

Temporálne databáze

1

- DB s časovou dimension
- obsah sa mení v čase a my túto zmenu chceme zaznamenávať
-
- **čas platnosti** - od kedy do kedy sú údaje platné
 - systém zaznamenáva platnosť sámu
 - mysl. select bude agragovať výsledok (autom. obsahuje aj záznamy o rôznych časových obdobiaci)
 - ↳ číše od do platilo toto, teraz plati toto
- **čas transakcie** - čas, kedy je údaj prítomný v DB a môže byť získaný
 - ↳ slúži na zaznamenávanie zmen dát v histórii - bezpečnosť a kontrola konsistencie + je to čas kedy bol fakticky uložený v DB
- **časový ohannik** - bod na časovej osi
- **časový úsek** - doba medzi 2 časovými ohannikmi
- **časové tracie** - časový úsek so základom dátou ale nesmeštnym začiatkom a koncom
 - najmenšia jednotka merania hodín sú **chronomy**
- **časový (temporálny) element** je konečné siedemkrát ľahšie úsekov

Tabuľky v TDB

- minulosť
- platného času
- transakcie
- obojakoľko času (tab. čas platnosti + čas transakcie)

Problém TDB

- ↳ preodlivelosť rôznych modelov - môže sa stať, že 5.1.2000 som ponielal niekam dopis, ale dnes už tá adresa neexistuje.
- ↳ funkčné závislosti, normálne formy (nebudeme riešiť)

Stavebne' kameňe TDB

- časová doména $(T, <)$ - lineárne usporiadana', neobmedzena'
- relácia Δ DB $DB(D, \rho)$ nad $(D, =)$ a schéma ρ

$$\hookrightarrow (D, =, r_1, \dots, r_n)$$

 ρ

\hookrightarrow koncretná instance nad D

doména
interpretovaných
konštant

Treba spojiť napl. 2 veci (kameňe):

- Snímky DB (snapshots)
- čas platnosti (timestamp)

\hookrightarrow Relačná DB nemá pamäť - práve teraz, uklada' vždy aktuálne dátá, ale nemá prehľad o tom, kedy čo bolo platné.

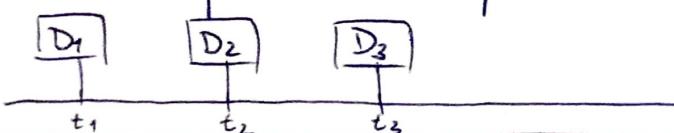
1.) Snímka' tabuľka

- \hookrightarrow pri každej zmene sa uloží nový snapshot DB, keďže potom platí
- \hookrightarrow časy transakcie a platnosti sú zachytené tiež (platnosť faktu odrodine z trvania faktu cez niekoľko snapshotov)
- \hookrightarrow je doby' pre štatistické dolazy - vyberieme 1 snapshot a spravime nad ním nepále selecty / štatistiky

• analógia s menovkami na dverach

- 1990 - aristent
- 1995 - odr. aristent
- 2000 - docent

- snapshot má blízko k temporalnej logike (TL)
- každá DB popisuje stav sveta v konkretnom časovom okamihu
- relácia usporiadania potom definuje tok času.



Nevýhoda snapshot modelu

- nie je vhodný pre dokazy typu $\{t : DB \models \varphi(t)\}$
- aby sme zistili kedy bola podmienka φ splnena tak musíme prejsť všetky snapshots, či φ bola splnena

2.) Transakčná tabuľka

- iba pripojenie dát
- transakcie, roll-back v dolasoch
- napr. vyplativa páša
- transakcie verzia na zostavanie histórie, väčšinu vektorovia k aby zmeny došlo
- ratične s tým, že pri ~~pre~~ transakcii môže dojst' k chybám

3.) Tabuľka s platujúmi časmi

- možnosť modifikácie
- v každom riadku informácie od kedy do kedy platila
- pre aktuálne dátu je od istého času doleras
- umožňuje dokazy na históriu
- napr. prehľad zamestnaní v CV
 \hookrightarrow odvtedy dovedeš som pracoval na firme A

