt}x] = [\, B_{wt}x] \vee ([A_{wt}x] = \{\ \} \wedge [\, B_{wt}x] = \{\perp\}))$
  kde { } je prázdná množina, to v případě, že C je na x Nepravda, a
  $\{\perp\}$ je třída taková, že o žádném prvku daného typu nemůžeme říci, že patří do této třídy proto, poněvadž C_1 je na x nedefinováno.

V dalším bude A = B značit i semiidentitu.

Definovatelnost

- Atribut A je definovatelný nad množinou atributů $\{B_1, ..., B_n\}$ právě když
  $\exists f\, (\forall wt\, ([A\ wt] = [f\, ([B_1\ wt], ..., [B_n\ wt])]))$
  kde f je analytická funkce splňující základní předpoklad, w je možný svět, t je časový okamžik.
  Píšeme: $A \leftarrow (B_1, ..., B_n)$
- Množina atributů M je definovatelná nad množinou atributů N, je-li každý prvek z M definovatelný nad nějakou podmnožinou N.
  Píšeme: $M \leftarrow N$
- Atribut A je definovatelný nad atributem B, když je definovatelný nad množinou $\{B\}$.
  Píšeme: $A \leftarrow B$

Informační ekvivalence

- Nechť $M = \{A_1, ..., A_n\}$, $N = \{B_1, ..., B_m\}$ jsou množiny atributů a nechť $M \leftarrow N$ a zároveň $N \leftarrow M$. Potom říkáme, že M a N jsou informačně ekvivalentní. Píšeme: $M \approx N$
- Intuitivně je zřejmé, a lze dokázat, že dvě informačně ekvivalentní množiny atributů generují v každém stavu světa stejnou třídu propozic.
- Jiná intuice: z n-tice tabulek daných množinou atributů M dostaneme tutéž informaci jako z m-tice tabulek daných množinou atributů N
- VĚTA: Relace $\approx$ na množině atributů je ekvivalence.

Důsledek

- Nechť atribut A je definovatelný nad množinou atributů $\{B_1, ..., B_n\}$.
  Potom $\{A, B_1, ..., B_n\} \approx \{B_1, ..., B_n\}$.
- Intuitivně: přidáním definovatelného („odvoditelného“) atributu nic nového nezískáme, nebo
  odebráním definovatelného („odvoditelného“) atributu nic z původního neztratíme
- Co je to, co nezískáme ani neztratíme ?

quasi-uspořádání

- Relace definovatelnosti mezi množinami atributů je:
  - reflexivní (evidentně $M \leftarrow M$, za f stačí vzít identitu)
  - tranzitivní (viz důkaz předchozí věty)
- Avšak není antisymetrická:

  tj. z $M \leftarrow N$ a $N \leftarrow M$ neplyne $M = N$

  viz předchozí důsledek
- Tedy není to relace částečného uspořádání, ale pouze quasi-uspořádání
- Z Algebry:: lze definovat částečné uspořádání na množině ekvivalenčních tříd

Informační schopnost

- Nechť **BZT** je nějaká báze základních typů. Nechť **ATR** je množina všech možných HIT-atributů nad **BZT**.
- Faktorová množina **ATR** / ≈ (tj. množina všech tříd ekvivalence nad **ATR**) je částečně uspořádaná relací indukovanou definovatelností. Značíme ∠
- Každá třída ekvivalence z **ATR** / ≈ definuje určitou **informační schopnost**. Relace ∠ je částečným uspořádáním informačních schopností.

Tvrzení o informační schopnosti

- Nechť C_M, $C_N \in$ **ATR** / $\approx$, $M \in C_M$, $N \in C_N$
- $C_M \angle C_N$ právě když $M \leftarrow N$
- **Tvrzení 1:**
  všechny prvky třídy C_M mají stejnou informační schopnost
- **Tvrzení 2:**
  každý prvek C_M má menší informační schopnost než každý prvek třídy C_N
- **Tvrzení 3:**
  Jestliže pro C_M a C_N neplatí ani $C_M \angle C_N$, ani $C_N \angle C_M$, pak informační schopnost libovolného prvku z C_M je nesrovnatelná s informační schopností libovolného prvku z C_N.

