

P114

Pojmy metody HIT

sortalizace, základní a jednoduché
typy

7

Témata

- zaostření pozornosti a báze základních typů
- definice bazových typů
- HIT-atributy
- sémantika HIT-atributů
- integritní omezení
- poměr HIT-atributu
- podstata modelování

„Zaostření“ pozornosti

- proces modelování (zájmové části) světa se podobá zaostřování při fotografování: před našim objektivem je vše, my ale snímkem zobrazíme jasně pouze to, na co jsme zaostřili
- To znamená:
 - nad **EB** vybereme určité typy a na ty zaostříme pozornost
 - modely pak obsahují pouze konstrukce těchto „zaostřených“ typů
- Zaostřené typy jsou: funkce tzv. jednoduchých typů (viz přednáška 3) a funkce definující tzv. sorty (viz dále)

vztah sort k modelování

- **entitní a deskriptivní sorty** jsou objekty zájmu, o kterých hovoří uživatelé IS, a ze kterých bereme hodnoty funkcí a jejich argumentů při popisu výseku reálného světa
- jejich podmnožiny jsou definiční obory a obory hodnot funkcí uložených formou tabulek v **databázích**
- v **modelech** nás zajímají **konstrukce** těch **objektů** (tj. funkcí nad **EB**), které „v rozumném časovém okolí přítomnosti“ a v aktuálním světě poskytují tyto sorty (*pro případ entitních sort*), resp. poskytují tyto sorty nezávisle na stavu světa (*pro případ deskriptivních sort*)

... zopakování

- **EB** = {Bool, Univ, Tim, Wrd}
- Typ nad **EB**:
 - Bool, Univ, Tim, Wrd
 - jsou-li T_1, \dots, T_n typy nad **EB**, pak (T_1, \dots, T_n) je typ nad **EB**
 - jeli ještě T typ nad **EB**, pak $((T_1, \dots, T_n) \rightarrow T)$ je typ nad **EB**
- Nad **EB** máme celou hierarchii typů
- jejich prvky jsou vesměs funkce
- tyto prvky umíme identifikovat pomocí konstrukcí
 - atomických (přesněji atomických + „trivializace“)
 - aplikace
 - n-tice, projekce
 - abstrakce
- pro obecné manipulace máme modifikovaný typovaný lambda-kalkul

Princip zaostření pozornosti (1)

- Nad **EB** *definujeme* tzv. **uzlové typy**, kterými jsou entitní nebo deskriptivní sorty
- *POZOR!!! Ve slovním spojení **uzlové typy** má nyní slovo **typ** jiný význam než v definici typů nad bází *B* resp. nad epistémickou bází.*
- Přidáme bázový typ Bool a n-ticové typy, čímž získáme množinu **základních typů**
- *Provádíme přesun od epistémické báze k „bázi základních typů“, která je tvořena něčím jiným než původními typy a dokonce ani nemá vlastnost báze v původním smyslu !!!*

Princip zaostření pozornosti (2)

- Nad základními typy *konstruujeme* HIT- atributy, jako **prvky** tzv. jednoduchých typů
- TEDY: v hierarchii typů nad **EB** provedeme **zaostřením pozornosti** jakousi redukci, a jednotlivé prvky (objekty) konstruujeme jenom nad základními typy;
konstrukcí prvků základních typů se nezabýváme ! Nahrazujeme ji definicemi !
- Podrobněji viz DM2

modifikovaná hierarchie typů pro DM

- E-typy, D-typy
- uzlové typy
- n-ticové typy
- základní typy
- prvky H-typů



sortalizace

- Určení základních typů pro danou zájmovou oblast, kterou chceme modelovat, se nazývá **sortalizace**.
- Název souvisí s bází sort BS - viz minulá přednáška. Sortalizace je to, co jiní autoři nazývají klasifikací, či výběrem základních tříd a pod.
- Sortalizaci provádíme procesem „zaostření pozornosti“

Entitní typy (E-typy)

- E-typem nazýváme každou entitní sortu (extenzi) definovanou „v rozumném časovém okolí přítomnosti“ a v aktuálním světě pomocí vlastností nad **EB**, tj. pomocí intenzí
- Samotný E-typ je extenze !
- typicky, ale ne vždy, bývá E-typ podmnožinou Univ (viz dále)
- E-typy jsou **nedisjunktní** a typově-teoreticky **polymorfní**
- ZBOZI, DODAVATEL, ODBERATEL, ZAMESTNANEC, ...

E-typy (pokračování)

- Při praktickém použití jsou E-typy často třídy individuí, které vnímáme závisle na stavu světa (na wt)
- Ale mohou to být třídy libovolně složitých typů T nad **EB** (např. DODÁVKA, VÝPUJČKA, ALGORITMUS, DRUH VÝROBKU, ...)
- Abychom se dorozuměli (veškeré modelování děláme pro zlepšení komunikace !), **musíme každý E-typ definovat**
- Pro definici E-typu používáme vždy takového pojmového systému, který pokládáme za srozumitelný těm, kterým definici předkládáme (o tom je třeba učinit dohodu na tzv. formovacím semináři)
- Praktický problém je ve vnímání množiny Univ: říkáme sice, že Univ je nám dána a priori, ale jak se postupně „učíme“ rozumět určitému prostředí, tak se naše vnímání množiny Univ obohacuje o nové prvky, které jsme v předchozím stádiu učení ještě nevnímali.

definice E-typů

- Prvkem typu (#jmeno E-typu) je každý (takový objekt), pro který platí ...
- Příklady (pragmaticky zjednodušené):
 - Objektem typu (#Artikl) je každý produkt nebo služba nebo právo, který může být předmětem nákupu či prodeje a to včetně produktů, služeb nebo práv dosud neexistujících, ale potenciálně vytvořitelných pro účely rozvojových aktivit obchodní společnosti.
 - Objektem typu (#Dokument) je každý záznam nebo zpráva, jehož/jejíž zaznamenání má pro organizaci smysl.
 - Objektem typu (#Business Partner) je každé takové individuum, které je, bylo nebo může být účastno obchodních aktivit naší společnosti a které je zajímavé z pohledu rozvojových aktivit naší společnosti.

