

- 5 *Realizace jazyka* nebo též *interpretace jazyka* je...
Mlček to nazývá *model jazyka*.
- 5 Realizace termů se uvažuje pro daný jazyk L a jeho realizaci M .
Ohodnocení proměnných...
Nečekaný přechod do výrokové logiky.
- 5 (pozměněné ohodnocení – v predikátové logice)
Vše, co se týká ohodnocování proměnných patří do výrokové logiky.
- 6 (důkaz z předpokladů $T \vdash$)
Tahle verze platí ve výrokové logice, predikátová má ještě další odvozovací pravidla.
- 11 (O úplnosti.)
Nenechat se zmást tím, že *úplnost* o které je tahle věta je úplně jiný pojem, než *úplnost* zavedená na předchozí stránce.
- 12 Prolog pracuje s hornovskými klauzulemi, což jsou klauzule, ve kterých se vyskytuje nejvýše jeden pozitivní (nenegovaný) literál.
K lepší orientaci by mohla pomoci poznámka ve stylu: Formule tvaru $\neg x_1 \vee \dots \vee \neg x_n \vee y$ se dá napsat ekvivalentně jako $(x_1 \wedge \dots \wedge x_n) \rightarrow y$, tedy přesně jako v Prologu.
- 14 ... z principu nedeterministický; vždy se nahrazuje vrchol zásobníku, nechte ale pokaždé vstupní symboly.
Z tohoto ten nedeterminismus obecně nepřamení - mylná představa (navádí k ní i formulace u Bartáka). Místo z *principu* bych řekl z *definice*.
- 11 Teorie má model (je splnitelná), právě když každá její konečná část má model. ($T \models A$ právě když $T \cup \{A\}$ pro nějakou konečnou podteorii $T \cup \{A\} \subseteq T$.)
Místo *nějakou* patří *každou*.
- 13↓2 ...která (což je nejdůležitější) rozkládá všechna slova jazyka do konečně mnoha tříd.
Nejdůležitější je, že rozkládá *úplně všechna* slova do konečně mnoha tříd.
- 13 Třída regulárních jazyků nad abecedou X je nejmenší třída, která obsahuje \emptyset , $\forall x \in X$ obsahuje x a je uzavřená na sjednocení, iteraci a zřetězení.
Kdo v tom trochu plave, ten by ocenil přesnou formulaici: $\forall x \in X$ obsahuje $\{x\}$.

- 16 S $L1 \supset L2$ nastává problém, protože bezkontextové gramatiky umožňují pravidla tvaru $X \rightarrow \lambda$. Řešením je převod na nevypouštějící bezkontextové gramatiky - takové bezkontextové gramatiky, které nemají pravidla typu $X \rightarrow \lambda$.
Poznámka by mohla být pochopitelnější, kdyby se napsalo $\mathcal{L}1 \supseteq \mathcal{L}2$ (o ostrost nejde).
- 16↓6 definuje ukončení výpočtu
Nesmysl, s ukončením výpočtu to nesouvisí. Jde zase o to, aby nebylo nikdy víc možností.
- 17 4.3 Existuje uzávěrová vlastnost, na kterou nejsou uzavřené jazyky typu 0? TODO
Třída rekurzivně spočetných jazyků není uzavřená na doplněk. Např. halting problem v ní je, ale jeho doplněk není.
- 20 Jiná definice: Mějme rozhodovací problém A, výsledek problém chápeme jako funkce A(x) na vstupu x, pak problém A je redukovatelný na...
Je to úplně totéž, takže bych neřekl *jiná definice* ale spíš *jinak řečeno*.
- 20 typu CNF, tj.:...
To není obecná CNF ale 3-CNF
- 20 ...nemůžu spodní vrchol obarvit jinak, než jako logický or tří barev
To neplatí. Platí ale „něco nahoře je bílé \Leftrightarrow ex. obarvení s bílou dole”
- 21 O algoritmech ve třídě P také říkáme, že jsou efektivně řešitelné.
Patří *problémech* místo *algoritmů*
- 22 testování isomorfismu grafů (jsou 2 grafy izomorfní?) – není NP-úplný, ale není P.
Omyl! Objev ne-NP-těžkého problému by znamenal $P \neq NP$. O „graph isomorphism” problému se neví jestli je v P, ani jestli je NP-úplný.
- 22 Sptal sa ma ci viem dokazat, ze SAT je NP uplny. To som fakt nevedel. za 2
Dr. MJ to dělá pomocí převodů „simulace $p(n)$ kroků daného NTS (pro pevný polynom p) \rightarrow kachlíkování” a „kachlíkování \rightarrow SAT”
- 22 2. IF NP, potom, je NP-úplný? -Neplatí
Omyl, to se neví.
- 29 Je-li i strýc červený, přebarvit otce a strýce načerno a přenést chybu o patro výš (je-li děd černý, končím, jinak můžu pokračovat až do