dohoda na označení, příklad

- A = počet (PrirozCislo) druhů výrobků vyráběných daným výrobcem (#Vyrobce) / 0,1:0,M
- $A/(Wrd \rightarrow (Tim \rightarrow (\#Vyrobce \rightarrow PrirozCislo)))$
  to lze psát (Shönfinkelova redukce):
  $A/((Wrd,Tim) \rightarrow (\#Vyrobce \rightarrow PrirozCislo))$
  toto platí pro všechny HIT-atributy
- Označíme (Wrd, Tim) = W ... stav světa, potom:
  $A/(W \rightarrow (\#Vyrobce \rightarrow PrirozCislo))$
  a konstrukce A má tvar:
  $A = \lambda w\ \lambda x\ \iota y\ ([[Aw]\ x] = y)$, kde
  w::W, x::#Vyrobce, y::PrirozCislo
- Aplikace [Aw] je velmi častá; proto zkracujeme: A_w
- A_w je extenze atributu A ve stavu světa w (je to tabulka)

Příklad

- Atribut A je definovatelný nad atributem B = druhy výrobků (#Vyrobek)-s vyráběné daným výrobcem (#Vyrobce) / 0,M:0,M
- Odvození: (x::#Vyrobce, z::#Vyrobek, y::PrirozCislo)
  $B = \lambda w\ \lambda x\ \lambda z\ ([[B_w\ x]\ \ z])$
  hledáme f tak, aby pro všechna w
  $A_w = [f\ B_w]$:
  $b :: (\#Vyrobce \rightarrow (\#Vyrobek \rightarrow Bool))$,
  $a :: (\#Vyrobce \rightarrow PrirozCislo)$ jsou proměnné, pak
  $f = \lambda b\ \iota a\ (\forall x\ ([ax] = [Card\ [bx]]))$,
  kde Card je funkce kardinality množiny
  Skutečně (beta-pravidlo):
  $[f\ B_w] = \iota a\ (\forall x\ ([ax] = [Card\ [B_w x]])) = A_w$

méně formálně

- **f** je algoritmus, který nezávisle na stavu světa vypočítává z tabulky **b** (mimochodem nenormalizované) tabulku **a**.
- V praxi se netrápíme hledáním konstrukce funkce f ve tvaru lambda-termu, ale stačí, když se přesvědčíme o existenci algoritmu, který **- ale v každém stavu světa -** vypočítává z jedné tabulky tu druhou
- lambda-termy potřebujeme pro důkazy obecných tvrzení, které aplikujeme v praxi při vytváření konkrétních datových modelů

Příklad

- A_1 = plat (Pl) daného (#Zamest) / 0,1:0,M
  A_2 = (#Zamest)-s daného (#Podnik) / 0,M:1,M
- B = průměrný plat (Pl) v daném (#Podnik) / 0,1:0,M
- B je definovatelný nad $\{A_1, A_2\}$
  toto intuitivně cítíme;
  (mohl by poměr B být 1,1:0,M ??)
- **Jak se hledá konstrukce příslušné odvozovací fce f ? Tj. její algoritmus ?**
- fce f počítá v každém $w \in W$ tabulku atributu B z tabulek atributů A_1 a A_2

Příklad - pokračování

- tedy při označení z :: #Zamest, p :: #Podnik
- for all p:Podnik do

  write p (* první sloupec tabulky B *)

  $Sum_1 = 0$; $Sum_2 = 0$

  for all z:Zamest do

  if $[[A_2 p]z]$ then $Sum_1 = Sum_1 + [A_1 z]$

  $Sum_2 = Sum_2 + 1$

  endfor

  write Sum_1/Sum_2 (* druhý sloupec tabulky B *)

  endfor

Rotace atributu

- Nechť atribut A je dán konstrukcí
  $A = \lambda w\ \lambda x_1 \ldots x_{n-1}\ \Box x_n\ ([A_w(x_{1,} \ldots, x_{n-1})] * x_n)$
  kde $x_i :: T_i$, T_i jsou uzlové typy.
  Nechť $(i_1, \ldots, i_n)$ je libovolná permutace indexů $(1, \ldots, n)$.
  Potom atribut rotA daný konstrukcí
  $rotA = \lambda w\ \lambda x_{i1} \ldots x_{in-1}\ \Box x_{in}\ ([A_w(x_{1,} \ldots, x_{n-1})] * x_n)$
  se nazývá rotací atributu A.
- Jestliže □ v rotA zastupuje λ, říkáme že je to λ-rotace, resp. plurální rotace. Jestliže □ v rotA zastupuje ι, říkáme že je to ι-rotace, resp. singulární rotace.
- Rotace atributu A se nazývá **přípustná**, jestliže $A \leftarrow rotA$.
- Nebude-li řečeno jinak, označuje rotA v dalším přípustnou rotaci.