Deskriptivní typy (D-typy)

- D-typem je každá analytická funkce, která poskytuje deskriptivní sortu
- obvykle charakteristická funkce ($T \rightarrow \text{Bool}$), kde $T = \text{Tim}$ nebo $T = \text{Univ}$, při čemž do Univ zahrnujeme všechny možné řetězce znaků reprezentovatelné na počítači
- D-typy jsou vždy extenze
- DATUM, PLAT, RODNE CISLO, TEL-CISLO, JMENO, ...
- každý D-typ použitý v modelu musí být definován

definice D-typů

- Prvkem typu (jméno D-typú) je každý řetězec znaků (každé číslo), který splňuje podmínky ...
- Příklady:
 - Prvkem typu (Datum) je každý řetězec číslic 8 znaků dlouhý, který má tvar RRRRMMDD, kde RRRR je číslo roku, MM je číslo dvouciferné měsíce a DD je dvouciferné číslo dne v měsíci.
 - Prvkem typu (Jméno) je každý maximálně 45 znaků dlouhý řetězec písmen a znaků „-“, „.“, „ „, který začíná písmenem, a ve kterém v každé dvojici sousedních znaků je alespoň jedno písmeno.
 - Prvkem typu (Mnozství) je každé přirozené číslo nebo nula.
 - ...

n-ticové typy

- n-ticový typ je každý typ tvaru (D_1, \dots, D_m) , kde každé D_i je D-typ (pragmaticky zúžená definice)
- D-typy v n-ticovém typu nemusí být nutně různé
- n-ticové typy jsou extenze
- POZN.: někteří autoři připouštějí i n-ticové typy, ve kterých se jako komponenty vyskytují i E-typy. To je pro modelování zbytečné (vždy lze obejít vhodnou konstrukcí prvku H-typu) a v praxi spíše nevýhodné. Avšak pro formulace a důkazy obecných tvrzení je potřebné n-ticový typ brát v tomto širším smyslu.
- n-ticový typ má svoje jméno (jako každý jiný základní typ)
- OBDOBI(OD, DO), ADRESA(PSC, MESTO, ULICE, CISLO_DOMU)

definice n-ticového typu

- $T = (D_1, D_2, \dots, D_m)$,
kde
 D_1, D_2, \dots, D_m
již byly definovány jako D-typy
- $Adresa = (PSC, Mesto, Ulice, Cislo_domu)$,
(**Mesto** znamená jméno města, **Ulice** znamená jméno ulice)

H-typy

- T_1, \dots, T_n necht' jsou uzlové (ne nutně různé) typy, z nichž alespoň jeden je E-typem.
- S necht' je libovolný základní typ
- H-typem je každý typ formy
 $(\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow S)))$
nebo
 $(\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow (S \rightarrow \text{Bool}))))$
- ve druhém případě S nesmí být Bool

HIT-atributy (v TIL s jtt)

- Objekt A necht' je nějakého H -typu

tj.

$A / (\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow S)))$

resp.

$A / (\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow (S \rightarrow \text{Bool}))))$

- potom objekt A nazýváme HIT-atribut

- Každý HIT-atribut je intenze

- HIT-atribut A je konstruován konstrukcí

$\lambda_{\text{wt}} \lambda_{x_1 \dots x_n} \iota y ([A_{\text{wt}}(x_1, \dots, x_n)] = y)$ v prvním případě

a

$\lambda_{\text{wt}} \lambda_{x_1 \dots x_n} \lambda y [[A_{\text{wt}}(x_1, \dots, x_n)] y]$ ve druhém případě

HIT-atributy - pokračování

- A/T, kde T je H-typ

tj.

$A / (\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow S)))$

resp.

$A / (\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow (S \rightarrow \text{Bool}))))$

- číslo $n+1$ nazýváme složitost atributu
- Říkáme, že HIT-atributy jsou konstruovány nad bází základních typů **BZT**.
- obecně píšeme, že HIT-atribut je konstruován konstrukcí
 $\lambda_{wt} \lambda_{x_1 \dots x_n} \square y ([A_{wt}(x_1, \dots, x_n)] * y)$
kde \square stojí namísto λ nebo ι ,
a $*$ stojí namísto aplikace nebo identity