Model s časom platnosti

- pre každé $r_i \in \rho$ definujeme reláciu R_i tak, že

$$R_i = \{(t, a_1, \dots, a_n) : (a_1, \dots, a_n) \in r_i \text{ in } R\}$$

\downarrow
v každom riadku

pripojený časový údaj

- štruktúra TDB s časovými pečiatkami (timestamp)

$$(D, T, \leq, r_1, \dots, r_n) \rightarrow \text{konečná instance } \rho \text{ nad } D, T$$

konečná inst. ρ nad D \hookrightarrow časová doména

- ten čas platnosti neme separačne uladab' k dátam so forme množiny intervalov, kedy boli dát'a platné → neme ľahšie zistit, kedy platila dana' podmienka. → rel. dát'a + časové' dát'a
- ďalší typ u modelu s timestampami:

$$(T \times D^n) \rightarrow \text{bool}$$
- porovnanie so snapshot modelom:

$$T \rightarrow (D^n \times \text{bool})$$
- Model so snapshotsmi a model s timestampami sú ekvivalentné.

4.) Tabuľka s obdobujúcim časom

- iba pripojuje
- čas platnosti aj transakcie
- podpora pre roll-back
- dokazy do histórie
- napr. CV v Šanone za posledujúcich X rokov

Meno	Ústav	Platný od	Transakcia
Jan	UIVT	1	1
Jan	UIVT	2	1
:	:	:	:
Jan	UIVT	1	2
Jan	UIVT	1	2
:	:	:	:
Jan	UMAT	6	8
Jan	UMAT	7	8

Platnosť

Transakcia - t.j. kedy som túto
informáciu zmienil

Dotasoracie jazyku

• Relačný jazyk

$$L ::= r_i(x_1, \dots, x_k) \mid x_i = x_j \mid L \wedge L \mid \neg L \mid \exists x. L$$

napr. Vypíš všetkých zamestnancov IBM

$$\{x : \exists y. zam(x, y) \wedge y = \text{IBM}\}$$

-typické rozšírenia:

- implicitné odkazy na časové údaje
→ časové spojky
- explicitné odkazy na časové údaje
→ premenné a kvantifikátory

↪ ale potrebujeme sa opriť o nejaký formálny matemat. apáčik

1. Výroková logika (prepozícia) - Booleova algebra.

2. Logika 1. rádu (predikátora) - universálne a exist. kvantifikátory

3. Logika vyššieho rádu - dedukcia nad množinami a funkciami

4. Modelová / temporalná logika - dedukcia o tom, čo sa stane, môže slabiť

Temporalná logika

• spojky

↪ X_1 until X_2 : $\exists t_2. t_0 < t_2 \wedge X_2 \wedge \forall t_1 (t_0 < t_1 < t_2 \rightarrow X_1)$

→ X_1 prestalo platiť, keď začalo platiť X_2 (nevraň kedy začalo platiť X_1)

↪ X_1 since X_2 : $\exists t_2. t_0 > t_2 \wedge X_2 \wedge \forall t_1 (t_0 > t_1 > t_2 \rightarrow X_1)$

→ X_2 prestalo platiť, začalo platiť X_1

- $\Diamond X_1 = \text{TRUE until } X_1$
 $\hookrightarrow \underline{\text{nichdy}} (\vee \text{ budúcnosť})$

- $\blacklozenge X_1 = \text{TRUE since } X_1$
 $\hookrightarrow \underline{\text{nichdy}} (\vee \text{ minulosť})$

- $\Box X_1 = \neg \Diamond \neg X_1$
 $\hookrightarrow \underline{\text{vždy}} (\vee \text{ budúcnosť})$

- $\blacksquare X_1 = \neg \blacklozenge \neg X_1$
 $\hookrightarrow \underline{\text{vždy}} (\vee \text{ minulosť})$

- formule until a since môžu byť rozdelené na podčasti
 obrazujúce iba dôkazy na minulosť, súčasnosť a budúcnosť

Temporalna logika 1. rádu

- reči \rightarrow je konečná množina tempor. spojek

$F ::= r_i(x_{i1}, \dots, x_{ik})$ - DB schéma

$F \wedge F, \dots$ - logické spojky

$x_i = x_j$

$\exists x_i. F$ - dátá (premenne)

$w(F_1, \dots, F_k)$ - temporalne spojky

Priklady:

- odpoved' obecne $\rightarrow \Psi(\text{DB}) = \{t, \theta : \text{DB}, \theta, t \models \Psi\}$

- Historia zamensluania jína:

$\exists x. \text{Zam}(x, \boxed{y}) \wedge x = \text{Jan}$

\hookrightarrow volná premenná - vybubľovacia povaha

- výsl. je: zamensluvateľ 1 čas 1
 zamensluvateľ 2 čas 2

- v ktorich relačných DB aby sme dostali aktuálneho zamensluvateľa

- Všetci, kto boli zavolať k rovnakému zamestnávateľovi

$$\exists y. \text{Zam}(x, y) \wedge \Diamond(\neg \text{Zam}(x, y) \wedge \Diamond \text{Zam}(x, y))$$

- Všetci, ktorí medzi 2 zamestnania boli nezamestnaní:

$$\exists x. \text{Zam}(x, y) \wedge \neg \exists y. (\text{Zam}(x, y)) \wedge \Diamond \exists y. (\text{Zam}(x, y))$$

- Všetci, ktorí pracovali v IBM od tej doby, čo jan odštiel
since jan už nie je \wedge všetci, ktorí tam vtedy pracovali

Temporalne relačné výskazy

\hookrightarrow prístup, kde čas je miestom dát implicitne (ako ide o temporálne tab.)

$M := R_i(t_j, x_i, \dots, x_n)$ - ~~základna~~ schéma DB

$M \wedge M, \neg M$ - log. správy

$x_i = x_j, \exists x_i. M$ - data ('premenne')

$t_i < t_j, \exists t_i. M$ temporalne premenne'

- jedna' sa o 2-dmiersnu logiku 1. rádu (2 druhov' = čas + data)

Formát odpovede na dotaz:

$$\Psi(DB) = \{ \theta : DB, \theta \models \varphi \} \Rightarrow \text{čas je miestom dát}$$

- porovnanie s TL 1. rádu:

$$\Psi(DB) = \{ t, \theta : DB, \theta, t \models \varphi \}$$

Príklady

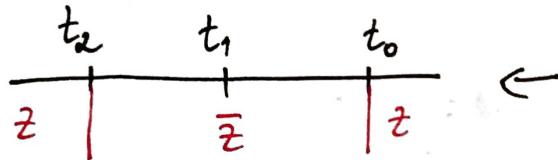
- História Honzovho zamestnania:

$$\exists x. \text{Zam}(t_0, x, y) \wedge x = \text{jan}$$

ročné 'premenne' - bude sa výsledkom "variant"

- Všetci, čo boli zmenu prijali k rovnakému zamestnávateľovi

$$\exists y. \text{zam}(t_0, x, y) \wedge \exists t_1 (t_1 < t_0 \wedge \neg \text{zam}(t_1, x, y)) \wedge \exists t_2 (t_2 < t_1 \wedge \neg \text{zam}(t_2, x, y))$$



- Všetci, čo boli medzi 2 zamenskami nezamenili'

$$\exists y. \text{zam}(t_0, x, y) \wedge \exists t_1 (t_1 < t_0 \wedge \neg \exists y \text{zam}(t_1, x, y)) \wedge \exists t_2 (t_2 < t_1 \wedge \exists y \text{zam}(t_2, x, y))$$

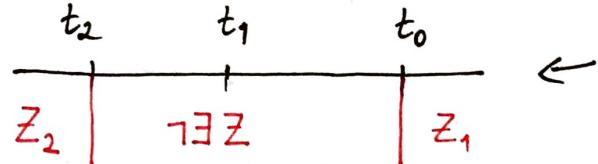
v t₁ netočili zamensku'
 zamensk. z₁
 v čase t₀

v t₂ boli zamensku' (v z₂)

- ak vypustime 2. riadok, dostaneme,

aj ľudí, ktorí napr. zmenili len

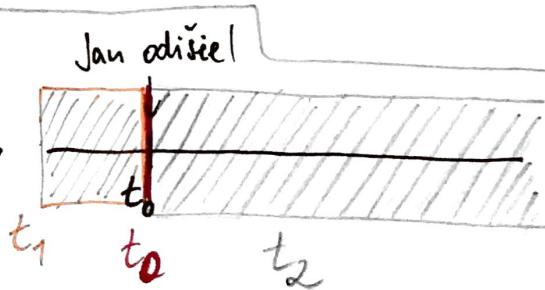
zamestnávateľa, ale netočili nezamenili' medzi tým.



