MOŽNOSTI VYUŽITÍ DEDUKTIVNÍCH PRAVIDEL PŘI ANALÝZE A NÁVRHU DATABÁZOVÝCH SYSTÉMŮ

(POSSIBILITY OF USING DEDUCTIVE RULES IN ANALYSE AND DESIGNING OF DATABASE SYSTEMS)

Zdeňka TELNAROVÁ
Ostravská univerzita, Přírodovědecká fakulta, Katedra informatiky a počítačů, 30.dubna 22, Ostrava 1, Česká republika
tel.: +4206160216, e-mail: telnaro@osu.cz

SUMMARY

Theoretical development in the area of databases systems has been keeping its dynamics. One of its directions is unfolding the conception of derived data. Data is a valuable resource whose quality, consistency and robustness is very important and essential. This paper focuses on using of deductive approaches to analysis and design of IS. This approach can minimise huge amount of data with not loosing information, which is advantage for maintenance of databases. Another area that is concerned with this paper is possibility to define integrity constraint with deductive rules and improve expressive power of database languages .

The paper deal mainly with deductive database specifics. Deductive databases are the outcome of the combination of logic programs and databases. They can potentially solve many problems in analysis, design, prototyping and implementation of databases and information systems. Deductive rules are in this paper demonstrated as a tool for derivation of classes/relations, derivation of attributes, views, for defining of integrity constraints. Deductive rules can be also useful for designing of query (this is not discussed topic of the paper). The paper is documented by a number of small examples which illustrate relation and object approach and which deal with classes, attributes and views. There are written in Chimera language.

Keywords: *Deductive rules, integrity constraint, stratification, deductive integrity constraint, relation approach, object approach.*

1. ÚVOD DO DEDUKTIVNÍCH PRAVIDEL

Deduktivní pravidlo je výraz ve tvaru:

Hlava ← Tělo,

kde hlava je atomická formule a tělo je libovolná formule. Každá proměnná v hlavě pravidla se musí vyskytovat rovněž v těle pravidla. Existují čtyři typy deduktivních pravidel dle toho, co je těmito pravidly definováno. Jedná se o pravidla definující třídy, pravidla definující pohledy, pravidla definující atributy a pravidla definující integritní omezení. K definování téhož konceptu, kdy konceptem je v této souvislosti rozuměna třída, pohled, atribut či integritní omezení, je možno použít několik pravidel. Veškeré příklady konceptů, uváděné v textu, jsou zapsány v jazyku Chimera, konceptuálním jazyku IDEA metodologie.
Např. pro definování platu je možno použít následujících deduktivních pravidel:

X.Plat = 15000 ← Inženýr(X), X.Věk < 35
X.Plat = 20000 ← Inženýr(X), X.Věk > =35
X.Plat = 25000 ← Manažér(X), Skupina(Y), Y in X.Vedoucí
X.Plat = 35000 ← Manažér(X), Oddělení(Y), Y in X.Ředitel

Stratifikace

Deduktivní pravidla mohou na sobě navzájem záviset různým způsobem, buď hierarchicky nebo rekurzivně. Tato závislost může být vyjádřena tvrzením, že koncepty odvoditelné na vyšší úrovni mohou být vyhodnoceny pouze tehdy, pokud koncepty na nižší úrovni, na nichž jsou tyto koncepty vyšší úrovně závislé, již byly dříve vyhodnoceny. Každá úroveň je označována „strata“ a celkové vytváření úrovní se nazývá stratifikace množiny pravidel. Tento proces musí být pečlivě kontrolován, pokud se v pravidlech objeví negace.
Negativní literály v deklarativních výrazech nemohou produkovat žádná nová tvrzení, ale mohou pouze potvrdit nebo vyvrátit tvrzení vytvořená v průběhu vyhodnocení jejich pozitivních výskytů se stejnými proměnnými. Před vyhodnocením negace je třeba znát všechna pozitivní fakta, ze kterých může být vyjádřen závěr ve formě negace. Máme-li např. definována tvrzení kdo je koho potomkem, pak pouze jejich plný výčet umožňuje potvrdit či vyvrátit tvrzení kdo koho potomkem není. Stratifikace nemůže být dosaženo, pokud existuje odvozený koncept, který závisí jak rekurzivně, tak negativně na sobě samém. Množina pravidel obsahující takovýto rekurzivní cyklus, který zahrnuje negace, se nazývá nestratifikovatelná.