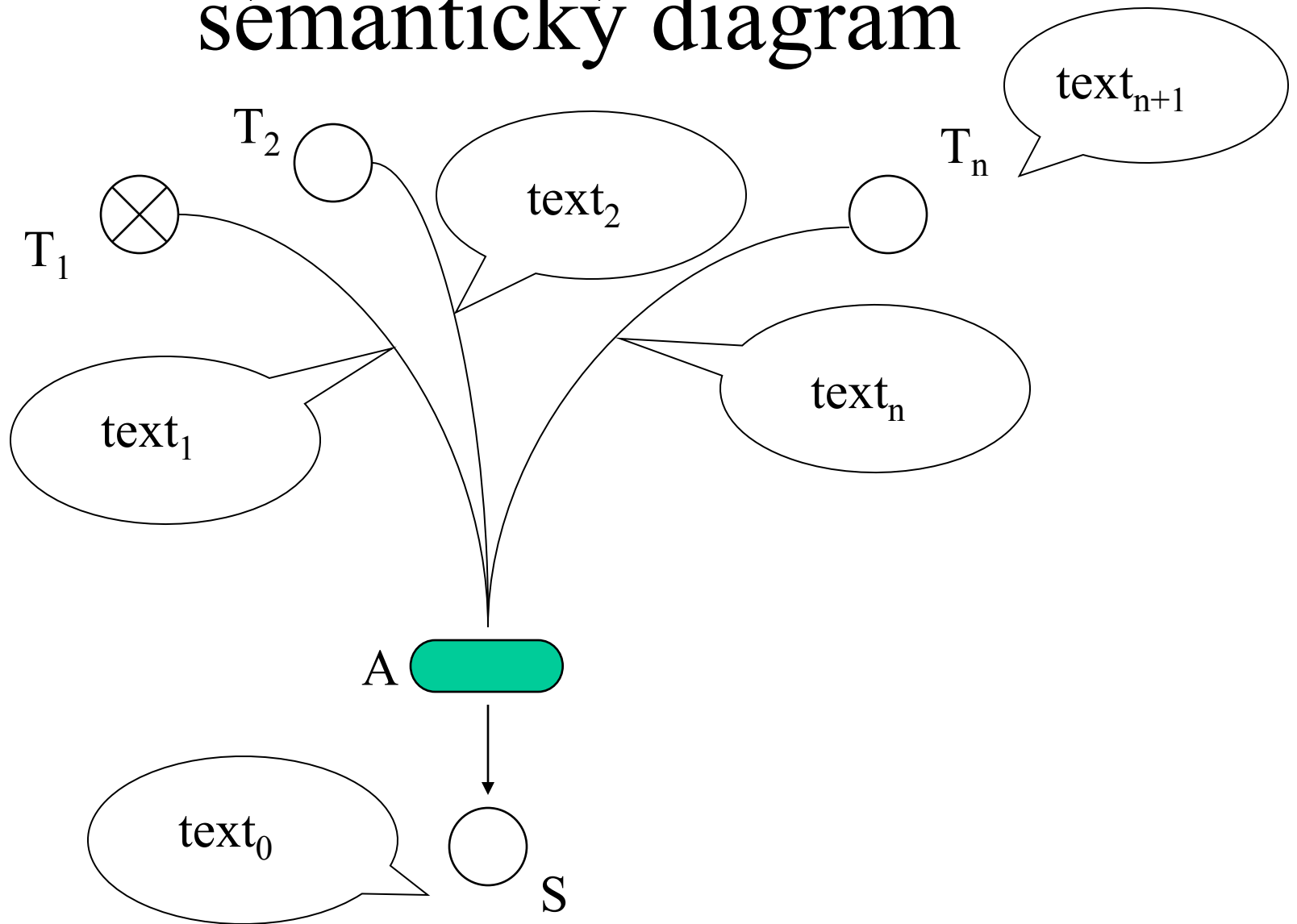
zápis sémantiky

- povahu matematické funkce vyjádříme výrazem umělého jazyka
- znalost interpretace symbolů tohoto umělého (matematického) jazyka nám umožňuje rozumět sémantice matematické funkce
- **povahu funkce** popisující svět, tj. povahu HIT-atributu vyjádříme **výrazem přirozeného jazyka**
- znalost komunikace v přirozeném jazyce nám umožňuje rozumět sémantice těchto funkcí

zápis sémantiky (2)

- $A / (\text{Wrd} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow S)))$
- $A = \text{text}_0(S) \text{text}_1(T_1) \text{text}_2(T_2) \dots \text{text}_n(T_n) \text{text}_{n+1}$
- kde pouze text_0 a text_{n+1} mohou být vynechány a **celý výraz (čtený jako část věty přirozeného jazyka) nám sděluje funkci, která dává hodnoty z S na argumentech z n -tice (T_1, \dots, T_n)**
- tomuto jednoznačně odpovídá (sémantický) diagram