- Všetci, čo pracovali u IBM od tej doby, čo Ján odišiel

$$\exists t_1 < t_0. (\text{zam}(t_1, \text{Ján}, \text{IBM}) \wedge \forall t_2. t_2 > t_1 \rightarrow \neg \text{zam}(t_2, \text{Ján}, \text{IBM})) \wedge \forall t_2. t_1 < t_2 \leq t_0 \rightarrow \text{zam}(t_2, x, \text{IBM})$$

- zjednodušene zapísane', malo by to byť
čas premenne' $x = \text{Ján} \wedge y = \text{IBM}$



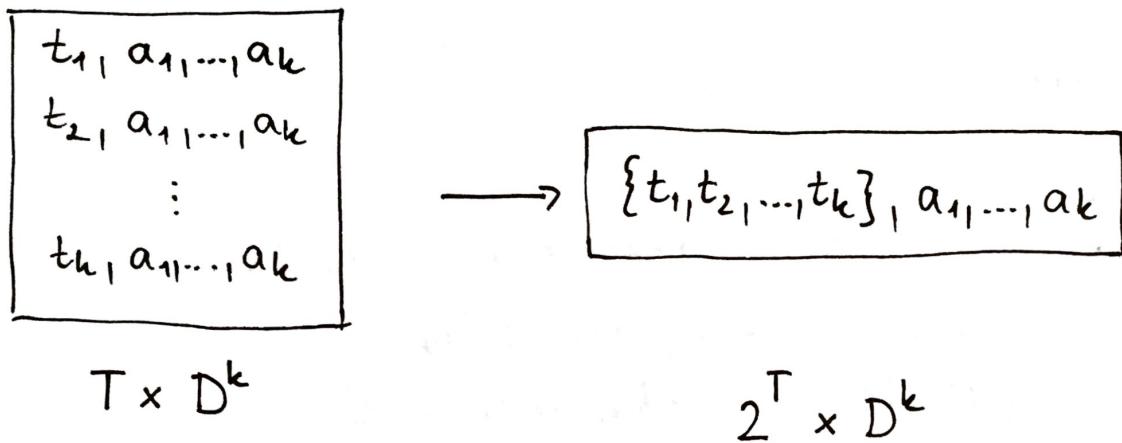
ja by som to napísal ako

~~aaaa~~

$$\forall t_1 > t_0. (\neg \text{zam}(t_1, \text{Ján}, \text{IBM}) \wedge \text{zam}(t_1, x, \text{IBM})) ??$$

Konkretné TDB

- záskazy s rovnakými dátami ale rôznymi časmi sa slúčia a časové okamihy sa nejako kódujú
- 1 dátova n-tica je časovo vstázena k nekolíjnym časovým okamihom



Integritné temporálne obmedzenia

\hookrightarrow ale niekoľko výhodív, tak ho už nemôžem zobrať naspäť)

lebo

Konečné kodoranie obmedzení

- umožňuje časového okamihov sú definované mojimi charakteristickými funkiami (formulami)
- jazyk obmedzenia umožňuje odstraňovať kuantifikátory

Kodoranie intervalov

- nech T_p je časová doména
- def. umožníme

$$I(T) = \{(a, b) \mid a \leq b, a \in T \cup \{-\infty\}, b \in T \cup \{\infty\}\}$$

• Relácia na umožnenie $I(T)$:

$$\hookrightarrow ([a, b] \leq [a', b']) \Leftrightarrow a < a'$$

$$\hookrightarrow ([a, b] \leq_+ [a', b']) \Leftrightarrow a < b'$$

$$\hookrightarrow ([a, b] \leq_{+-} [a', b']) \Leftrightarrow b < a'$$

$$\hookrightarrow ([a, b] \leq_{++} [a', b']) \Leftrightarrow b < b'$$

Štruktúra $T_I = (I(T), \langle -, \langle_-, \langle_{+-}, \langle_{++})$ sa nazýva
intervalová časová doména vzhľadom k T_P

→ 2 rôzne TDB násťa ine' výsledky !!