Např.
Válečník(X) ← not Pacifista(X)
Pacifista(X) ← not Válečník(X)

Obě pravidla je třeba umístit na stejnou úroveň, protože jsou rekurzivně závislá. Zároveň by však měla být umístěna do různých úrovní, protože jeden na druhém negativně závisí. Taková situace je označována jako logický paradox a musí jí být zabráněno.
Další paradoxní situace může nastat, pokud k odvození konceptu je třeba vypočítat agregační funkci, jejímž elementem je odvozovaný koncept. Tuto situaci uveďme na příkladu:

X in Y.StejnýPříjem ← Zaměstnanec(X), Zaměstnanec(Y), X.Plat = avg(Y.StejnýPříjem.Plat)

Zaměstnanec X má StejnýPříjem jako Zaměstnanec Y, jestliže X má Plat rovný aritmetickému průměru Platů Zaměstnanců se StejnýmPříjmem jako Y. Takovéto rekurzivní cykly zahrnující agregační funkce musí být rovněž vyloučeny.

Stratifikovaná množina pravidel je pak množina pravidel, která vylučuje negační a agregační paradox.

Všeobecné principy dobře formovaných deduktivních pravidel
Pravidla musí splňovat několik syntaktických a sémantických požadavků, aby mohla být vyhodnocena. Tyto požadavky lze shrnout do následujících bodů:

- Pravidla musí být v omezeném rozsahu, tzn. všechny jejich proměnné musí být omezeny vhodnými formulemi.
- Pravidla musí být bezpečná, tzn. všechny proměnné v hlavě pravidla se musí objevit minimálně v jedné pozitivní formuli v těle pravidla.
- Pravidla s negativními formulemi musí být stratifikována, tzn. nesmí být rekurzivně použita v těle jiných pravidel tak, aby vytvářela cyklus rekurzivních pravidel, která vyvolávají negativní formuli.
- Pravidla s termy musí být stratifikována, tzn. že nesmí být rekurzivně použita v těle ostatních pravidel tak, aby tvořila cyklus rekurzivních pravidel, která volají množinu termů.

2. ODVOZOVÁNÍ DAT

Deduktivní pravidla mohou být využita pro odvozování dat a pro definování integritních omezení. Pravidla pro odvozování dat mohou odvozovat atributy, pohledy a třídy. Integritní omezení jsou chápána jako implicitní (ve fixním formátu) nebo jako uživatel definovaná (v generickém formátu). Návrh deduktivních pravidel je iniciován v průběhu analýzy. Během návrhu může být objeveno mnoho dalších požadavků. Návrhář může definovat všechna odvozená data, která obohatí znalosti podporované systémem a odhalí všechny zdroje kontrol konzistence, které mohou být formulovány jako integritní omezení. Velkou výhodou deduktivních pravidel je jejich deklarativnost.

Deklarativnost představuje popis znalostí, který nevyžaduje žádné procedurální detaily a tudíž je naprosto nezávislý na implementaci.

Na druhé straně z toho vyplývají i omezení v nemožnosti vyjádřit procedurální rysy, které jsou někdy zapotřebí. Jsou-li deduktivní pravidla pro odvozování dat transformována do aktivních pravidel, dochází k tzv. materializaci dat. V takovém případě jsou odvozená data extenzionálně uložena v databázi.
Jazyky pro definování pravidel (Prolog nebo Datalog) jsou používány v kontextu relačních databází. V případě použití objektově orientovaných databází, objektová reprezentace vede k různým stylům pravidel. Proměnné odkazují na objekty spíše než na hodnoty a přístup k jednotlivých hodnotám atributů se realizuje prostřednictvím tečkové notace.

2.1. Příklady implementace v jazyku Chimera

Odvozování tříd
Využitím deduktivních pravidel lze odvodit další třídy, jejichž populace závisí na hodnotě nějakého atributu definované supertřídy. Populace mohou být odvozeny pouze pro subtřídy pomocí pravidel, která jsou definována v supertřídách.

Např.
Define implementation for Zaměstnanec
Population Zaměstnanec(X) ← Osoba(X),
X.Profese = „zaměstnanec“
End

Tímto způsobem lze odvodit třídu Zaměstnanec ze supertřídy Osoba, použitím deduktivního pravidla. Jestliže Osoba změní Profesi, musí být automaticky vyňata ze třídy Zaměstnanec.