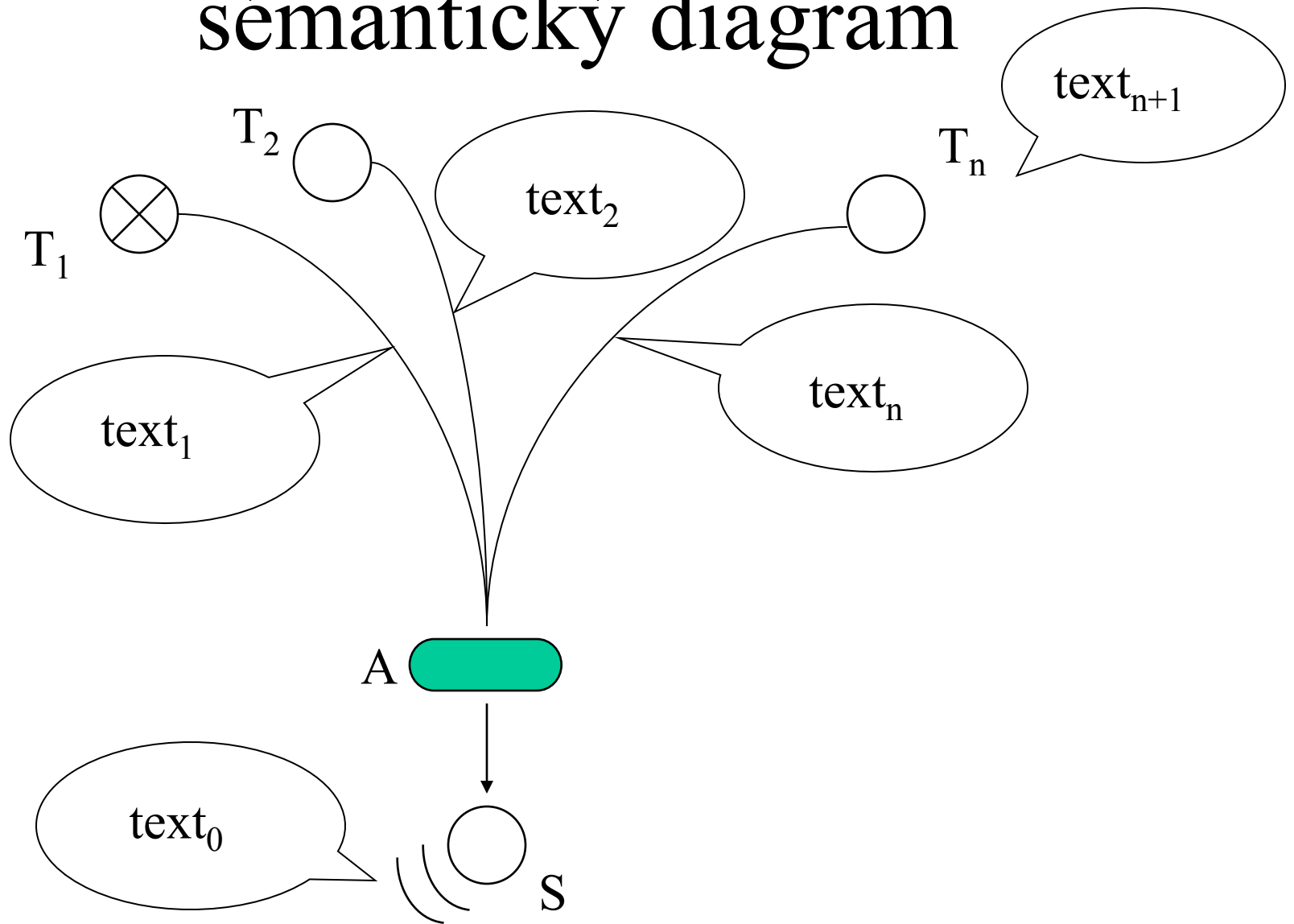
sémantický diagram



zápis sémantiky (3)

- $A / (\text{Wrđ} \rightarrow (\text{Tim} \rightarrow ((T_1, \dots, T_n) \rightarrow (S \rightarrow \text{Bool}))))$
- $A = \text{text}_0(S)\text{-s text}_1(T_1) \text{ text}_2(T_2) \dots \text{text}_n(T_n) \text{ text}_{n+1}$
- kde pouze text_0 a text_{n+1} mohou být vynechány a **celý výraz (čtený jako část věty přirozeného jazyka) nám sděluje funkci, která dává hodnoty z 2^S na argumentech z n-tice (T_1, \dots, T_n)**
(přípona -s odlišuje od prvního případu a nabádá ke čtení v množném čísle)
- tomuto jednoznačně odpovídá (sémantický) diagram

sémantický diagram



Poznámky:

- text_0 (je-li uveden) vyjadřuje roli oboru hodnot funkce
- text_i ($1 \leq i \leq n$) vyjadřuje roli i -tého argumentu funkce
- text_{n+1} je nepovinný dodatek pro lepší porozumění
- *proč je špatně, napsat pouze oznamovací větu (propozici) namísto funkci vyjadřujícího výrazu !*
- *Problém jednoznačné interpretace výrazů NL, který je přirozeně nejednoznačný*

Integritní omezení

- vycházejí z našeho porozumění výrazům přirozeného jazyka a z naší znalosti popisované reality
- vyjadřují to, že ne všechny funkční hodnoty datových funkcí (E-typů nebo HIT-atributů) jsou přípustné
- integritní omezení je dáno pravdivostí určité propozice, tj. vždy přináší informaci
- nazvěme databázovým stavem každé možné naplnění databázových tabulek
- integritní omezení definují přípustná naplnění databázových tabulek, tzv. databázový prostor
- speciálním případem je tzv. **poměr HIT-atributu**
- ponecháváme stranou tzv. analytická integritní omezení ...

Poměr HIT-atributu

- Poměrem HIT-atributu A rozumíme zápis

$$p,m:q,n$$

kde: $p, q = 0 \vee 1$; $m, n = 1 \vee M$

- $p = 0$ znamená, že A je parciální v užším smyslu
- $p = 1$ znamená, že A je totální funkce
- $m = 1$ znamená $A/(\dots \rightarrow S)..$ jednoznačná fce
- $m = M$ znamená $A/(\dots \rightarrow (S \rightarrow \text{Bool}))..$ mnohoznačná fce
- q, n má stejný význam pro funkci „obrácenou“ A^{-1}

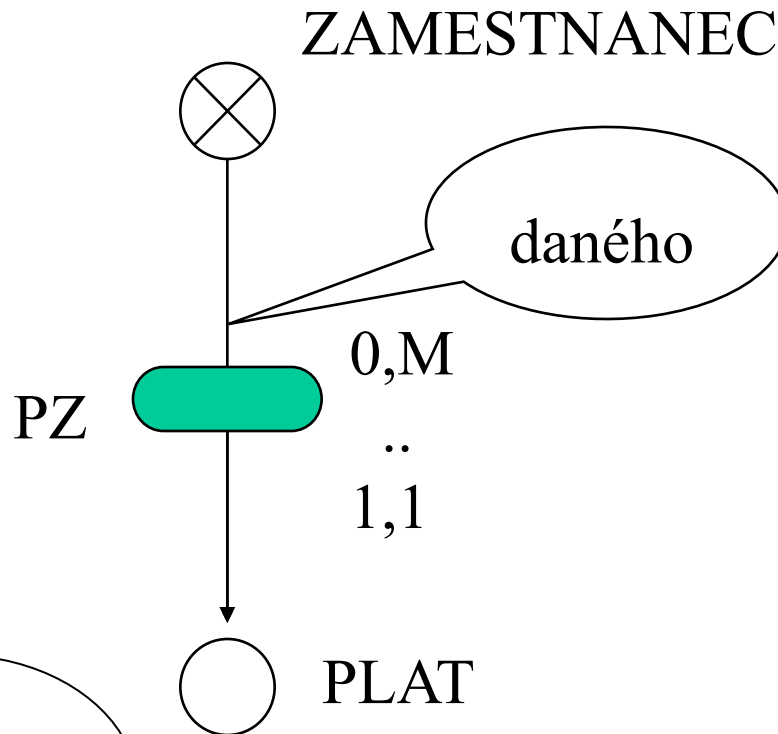
„obrácená“ funkce

- $A / ((T_1, \dots, T_n) \rightarrow S)$,
pak „obrácená“ funkce je
 $A^{-1} / (S \rightarrow (T_1, \dots, T_n))$ nebo
 $A^{-1} / (S \rightarrow ((T_1, \dots, T_n) \rightarrow \text{Bool}))$
- $A / ((T_1, \dots, T_n) \rightarrow (S \rightarrow \text{Bool}))$,
pak „obrácená“ funkce je
 $A^{-1} / (S \rightarrow (T_1, \dots, T_n))$ nebo
 $A^{-1} / (S \rightarrow ((T_1, \dots, T_n) \rightarrow \text{Bool}))$
- Pozor: obrácená funkce není totéž co funkce inverzní v matematice