• Intervalove dokazy —

$M ::= R_i(I_j, x_{i1}, \dots, x_{in})$ - rozšírená schéma DB

$M \wedge M, \neg M$ - logické spojky

$x_i = x_j, \exists x_i. M$ - dátka, premenne'

$I_i^* = I_j^* ; \exists t_i. M$ - temporálne premenne'

Generickosť dokazov

operácia získania výsledku

Dokaz f je genericky vzhľadom k $\boxed{\|.\|}$ ak plati', že

$$\|D_1\| = \|D_2\| \Rightarrow \|f D_1\| = \|f D_2\|$$



Príklad

$$R^{D_1} = \{([0,3], a)\}$$

$$R^{D_2} = \{([0,2], a), ([1,3], a)\}$$

$$\rightarrow \exists i, j : \exists x (R(i, x) \wedge R(j, x) \wedge i \neq j)$$

↪ plati' v D_2 , ale neplati' v D_1 \Rightarrow nie je genericky

↳ Problem: Dokaz, ktorý nie je genericky, môže "zpracovať" určité fakty iba podľa toho, ako mi dala učosene'

↳ možnosti riešenia

1. zmena semantiky - veľmi složite'

2. zmena štruktúry DB - je možné

- budeme DB mitiť ukladať dátá tak, aby k tomu nedoslo.
- po každej operácii námne medzivýskol preto preniesť do "štúnej formy" \Rightarrow coalescing

Žhlukoranie (coalescing)

DEF

1-rozmerne temporálne relácia obsahuje žhluky, ak je každý faktus spojovaný s najkratším koncovým počtom neprekrývajúcich sa intervalov:

- je treba zaručiť, že mi nad reláciami vykonávané ne-logické operácie
- relačné operátory žhlukoranie využívajú (proj., ajedn., unív., rozdiel)

Vysvetlivky:

1-rozmerne = 1 čas
faktus = n-tica podľa času

- niekedy je treba spojiť intervale krôži
datnému spracovaniu, intervale, ktoré
sa prekrývajú alebo na seba narázujú
sa dajú hľadu spojiť.

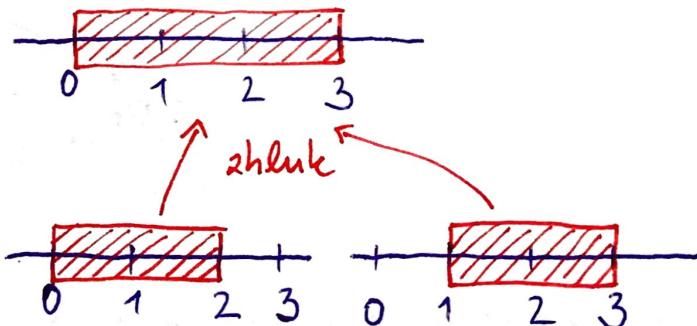
Príklad

- DB so žhlukmi:

$$R^{D_1} = \{([0, 3], a)\}$$

- DB bez žhlukov:

$$R^{D_2} = \{([0, 2], a), ([1, 3], a)\}$$



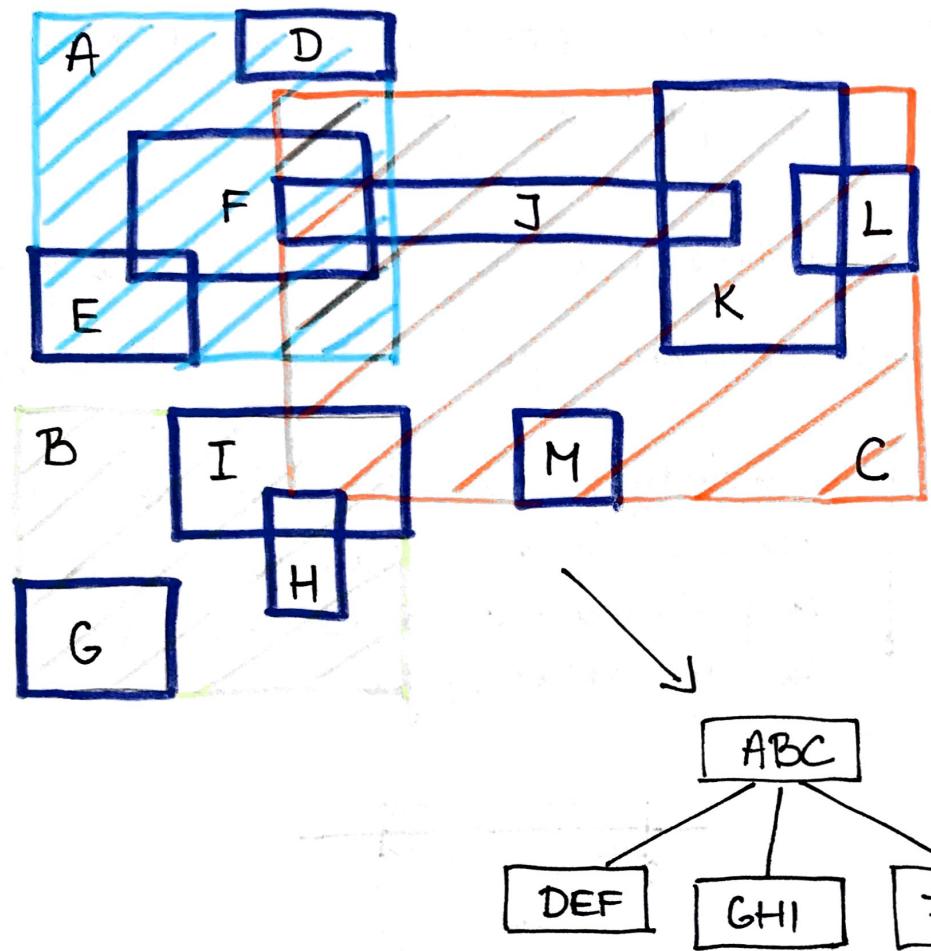
VIEDIEŤ \rightarrow genericky dokaz (že to poruší, problem)
 \rightarrow žhlukoranie (že to opravuje)