Odvozování pohledů
Pohledy jsou chápány jako virtuální objekty, které vzniknou vyhodnocením dotazu a tím usnadňují tvorbu aplikace. K pohledům lze nastavit individuální přístup pro aplikace, což umožňuje vytvářet mechanizmus ochran. Data jsou vybírána z více tříd a k těmto datům je nastaven selektivní přístup.

Např.
Define view Manželé(Manžel:Oosoba,
Manželka:Osoba)

Manželé(X,Y) ← X.Manžel = Y, X.Sex = "žena"
End

Odvozování atributů
Jednou z důležitých možností využití deduktivních pravidel je výpočet odvozených atributů, tj. takových atributů, jejichž hodnota může být odvozena z hodnot jiných atributů. Návrhář by měl odpovědět na následující otázku: *Existují nějaké atributy, jejichž hodnoty mohou být odvozeny?* Příklad odvozeného atributu by mohl být následující:
Mějme atribut Datum_narození u třídy Osoba. Z atributu Datum narození odvoďme atribut Věk.

```
Define object class Osoba
        Attributes
        Datum_narození:Date,
        Věk:integer derived
End

Define external formula castToVěk(in D:date, out
A:integer)
End

Define implementation for Osoba
        Attributes Self.Věk = I := integer(I),
        castToVěk(Self.Datum_narození, I)
End
```

Speciální pozornost je třeba věnovat rekurzívním atributům. Nejjednodušší a nejpoužívanější forma rekurze je tranzitivní uzávěr binárního vztahu. Vztah je modelován tak, že typ atributu objektu odkazuje zpět na tento objekt. Tento typ vztahu bývá nazýván kruhovým vztahem.

Mějme následující fragment schématu databáze :

```
Define object class Osoba
        Attributes
        Děti: set-of(Osoba),
        Rodiče:set-of(Osoba), derived,
        Sourozenci:set-of(Osoba), derived,
        Předci:set-of(Osoba), derived
End
```

Využitím deduktivních pravidel můžeme navrhnout odvozené rekurzivní atributy Rodiče, Sourozenci, Předci a Bratranci např. následujícím způsobem:

```
Define implementation for Osoba
        Attributes
        X in Self.Rodiče ← Osoba(X), Self in
        X.Děti;
        X in Self.Sourozenci ← Osoba(X),
        X.Rodiče in Self.Rodiče,
        X != Self;
        X in Self.Předci ← Osoba(X), X in
        Self.Rodiče;
        X in Self.Předci ← Osoba(X), X in
        Self.Rodiče.Předci)
End
```

Za odvozené atributy budeme považovat atributy označené „derived“, které jsou odvozeny na základě deduktivního pravidla, které může být rekurzivní.

2.2. Srovnání relačního a objektového pojetí

Relační pojetí faktů
Fakta v deduktivně relační technologii jsou reprezentována relacemi, které jsou chápány jako atomické formule (logická reprezentace znalostí).

Např.
Osoba(Jméno, Adresa, Pohlaví)
Rodič(Rodič, Dítě)
Partner(Manžel, Manželka)

Objektové pojetí faktů
Fakta jsou v deduktivně objektové technologie reprezentována jako třídy objektů (rámcová – frame-reprezentace znalostí).

Např.
Object Osoba s atributy
        Jméno:string
        Adresa:string
        Sex:string
        Dítě: set-of Osoba
        Partner: Osoba

Deduktivními pravidly můžeme odvozovat pohledy. Pohledy jsou podobné relacím, ale jejich n-tice nejsou uloženy v databázi. Mohou se vypočítat pomocí dotazovacího jazyka nebo deduktivních pravidel. V objektovém pohledu deduktivní pravidla mohou odvozovat pohledy nebo atributy.

Pohledy pro relační styl
Matka(Matka, Dítě)
Sestra(Jméno, Jméno)
Předek(Jméno, Jméno)

Odvozené atributy pro objektový styl
Rodič: set-of(Osoba)
Matka:Osoba
Sestra: set-of(Osoba)
Předek:set-of(Osoba)

Všechna tato odvozená fakta můžeme vyjádřit pomocí pravidel.