Lemma 1

- Nechť A je dán v plurální rotaci, tj.
  $A = \lambda w\ \lambda x_1 \ldots x_{n-1}\ \lambda\ x_n\ [[A_w(x_{1,}, \ldots, x_{n-1})]\ x_n]$
  Potom každá jiná jeho plurální rotace je přípustná.

Přípustné singulární rotace

- A = (#Podnik)-s ve kterých pracuje daný (#Zamest)
- rot_1A = (#Zamest)-s daného (#Podnik)
  rot_2A = (#Zamest) daného (#Podnik)
- Zřejmě rot_1A je přípustná a rot_2A není přípustná:
  rot_2A / (W $\rightarrow$ (#Podnik $\rightarrow$ #Zamest))
  tj. v každém stavu světa je definovaná pouze na podnicích, které mají nejvýše jednoho zaměstnance
  tedy:
  z rot_2A nelze odvodit zpět atribut A, poněvadž neoprávněným použitím singularizátoru jsme ztratili informaci
- **POZN.: obrácená funkce** (viz přednáška 7) je speciální případ rotace atributu

Věta 1
(o informační ekvivalenci a rotacích)

- Každá přípustná rotace atributu A je informačně ekvivalentní s A. Každá nepřípustná rotace A není informačně ekvivalentní s A.

Pravidlo singularity

- Z daného atributu nelze žádným formálním způsobem odvodit, které jeho rotace jsou přípustné singulární rotace.
- To že nějaká singulární rotace je přípustná je netriviální empirický fakt.
- Studiem přípustnosti singulárních rotací určujeme poměr atributu.
- **PRAVIDLO:**
  Při datovém modelování je třeba s každým HIT-atributem prozkoumat všechny jeho rotace, a zjistit zda má nějaké přípustné singulární rotace.
  Pokud ano, do výsledného modelu zapisujeme vždy tyto přípustné singulární rotace, nikoli jiné - plurální - rotace.