Příklady HIT-atributů

- PZ: smlouvou deklarováný plat (Plat) daného zaměstnance (#Zamestnanec) v mateřském podniku /1,1:0,M
- DZ: dodavatelé (#Dodavatel)-s dodávající dané zboží (#Zbozi)
- ODZ: smluvní odběratelé (#Odberatel)-s kterým daný dodavatel (#Dodavatel) dodává dané zboží (#Zbozi) /0,M:0,M
- A32: sledovaná množství (Mnozstvi)-s daného druhu zboží (#Zbozi) dodaná daným dodavatelem (#Dodavatel) danému odběrateli (#Odberatel) /0,M:0,M
- A33: množství (Mnozstvi) daného druhu zboží (#Zbozi) dodané daným dodavatelem (#Dodavatel) danému odběrateli (#Odberatel) v daném čase (Cas) /0,1:0,M

Příklady:



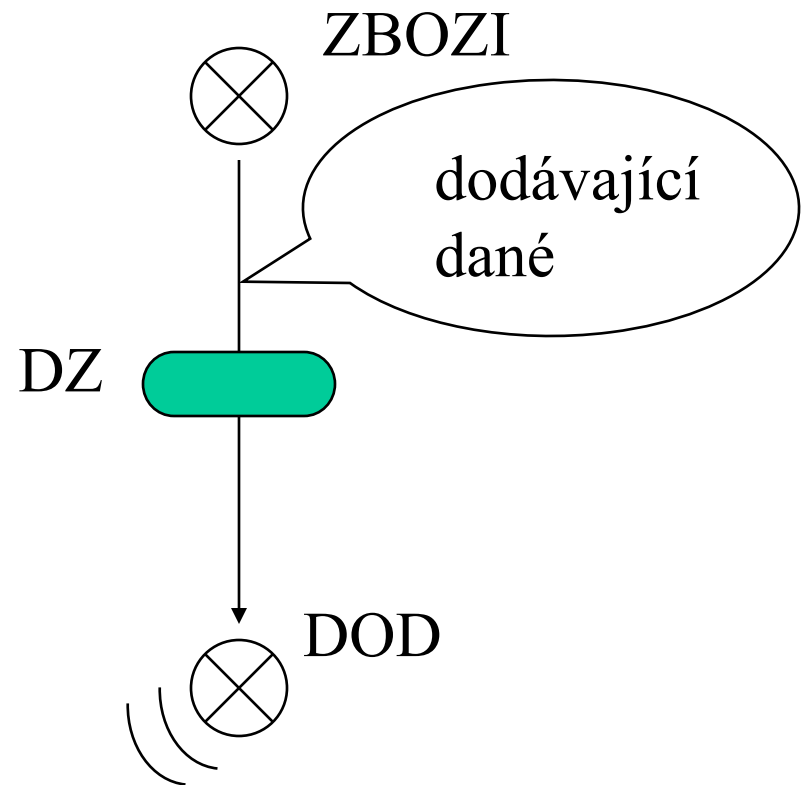
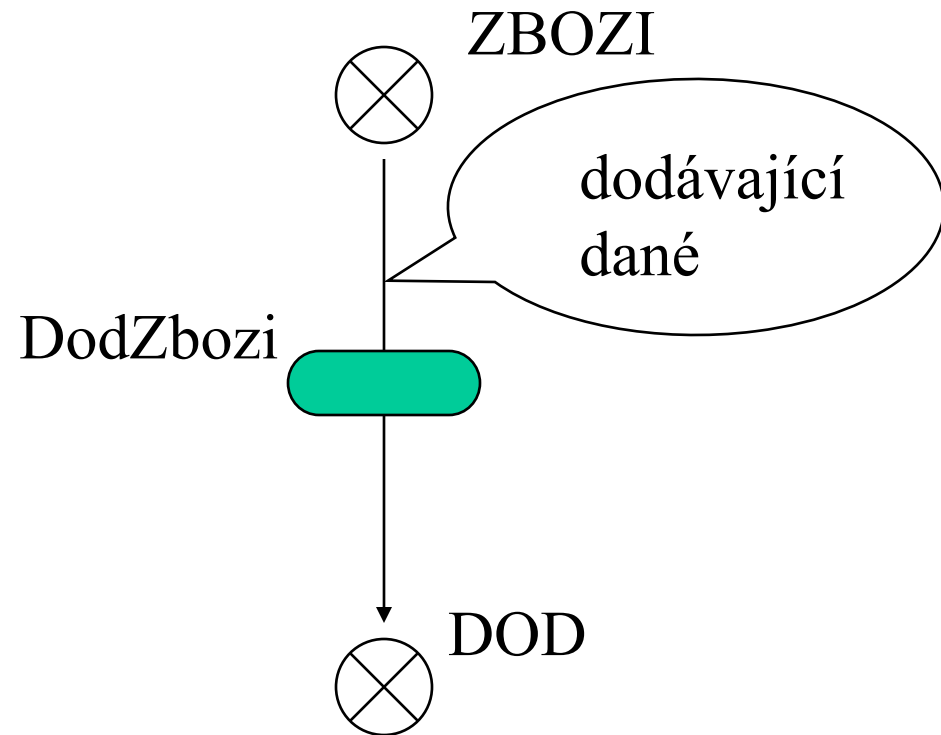
v mateřském
podniku