V relačním stylu
Matka(X,Y) ← Rodič(X,Y), Osoba(X,-,žena)
(*matka je rodič, žena)*

Sestra(X,Y) ← Rodič(Z,X), Rodič(Z,Y),Osoba(X,-,žena)
(sestra je žena, která má stejné rodiče jako osoba Y)

Předek(X,Y) ← Rodič(X,Y)
(*předek je rodič)*

Předek(X,Y) ← Rodič(X,Y), Předek(X,Y)
(*předek je rodič předka)*

V objektovém stylu
X in Y.Rodič ← Osoba(X), Osoba(Y), Y in X.Dítě
(X je rodičem Y, jestliže X aY jsou osoby a zároveň Y je dítětem X)

X in Y.Matka ← Osoba(x), Osoba(Y), X in Y.Rodič, X.Sex="žena"
(X je matkou Y, jestliže X a Y jsou osoby, X je rodičem Y a zároveň X je žena)

X in Y.Sestra ← Osoba(X), Osoba(Y), Osoba(Z), Z in Y.Rodič, Z in X.Rodič, X.Sex ="žena"
(X je sestrou Y, jestliže existuje osoba Z, která je rodičem X i Y a zároveň X je žena)

X in Y.Předek ← Osoba(X), Osoba(Y), X in Y.Rodič
X in Y.Předek ← Osoba(X), Osoba(Y), X in Y.Rodič.Předek
(X je předkem Y, je-li jeho rodičem. X je předkem Y, je-li rodičem předka.)

Na tomto příkladu je ukázáno, jak může být deduktivních pravidel použito k vyjádření inverzních vztahů (Rodič, Dítě). Nejfrekventovanější použití rekurzívních pravidel je pro vyjádření tranzitivního uzávěru binárních vztahů. V uvedeném případě pohled Předek definujeme jako tranzitivní uzávěr relace Rodič. Tranzitivní uzávěr vyžaduje dvě pravidla, startovací pravidlo (nerekurzivní), které zabezpečuje bázi rekurze a rekurzivní pravidlo, které zabezpečuje indukci. Objektový styl se vyhne zavádění existenčně kvantifikované proměnné Z (v relačním stylu) vyjádřením cesty, která kombinuje atributy Rodič a Předek (Y.Rodič.Předek)

3. DEFINOVÁNÍ INTEGRITNÍCH OMEZENÍ

Integritní omezení jsou vyjadřována pomocí deduktivních pravidel, přičemž databázi vyhovuje dané integritní omezení, je-li hodnota pravidla „false". Databáze není konzistentní, je-li hodnota deduktivního pravidla „true".

Integritní omezení je výraz ve tvaru:

Integritní_omezení_jméno ← formule,

kde formule se skládá z relací resp. tříd.

Integritní omezení můžeme rozdělit na omezení bezprostřední a omezení s odloženým vyhodnocením. Kontrola omezení s odloženým vyhodnocením je prováděna po potvrzení transakce, tzn. nejdříve je provedena celá transakce (např. aktualizace dat) a pak je kontrolována integrita dat. U bezprostřední kontroly je kontrola provedena po každé specifické operaci (nazývané řádek transakce).
V obou případech, je sémantika kontroly omezení následující. Jestliže po výpočtu příslušného deduktivního pravidla dojde k produkci hodnot do hlavy pravidla, tyto hodnoty jsou chápány jako nepovolená data a transakce je zrušena. Operace, prováděné transakcí, jsou zrušeny a databáze je uvedena do stavu před započetím transakce.

3.1. Vestavěná a generická integritní omezení

Další možné členění integritních omezení je na IO vestavěná (ve fixním formátu, implicitní, build in) a IO generická.

Vestavěná IO
Lze rozlišovat následující vestavěná integritní omezení:

- ***Not Null*** atributy jsou atributy, které musí být známy, tzn. nesmí nabývat hodnoty Null.
- ***Primary key*** je kolekce atributů, která jednoznačně identifikuje každou instanci třídy.
- ***Unikátní atribut*** je kolekce atributů, jejichž hodnota je unikátní pro každou instanci třídy.
- ***Kardinalita*** je integritní omezení, které určuje počet objektů, které mohou být asociovány ve vztahu.

Kardinalita může být vyjádřena:

- min, max – minimální, resp. maximální počet objektů vstupujících do vztahu
- *, nula nebo více objektů vstupuje do vztahu.
- +, jeden nebo více objektů vstupuje do vztahu.