- kořene, který už lze přebarvovat beztestně).
Místo *děd* patří *praděd*. A zdůraznil bych, že děd je přebarven na červeno (čímž se právě přenese ta chyba výš).
- 29 Odstraním uzel stejně jako v předchozím případě.
Žádný předchozí případ tam není.
- 38 Algoritmus s takovou časovou složitostí sice existuje, ale má velmi vysokou multiplikativní konstantu, takže se v praxi nepoužívá.
Přijde mi matoucí psát najednou *algoritmus* místo *třídící síť*.
- 40 ...velikost E je $m - |V| = n, |E| = m$.
Pomlčka zde není dobrý nápad :)
- 41 Algoritmus (Testování souvislosti grafu/počítání komponent souvislosti)
Chtělo by to poznámku, že pro orientované grafy (předtím vždy zmiňované) to nefunguje.
- 41 `topologické_třídění(v - vrchol) {...}`
Na jaký vrchol to tedy spustíme? A co když napoprvé neprojdou vše? A vůbec, bacha na to, že prohledávám orientovaný graf!
- 42 `proved' operaci Relax za použití cest vedoucích do v přes všechna možná u;`
Místo *cest* má asi být *hran*. (?)
- 42 Výsledek dává nejkratší cesty díky topologickému seřazení grafu – pro nejkr. cestu p z s do v platí $t(v_i) < t(v_{i+1})$...
Co je t ? Co je v_i ? Jak to souvisí s p ?
- 43 ...najít kostru T (acyklický souvislý podgraf) grafu (V, E) s celkovou minimální vahou hran.
***Podgrafem* se většinou nazývá i to, co obsahuje jen některé vrcholy z V . Když musí mít za vrcholy celou V , tak se tomu někdy (ale asi ne moc často) říká *napnutý podgraf*.**
- 44 pro každou hranu (u, v) je $0Kč$ $(0 \leq t((u, v)) \leq c((u, v)))$...
Jakýsi podivný překlep.
- 47 `transition funkcia $g : Q \times A \rightarrow Q + \{fail\}$`
Celý ten formální popis konečného automatu je blbě, žádný stav *fail* tam být nemá.
- 56 ...a $f(OPT)/f(APROX)$ pro maximalizační úlohy.
Vylézá mi to tu za okraj stránky.

- 56 Relativní chyba je pak definována jako $|f(APROX)f(OPT)|/f(OPT)$.
Chybí mínus.
- 57 Definice (schéma, model)...
Skopal píše, že „schéma je instancí modelu”. To mi přijde jako šikovná poznámka, a z tohodle odstavce nic takového není patrné.
- 62 $\dots R \subseteq A_1 \times A_2 \dots A_n$ jako množinu n -tic (a_1, a_2, \dots, a_n) (odpovídá konkrétním datům v tabulce) kde $a_i \in A_i$ a prvek relace $r \in R$ (odpovídá řádku v tabulce).
Když už se zavádějí ty domény D_i , tak bych tady čekal spíš $R \subseteq D_1 \times D_2 \dots D_n$ a $a_i \in D_i$.
- 63 přirozené spojení je vlastně kartézský součin (spojuje se přes prázdnou množinu)
Spíš „kartézský součin je vlastně přirozené spojení”.
- 65 Problémy řešené normalizací:...
Vůbec jsem nepochopil ty ilustrativní příklady s *fakultami*, *kurzy*, *adresami*...
- 65 ...neexistuje jediná závislost neklíčových sloupců tabulky.
Kvantifikátor „neexistuje jediná” matfyzáka zaručeně zmate :)
- 66 ...zachovává pokrytí závislostí, když $F^+ = F_1^+ \cup F_2^+$ – nesmí se ztratit závislost ani v rámci dílčího schématu, ani jdoucí napříč schématy.
To plusko se nikde nedefinovalo, musel jsem si ho najít u Skopala (je to uzávěr množiny závislostí vzhledem k Armstrongovým pravidlům).
- 67 Dotazy v SQL
Doufal jsem v nějaký přívětivý popis fungování různých joinů (včetně levého a pravého), a to tu není.
- 97 Vid'. bankéřův algoritmus.
<http://www.1000vecicomeserou.cz/453-viz-s-teckou/>
- 109 bazické řešení¹⁴
Poznámka č. 14 není nikde vysázená.
- 127 Datové typy v prologu se nazývají termy.
Dost matoucně řečeno
- 131 Tento příklad jasně ukazuje výhodu nad makry...
Trochu divoký příklad. Kvůli podobným hackům (compile-time výpočty) snad templaty nevznikly..