daného

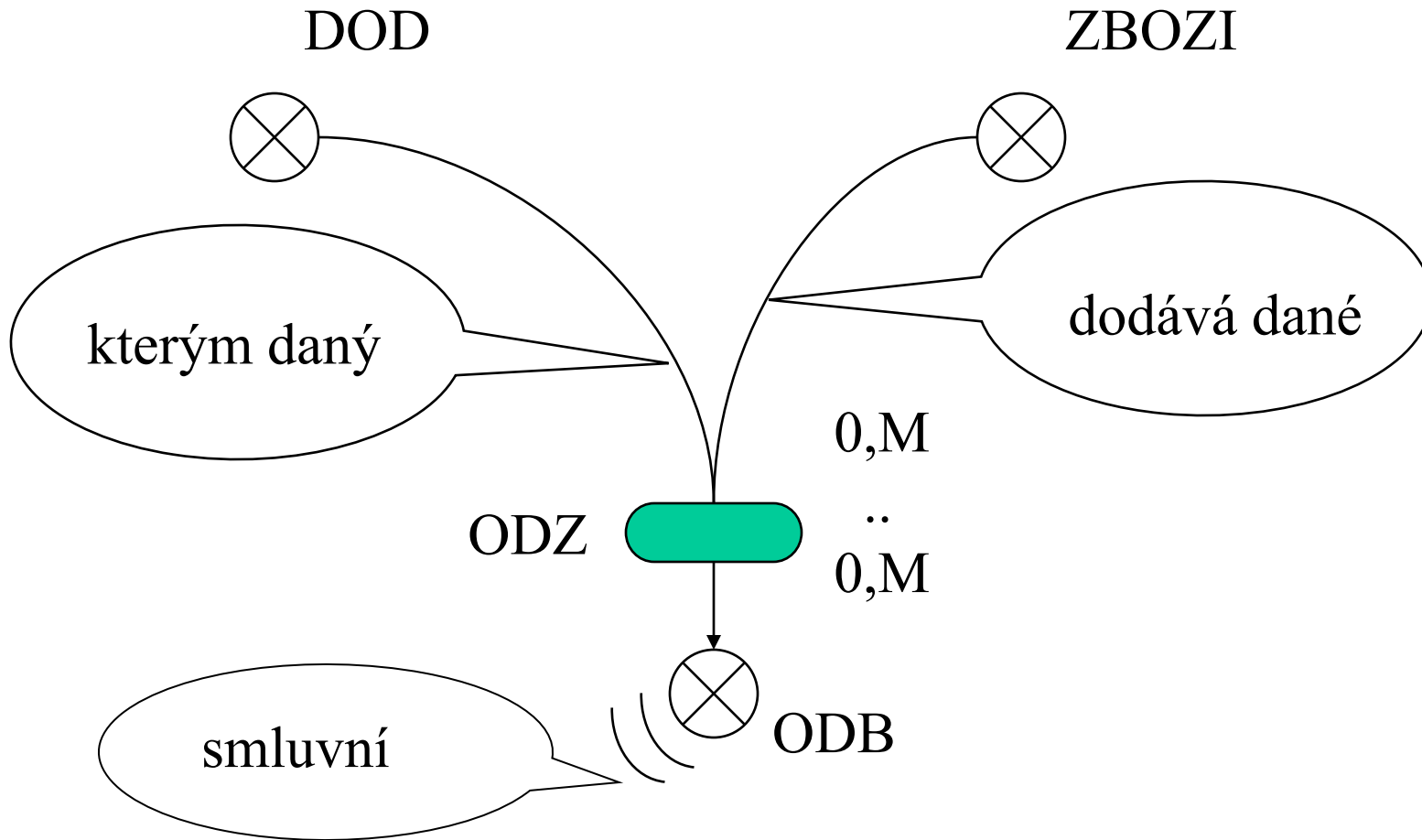
smlouvou
deklarovaný

- typ hodnoty funkce
- typ argumentů
- typ samotné funkce
- role argumentů
- sémantika přiřazení
- poměr