Referenční integrita je omezení binárního vztahu v tom smyslu, že jedna třída je označena jako tzv. „odkazující" a druhá jako tzv. „odkazovaná". Pro odkazovanou třídu to znamená, že nemůže v rámci této třídy existovat objekt, který by nevstupoval do vztahu s objektem třídy odkazující. Standardní způsoby zachování referenční integrity jsou "Restrict" a "Cascade", v některých případech se využívá "Set Null".

3.2. Generické integritní omezení

Konvenční databázové systémy podporují výše uvedená integritní omezení, která jsou rovněž nazývána integritní omezení ve fixním formátu. Existují i další integritní omezení, v tzn. generickém formátu, která závisejí na specifikované aplikační doméně. Tato omezení mohou být podporována

v jejich plné obecnosti pouze novými generacemi databázových systémů.

Generická integritní omezení mají dva rozměry:
Statická a dynamická omezení
Statická omezení mohou být vyhodnocena v jednoduchém stavu databáze, tzv. v kontextu objektů a vztahů. Tato omezení dále můžeme rozdělit na omezení bezprostřední a s odloženým vyhodnocením (viz výše).
Dynamická omezení srovnávají dva stavy databáze, nazývané inicializační a koncový stav a jsou navrhována pro monitorování změn stavů. Rovněž dynamické omezení můžeme rozdělit na omezení s bezprostředním a odloženým vyhodnocením.

Rozsah omezení
Omezení, definovaná v kontextu specifikovaných tříd, jsou nazývána cílená omezení. Jedná se o predikát, který dohlíží na stav individuálních cílových objektů a poskytuje pro každý cílový objekt specifickou pravdivostní hodnotu. Predikát může taky odkazovat na ostatní objekty, které jsou spojeny vztahem s cílovým objektem.
Např. Zaměstnanec jde do důchodu, pokud má věk větší než 60 let nebo má odpracováno více než 35 let.
Všechna ostatní omezení jsou nazývána necílená omezení.
Např. Student musí udělat zkoušky Kurzů, které navštěvuje. Generická integritní omezení jsou tvořena ve fázi návrhu, kdy jsou transformována do deduktivních pravidel.

ZÁVĚR
Deduktivní databáze, které jsou založeny na deduktivních pravidlech, spolu s objektovými, aktivními, temporálními, textovými a multimediálními databázemi zaznamenávají v současné době rychlý pokrok a to především proto, že jsou schopny věrněji modelovat realitu, efektivněji zpracovávat komplexní data a lépe zajišťovat nezávislost dat jak na uživatelských aplikacích, tak na jejich implementaci. V souvislosti s deduktivními databázemi, potažmo s deduktivními pravidly, hovoříme o tzv. znalostní nezávislosti, která je dalším krokem k flexibilitě databází a jejich aplikací. Proto je problematika analýzy a návrhu IS s využitím deduktivních přístupů jistě aktuálním tématem.

LITERATURA
[1] Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases, Addison-Wesley Pubs.Co, 1995
[2] Ceri, S., Fraternali, P.: Designing Database Applications with Objects and Rules, Addison-Wesley, 1997
[3] Telnarová, Z.: Metodická doporučení k integraci deduktivních pravidel do analýzy a návrhu IS s databází, Ostrava, 2001
[4] Ullman, D. J.: Principles of database and knowledge-base systems, Volume I, II, Computer Sc. Press, 1988
[5] Wagner, G.: Foundations of Knowledge Systems with Applications to Databases and Agents, Kluwer Academic Publishers, 1998
[6] Zaniolo, C. , Ceri, S. , Faloutsos, Ch. , Snodgrass, R.T., Subrahmanian, V.S. , Zicari, R.: Advanced Database Systems, Morgan Kaufmann Publishers, Inc., 1997

BIOGRAPHY
Ing.Zdeňka Telnarová, Ph.D. graduated in System Engineering at VSB-TU, Ostrava. Thesis: Analysis and classification of multidimensional data. She obtained pedagogical qualification in postgraduate study in Pedagogical Faculty at University of Ostrava. Thesis: ICT at Teaching Education. She received Ph.D. degree in Informatics at VSB-TU, Ostrava. Theses: Methodical recommendation for integration of deductive rules into analysis and design of information systems.
Since 1994 she has been working as a lecturer in the Faculty of Science at University of Ostrava. His scientific research is focusing on the database modelling and database techniques.