

DEDUKTIVNÍ PRAVIDLA V ANALÝZE A NÁVRHU DATABÁZOVÝCH APLIKACÍ

Zdeňka Telnarová

Ostravská univerzita, Přírodovědecká fakulta, Katedra Informatiky a počítačů, 30.dubna 22, 713 00 Ostrava 1, Zdenka.Telnarova@osu.cz

Abstrakt

Príspevek je zaměřen na základní seznámení se s problematikou deduktivních databází, založených na deduktivních pravidlech. Je diskutován jak relační datový model, tak model objektový a u obou modelů je naznačen způsob implementace deduktivních pravidel. Příklady v relačním pojetí jsou zapsány v Datalogu, v objektovém pojetí v jazyku Chimera a týkají se odvozování dat z extenzionální databáze. Okrajově je rovněž zmíněna problematika integritních omezení.

1. Úvod

Jedním ze základních paradigmat databázové technologie je tzv. datová nezávislost, která se odráží jak v nezávislosti dat na jejich fyzické implementaci, tak v nezávislosti dat na aplikacích, které s těmito daty manipulují. Deduktivní databáze, založené na deduktivních pravidlech, jsou dalším krokem k nezávislosti dat a to k znalostní. Pojem znalostní nezávislosti byl poprvé popsán Friesenem v roce 1994 a jeho podstata spočívá v tom, že znalost (odvoditelný fakt) není součástí programu (databázové aplikace), ale je součástí báze dat.

2. Deduktivní databáze

Deduktivní databáze jsou založeny na podpoře teorie dokazování a jsou schopné deduce dodatečných faktů z extenzionální databáze. Mají zabudovány specifikované deduktivní axiomy a pravidla dedukce. Deduktivní axiomy společně s integritními omezeními jsou obvykle označovány jako intenzionální databáze. Deduktivní databáze se tedy skládá ze dvou složek, extenzionální databáze (EDB) a intenzionální databáze (IDB).

Vezmeme-li za základ relační datový model, deduktivní přístup rozšiřuje tento model (jak známo založený na jediném konstrukt, nazývaném databázová relace) tím, že využívá dvou typů relací. Základní relace jsou uloženy v databázi (EDB) a odpovídají relacím v relačním datovém modelu. Odvozené relace nemusí být uloženy v databázi. Jsou obvykle dočasné a uchovávají mezivýsledky operací. Deduktivní pravidla, která tyto relace odvozují, společně s integritními omezeními, jsou označována jako IDB. Jazyk SQL a obdobné jazyky založené na relační algebře a relačním kalkulu nemají dostatečnou vyjadřovací sílu a nejsou schopny vyjádřit uzávěr tranzitivních závislostí v relačním datovém modelu. Logická pravidla, využívající funkčních symbolů, mohou vyjádřit jakýkoliv výpočet tak, že může být zapsán v konvenčním programovacím jazyku. Dokonce logická pravidla bez funkčních symbolů (např. jazyk DATALOG) mají sílu vyjádřit výpočet větší než konvenční DQL (Data Query Language).

Deduktivní databáze, mohou být rozšířením relačních databází, ale rovněž se velmi efektivně dají kombinovat s databázemi objektovými. Extenzionální databázi pak můžeme rozumět množinu objektů, intenzionální databázi množinu deduktivních pravidel, které jsou implementovány jako metody tříd.

3. Deduktivní pravidlo

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 je možno použít několik pravidel.

Např. pro definování platu je možno použít následujících deduktivních pravidel:

$X.\text{plat} = 10000 \leftarrow \text{inženýr}(X), X.\text{věk} < 35$

$X.\text{plat} = 15000 \leftarrow \text{inženýr}(X), X.\text{věk} \geq 35$

$X.\text{plat} = 25000 \leftarrow \text{manažér}(X), \text{skupina}(Y), Y \text{ in } X.\text{vedoucí}$

$X.\text{plat} = 35000 \leftarrow \text{manažér}(X), \text{oddělení}(Y), Y \text{ in } X.\text{ředitel}$

3.1 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.

$\text{poslanec}(X) \leftarrow \text{not podnikatel}(X)$

$\text{podnikatel}(X) \leftarrow \text{not poslanec}(X)$

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

3.2 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 formulami 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 použity v těle ostatních pravidel tak, aby tvořila cyklus rekurzivních pravidel, které volají množinu termů.

4. Využití deduktivních pravidel pro odvozování dat

Deduktivní pravidla mohou být rozdělena na pravidla pro odvozování dat a pravidla pro integritní omezení. Pravidla pro odvozování dat odvozují atributy, pohledy a odvozené třídy. Integritní omezení jsou chápána jako implicitní (fixní formát) nebo jako uživatel definovaná (generic format). 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 umožňuje popis znalostí, které nevyžadují žádné procedurální detaily a tudíž je naprosto nezávislý popis pravidel od jejich implementace.

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.