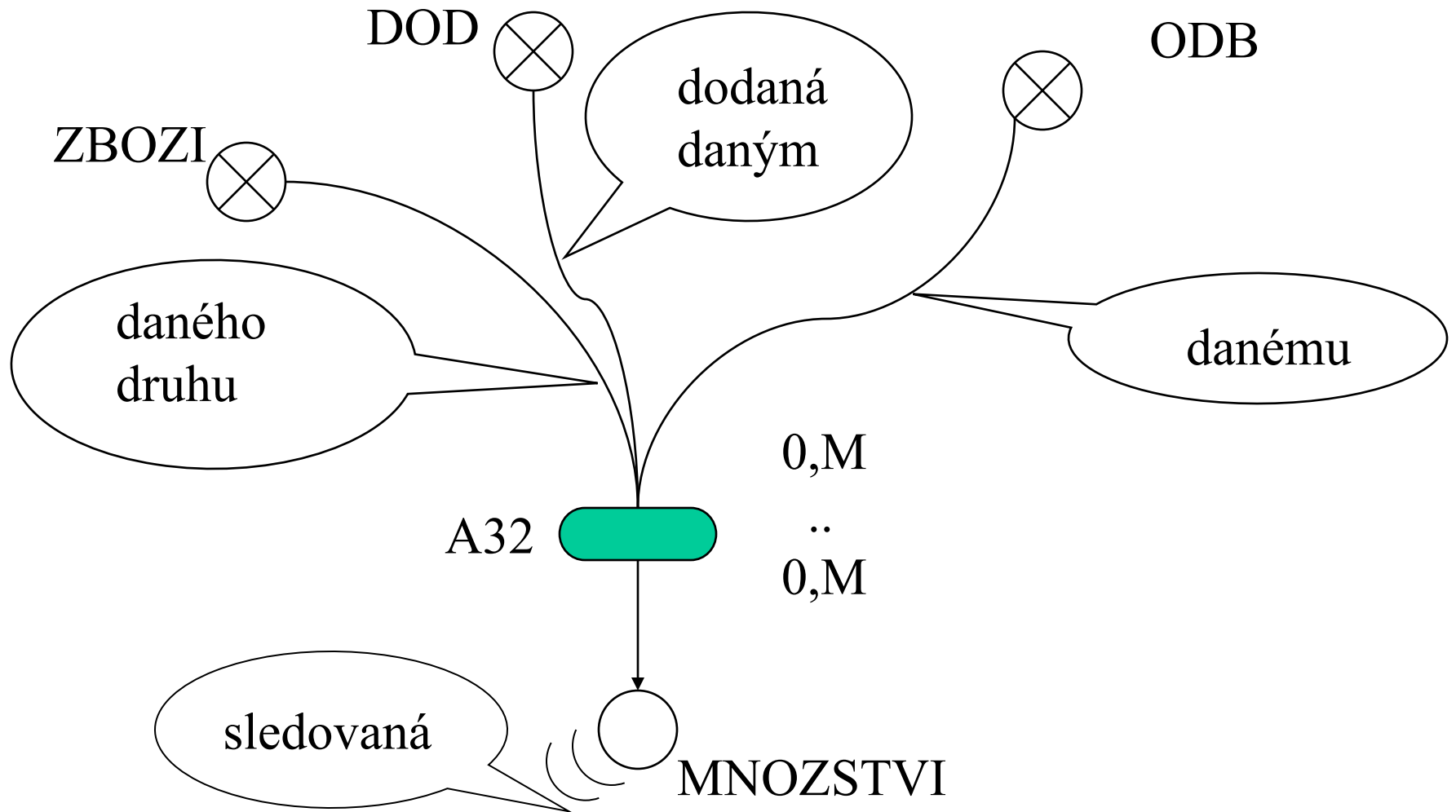
Příklady: co lépe vystihuje realitu



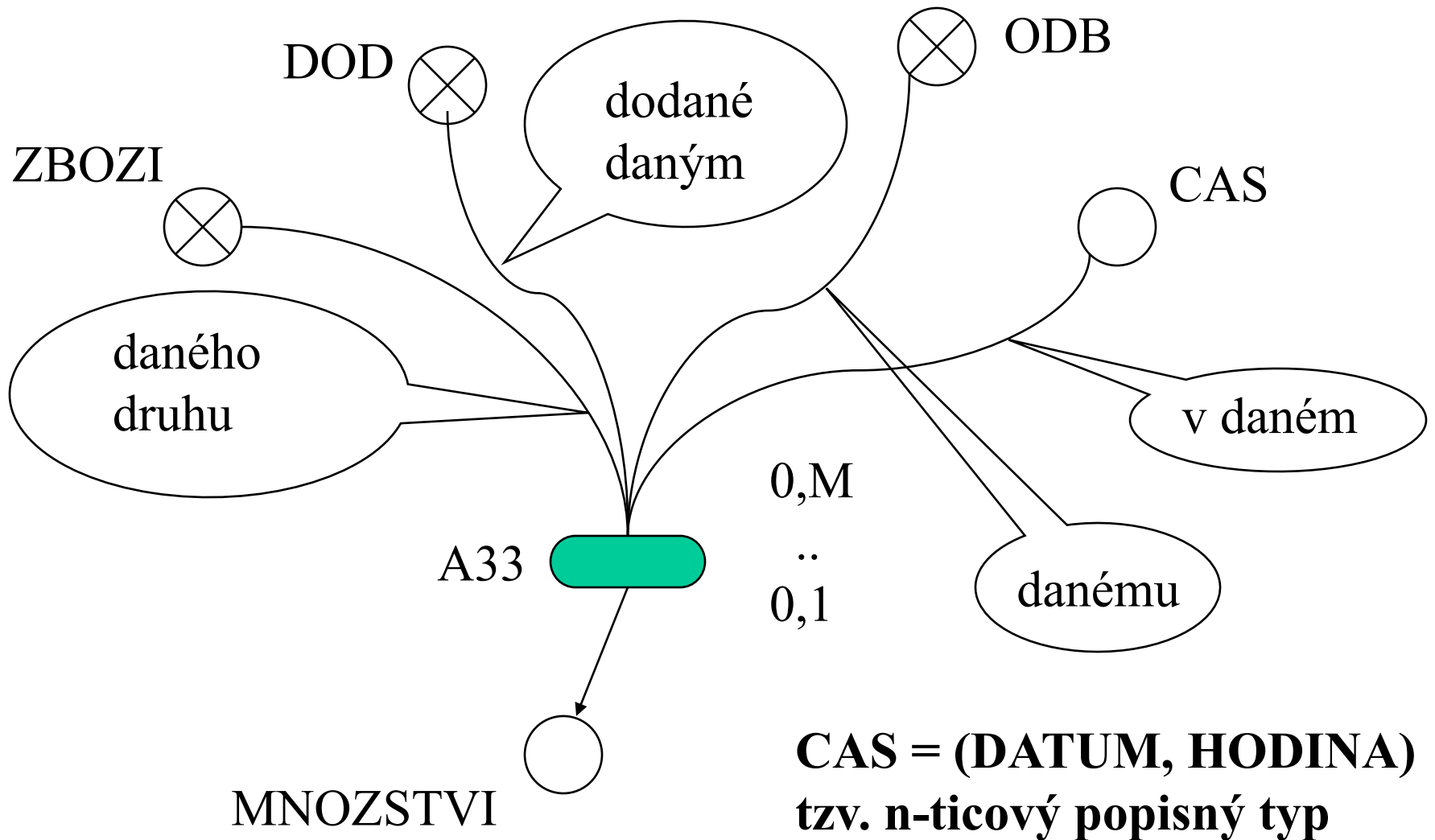
Příklady



Příklad: atribut (složitosti 4)