- žhluky - zjednodušujú selekciu, projekciu, spojenie
 - nejednodušujú temporálne operácie

Indexovanie v TDB

Problém: máme interval I, vyber všetky n-tice (údaje), ktoré majú s týmto intervalom neprázdný priesečník
 → výsledkom je nevyvážená štruktúra pre prístup k dátam

R-tree



- deliace oblasti sa môžu prekrývať
 tzn., že 1 objekt môže byť v 2 prehľadávacích časoch

Time index

- na začiatku a konci každého intervalu ulož zoznam všetkých intervalov obsahujúcich tento okamih
 - nad týmto bodmi posiaľ B-tree
- ↳ dočasy $\approx O(\log(n) + s/B)$
- ↳ priestor $\approx O(n^2/B)$ - nevýhoda (odstráni kradieľ)

Specifiche' problemy

- pri delení je složitý výber dieratických bodov na časovej osi
- optimalizácia pri dočasoranej si často založené na odhadu veľkosti výsledku

↳ data nie sú v čase rovnomerne rozložené: typicky existujú zhluky - ale potom ale zniť granularitu?

↳ v zhlukoch by mala byť nižšia, ale keď sa potom v DB 3 dni nie vedeje tak je zbytočné mať granularitu na minuty

Temporalne integrálne obmedzenia

- u relačných DB sú ^{obmedzenia} usavrené logické formule 1. rádu
- u temporálnych DB sú obmedzenia usavrele formule 1. rádu temporálneho dočasoracieho jazyka
- formule spravia, keď je obsah DB korektný
- v každom zaznamenanom okamihu musia integr. obmedzenia platí
- vieme, že v minulosti obmedzenie platilo,
- pre miernosť ho vieme vypočítať "privátko" - to znamená, že vieme, že platilo + ešte, či plati TERAZ po zmene.

Používanie integr. obmedzení

- zachytanie semantiky DB aplikácie
- dôvod zavedenia: ukladať iba "významné" dátá
- návrh DB:
 - ↳ normálne formy - predpíňajú vhodný spôsob formy DB
 - ak je to v NF, tak SRBD z toho môže vysledovať alebo fungovať efektívnejšie než keď to v NF nebolo
 - ak je to v NF tak sa zbarivíme → duplicit dát funkčných závislostí'

- v časových DB sa naviac jedna o závislosti generujúce obmedzenia
- to, čo platilo včera plati v kontexte včerajšku
- to, čo plati už dnes je v kontexte dneška.

- História H splňuje obmedzenie O ak O je pravdivé v každom stave H
- História je postupnosť zmen v DB
 - pri implementácii sa po každej zmeni kontroloje O v aktuálnom H

Príklady

- dřívě propuštění zaměstnanci nemohou být znova přijat
 - dřívě propuštění zaměstnanci nemohou být znova přijati pokud nebyli reabilitováni
 - rehabilitace je možná jen jednou
- $$\neg (\exists x) (rehab(x) \wedge \diamond rehab(x))$$