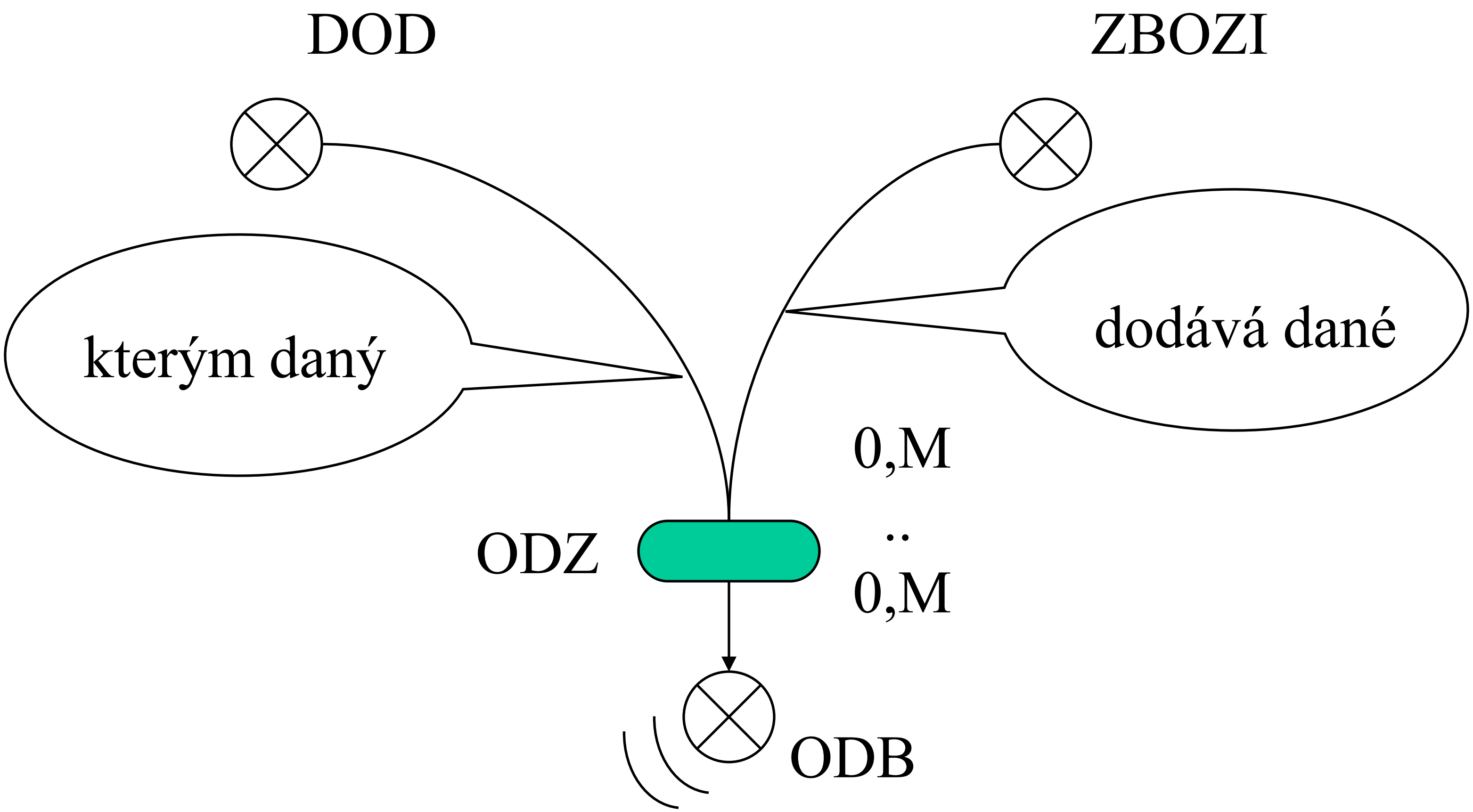
Triviální odvození

- Řekneme, že atribut A vznikl z atributu B triviálním odvozením, jestliže A ← B, tj. pro všechna w
  $A_w = [f\ B_w]$
  a funkce **f** obsahuje nejvýše identitu, přípustně použitý singularizátor a/nebo existenční kvantifikátor.
- Triviální odvození jsou nejčastěji používána jako nástroj odvození odpovědi na dotazy nad databází. Pro **formulování dotazů** je vhodnější využívat všech funkcí jednoduchých typů (přednáška 6) a neomezovat se pouze na H-typy. Pro vlastní modelování je toto omezení (pouze na H-typy) výhodnější.

obecná rotace atributu

- Vychází z obecnějšího pojetí atributu jako funkce jednoduchého typu:
  $A = \lambda w\ \lambda x_1 ... x_k\ \square\ x_{k+1} ...\ x_n\ [A_w(x_{1,},..., x_k)]*(x_{k+1}, ..., x_n)$
- Nechť $(i_1, ..., i_n)$ je libovolná permutace indexů $(1, ..., n)$.
  Potom atribut rotA daný konstrukcí
  rotA =
  $= \lambda w\ \lambda x_{i1} ... x_{ik}\ \square x_{ik+1} ...\ x_{in}\ [A_w(x_{1,},..., x_k)] * (x_{k+1} ..., x_n)$
  se nazývá rotací atributu A.
  Plurální a singulární rotace jako v původní definici.
- Používá se v důkazech a odvozování, nikoli jako modelovací konstrukt.

Příklad:



příklad - pokračování

- ROTACE:
  ZDO = (#Zbozi)-s dodávaná daným (#Dodavatel) danému (#Odberatel) / 0,M:0,M
  DZO = (#Dodavatel)-s kteří dodávají dané (#Zbozi) danému (#Odberatel) / 0,M:0,M
  podle lemmatu víme, že jsou přípustné. Empirickým výzkumem se přesvědčíme, že žádná singulární rotace není přípustná.
- TRIVIÁLNĚ ODVOZENÉ ATRIBUTY:
  ZD = (#Zbozi)-s dodávaná daným (#Dodavatel) / 0,M:0,M
  OZ = (#Odberatel)-s daného (#Zbozi) / 0,M:0,M
  OD = (#Odberatel)-s daného (#Dodavatel) / 0,M:0,M
- Příklad algoritmu odvození:

OD ← ODZ

- for all x:DOD do

  for all y:ODB do

  if $\exists$ z:ZBOZI ([[ODZ_w(x, z)] y])

  then write y

  else endif

  endfor

  endfor

příkazem write vytváříme tabulku odběratelů příslušných k danému dodavateli

příklad obecné rotace

- Dvojice zboží + dodavatel (#Zbozi, #Dodavatel) přiřazená k danému odběrateli (#Odberatel) taková, že onen dodavatel dodává ono zboží tomu dodavateli / 0,M:0,M
- nakreslete si schéma tohoto atributu

- Z lineárního zápisu je vidět, že obecné rotace nejsou vhodné jako modelovací konstrukty. Avšak pro úvahy o správnosti a adekvátnosti vystižení reality a pro důkazy určitých zákonitostí (viz následující přednáška) jsou potřebné.