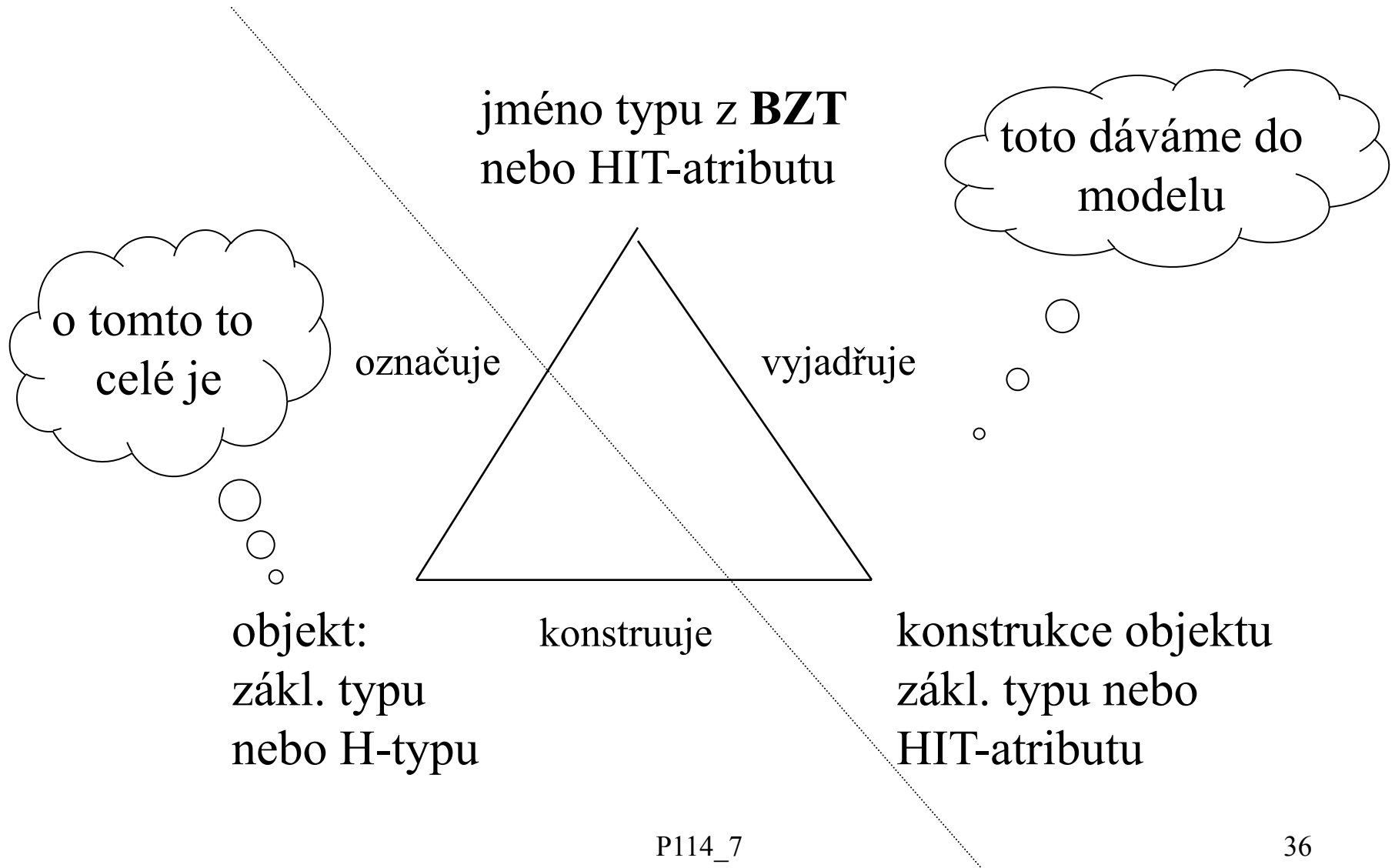
Příklad s chybkou:



Podstata modelování

- Složitost hierarchie typů nad **EB** a jejich konstrukcí je skryta v definicích uzlových typů patřících do **BZT**.
- Veškeré konstrukce, se kterými při práci s HIT-atributy pracujeme, jsou konstrukcemi jednoduchých typů (a) nebo (b) - viz přednáška 6.
- Toto odpovídá praxi implementace na počítačích: prvky **BS** (viz přednáška 6) reprezentujeme jako tabulky / soubory; konstrukce jednoduchých typů jako jednoznačné nebo mnohoznačné vztahy

KTO v HIT metodě



... a kvůli tomuto to celé děláme:

- aplikací prvků H-typů na *wt* dostaneme
...
- extenze HIT-atributů, které vyjadřují vztahy mezi konkrétními prvky sort (entitních či deskriptivních) z **BZT**
- ... čili to, co máme ve formě databázových tabulek uloženo v databázích

?

stav světa, datový model a databázový stav

- jaký je vztah datového modelu a aktuálního stavu světa?
- Databázový stav (DbSt): každá možná **populace sort**?
- ... nebo **naplnění kontejnerů** na prvky jednotlivých sort v dané databázi
- V databázi jsou ke každé sortě vždy jen konečné množiny
- Jak tedy definovat DbSt?
- Jaký je vztah DbSt a datového modelu?
- Jak se má DbSt k aktuálnímu světu?

?

aktualizace databáze a poznatelnost stavu světa

- Jak je možno popsat operaci aktualizace databáze?
- Podobnost s valuací?
- ... nebo je to stejné jako operace dotazu, jenom přehodíme co je vstupem a co je výstupem ...
- Je poznatelný aktuální svět?
- Zvýší se poznatelnost stavu světa zavedením informačního systému se „správně“ zkonstruovanou databází?
- V čem spočívá „přidaná hodnota“ databází ?