4.1 Příklady implementace v jazyku Chimera

4.1.1 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 - zaměstnanec, musí být automaticky vyňata z třídy Zaměstnanec.

4.1.2 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 mechanismus ochrany. Data jsou vybírána z více tříd a k těmto datům je nastaven selektivní přístup.

Např.

```
Define view Manželé(Manžel:Osoba, Manželka:Osoba)
    Manželé(X,Y) ← X.Manžel = Y, X.Sex = žena
End
```

4.1.3 Odvozování atributů

Jednou z důležitých aplikací 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 odvodíme atribut Věk.

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

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

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

Speciální pozornost je třeba věnovat rekurzivním 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 with
  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 Bratrance např. následujícím způsobem:

```
Define implementation for Osoba with
  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.

4.2 Srovnání relačního a objektového pojetí

4.2.1 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(Jméno_rodice, Jméno_dítěte)

Partner(Jméno_manžela, Jméno_manželky)

4.2.2 Objektové pojetí faktů

Fakta jsou v deduktivně objektové technologii 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.

4.2.3 Pohledy pro relační styl

Matka(Jméno_rodice, Jméno_dítěte)
Sestra(Jméno_rodice, Jméno_dítěte)
Předek(Jméno_předka, Jméno_potomka)

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

4.2.4 Odvozené atributy pro objektový styl

Rodič: set-of(Osoba)
Matka:Osoba
Sestra: set-of(Osoba)
Předek:set-of(Osoba)

X in Y.Rodič ← Osoba(X), Osoba(Y), Y in X.Dítě
X je Rodičem Y, jestliže X a Y 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 Z, 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
Tímto způsobem je vyjádřena rekurze. 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 rekurzivních pravidel je pro vyjádření tranzitivního uzávěru binárních vztahů. V uvedeném případě pohled Předek definujeme jako tranzitivní uzávěr relace(třídy) Rodič. Tranzitivní uzávěr vyžaduje dvě pravidla, startovací pravidlo (nerekurzivní) - zabezpečující bázi rekurze a rekurzivní pravidlo-zabezpečující indukci. Objektový styl se vyhne zavádění existenčně kvantifikované proměnné Z (v relačním stylu) vyjádřením cesty, která kombinuje atributy Rodič a Předek (Y.Rodič.Předek).

5. Využití deduktivních pravidel pro 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:

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 ukončení 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, sémantika kontroly omezení je stejná. 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, které byly transakcí provedeny, jsou zrušeny a databáze je uvedena do stavu před započítáním transakce.

5.1 Vestavěná a generická integritní omezení

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

5.1.1 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, které jednoznačně identifikují 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.
- 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“.

5.1.2 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.

Dynamika generických integritních 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 s bezprostřední a odloženým vyhodnocením.

Rozsah generických integritních 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.

6. Závěr

Analýza, návrh a implementace informačních systémů je složitý proces, realizovaný formou týmové práce. Informační systém je vysoce kreativní produkt, při jehož tvorbě, jako u všech tvůrčích procesů, hraje velkou roli lidská invence. Je v podstatě nemožné vytvořit informační systém, který by splňoval veškeré požadavky a neměl žádná úskalí a chyby. Uvedená skutečnost je důvodem toho, že již řadu let existuje snaha o vytvoření systematických metod a technik, které by činnosti spojené s tvorbou IS usnadnily a byly návodem pro jejich tvůrce. V devadesátých letech vznikly metodologie, které vycházejí z objektové orientace, a které charakterizuje společný přístup zaměřený na tzv. objektový a dynamický model. V posledních letech jsme svědky propojování objektových přístupů s přístupy deduktivními, které již prošly svým vývojem v souvislosti s možnými rozšířeními relačních databázových technologií. Základní ideou deduktivních databází je rozšíření expresivity databázových jazyků se současným zachováním jejich deklarativnosti. Zajímavý problém v této souvislosti je, jaké vzory chování lze dedukovat z celkem jednoduchých informací uložených v databázi. Zavedení deduktivních pravidel přináší dvě základní výhody. Jsou jimi nové prostředky pro zachování integrity dat a možnost odvozování nových dat (hlavně využitím rekurzivních pravidel). Spojení objektových a deduktivních přístupů v procesu tvorby informačního systému se jeví v mnoha ohledech přínosné a je dalším příspěvkem k nezávislosti dat.

Literatura:

1. G. Wagner: Foundations of Knowledge Systems with Applications to Databases and Agents, Kluwer Academic Publishers, 1998
2. S. Ceri, P. Fraternali: Designing Database Applications with Objects and Rules, Addison-Wesley, 1997
3. C. Zaniolo, S. Ceri, Ch. Faloutsos, R.T. Snodgrass, V.S. Subrahmanian, R. Zicari: Advanced Database Systems, Morgan Kaufmann Publishers, Inc., 1997
4. Z. Telnarová: Metodická doporučení k integraci deduktivních pravidel do analýzy a návrhu IS s databází, Ostravská univerzita, 2001