



Starší zkouškové varianty

1. Esej

Souhrnně pojednejte o časové a prostorové složitosti algoritmů a problémů. (Jaké typy složitosti rozlišujeme, jak složitost měříme). Uveďte co znamenají symboly O a Θ a co jsou to P-těžké, NP-těžké a NP-úplné problémy. Uveďte příklad aspoň dvou NP-úplných problémů.

Jak je definován deterministický a nedeterministický automat. Jak zásobníkový automat rozpozná, zda slova do jazyka patří či nepatří. Které jazyky lze rozpoznávat automatem?

Esej o Grafech (vysvětlit co je to cesta, sled, strom, podgraf, navrhnout nějaký algoritmus na výše uvedené a odhadnout jeho složitost)

Popište, jak může vypadat správně vytvořená formule predikátové logiky (predikátového počtu). Její syntaxi. Co znamená interpretace formule a kdy je tato formule splnitelná. Co víte o časové složitosti problému splnitelnosti predikátové formule.

Popište, co znamená, že formule predikátové logiky je logickým důsledkem množiny formulí predikátové logiky (tedy dané teorie). Jak probíhá logický důkaz? Co to znamená, že daná teorie je bezrozporná? Jaký je vztah mezi logickým důsledkem a sémantickým důsledkem dané množiny formulí? V čem spočívá princip rezoluční metody zjišťování, zda je daná formule sémantickým důsledkem zadané množiny formulí?

Formule v predikátorové logice, jak je tvořená, její interpretace, kdy je formule splnitelná, kdy je tautologie a kdy kontradikce + časová složitost predik. zápisu.

Souhrnně popište jak pracuje model výpočtu nazývaný Turingův stroj. Co znamená, že Turingův stroj rozhoduje jazyk a co znamená, že rozpoznává jazyk (problém). Co jsou rekurzivní a rekurzivně spočetné jazyky (problémy) a jaký je jejich vzájemný vztah?

2. Teoretická otázka

Uveďte princip prohledávání uzlů orientovaného a neorientovaného grafu do hloubky a do šířky. Jaké datové struktury jsou vhodné pro uchovávání informací o hranách grafu?

Maximální tok v síti, Ford-Fulkersonova věta, načrtnout algoritmus při hledání max. toku a jeho časová složitost, co je řez grafem a co to je tok v grafu

Stručně uveďte co znamená, že deduktivní systém (teorie) je bezrozporný a co znamená, že je úplný! Co platí pro přirozenou dedukci výrokové a predikátové logiky?

Uzávěrové vlastnosti regulárních jazyků a vůči jakým operacím jsou uzavřeny.

Formulujte problém maximálního toku v síti. Uveďte princip algoritmu pro řešení tohoto problému. Na čem závisí složitost tohoto algoritmu?

Definujte tyto pojmy z teorie grafů: sled, tah, cesta, souvislý graf, strom, isomorfní grafy, nakreslení grafu.

Uveďte stručně princip genetických algoritmů + kde se využívají?

Kostra grafu, minimální kostra grafu, definice, algoritmus řešení, výpočetní složitost.



3. Příklad

Množiny:

$$K = \{(x, y): x^2 + y^2 \leq 1\}$$

$$H = \{(x, y): y \geq 0\}$$

$$L = \{(x, y): (x + 0,5)^2 + y^2 \leq 0,25\}$$

$$P = \{(x, y): (x - 0,5)^2 + y^2 \leq 0,25\}.$$

Nakreslete a vyšrafujte v rovině množinu bodů $((K \cap P) \div L) \cup H$

Predikát:

Student, který látce rozumí, vyřeší všechny příklady v písemce. $\forall S: R(S) \Rightarrow V(S)$

Kdo vyřeší všechny příklady, uspěje u zkoušky. $\forall S: V(S) \Rightarrow U(S)$

Student, který látce rozumí, uspěje u zkoušky. $\exists S: R(S)$

Predikát:

Kdo dokončil maratón, je zdravý. $\forall X: D(X) \Rightarrow Z(X)$

Někteří z přihlášených závodníků maratón nedokončili. $\exists X: \neg D(X)$

Někteří z přihlášených nejsou zdraví. $\exists X: \neg Z(X)$

Prolog – Adélka:

muz(jaromir).

zena(helena).

zena(adelka).

rodic(helena,adelka).

rodic(jaromir,adelka).

rodice(R,D):- muz(R,D) ; zena(R,D).

?- rodice(R,D).

Relace

Mezi lidmi jsou definovány tyto binární relace

aPb , pokud je a předkem b v přímé linii dědičnosti,

aVb , pokud je a vlastním sourozencem b (to je mají oba rodiče společné)

aBb , pokud je a vlastním nebo nevlastním sourozencem (mají aspoň jednoho společného rodiče),

aRb , pokud mají a a b týž rok narození a .

aSb , pokud je a aspoň o 3 roku starší než b .

aZb , pokud a se narodil ve stejném roce nebo dříve než b .

Do následující tabulky doplňte logické hodnoty 0 = NE nebo 1 = ANO podle toho, zda relace příslušnou vlastnost má nebo nemá!

Relace

Mezi celými čísly jsou definovány tyto relace:

aDb , pokud a dělí b beze zbytku.

aZb , pokud a a b dávají při dělení číslem 7 týž zbytek.

aSb , pokud se čísla a a b liší o nejvýše 10.

aRb , pokud se čísla a a b liší aspoň o 10.

aPb , pokud součet čísel a a b je sudý.

aZb , pokud po zaokrouhlení na celé stovky na a' a b' (podle dohodnutých pravidel) je $a' > b'$

Do následující tabulky doplňte logické hodnoty 0 = NE nebo 1 = ANO podle toho, zda relace příslušnou vlastnost má nebo nemá!



Relace

Relace mezi množinami

aRb : a je podmnožinou b

aSb : a je vlastní podmnožinou b

aTb : a má mohutnost menší nebo rovnu než b

aXb : a má mohutnost menší než b

Výpočetní složitost

Nejjednodušší algoritmus výpočtu n -té mocniny čtvercové matice A řádu n (tedy s n řádky a n sloupci) spočívá v postupném násobení matic maticí A . Tedy $A^1=A$, $A^2=A^1.A$, ... $A^n=A^{(n-1)}.A$. Pokud pro násobení matic použijete algoritmus vyplývající přímo z definice součinu matic, odpovězte na otázky:

- jaká bude časová výpočetní složitost algoritmu vzhledem k řádu n dané matice?
- jaká bude časová výpočetní složitost algoritmu vzhledem k rozměru vstupu (tedy vzhledem k rozměru skutečného vstupu – všech prvků dané matice)?
- jak by bylo možné časovou složitost snížit?

4. Automat

Vstup 0 a 1, výstup počet jedniček dělitelný třemi.

Vstup písmena anglické abecedy, výstup $\{s,o,s\}$.

Vstup 0 a 1, ve výstupu každý znak minimálně dvakrát bez ohledu na to, v jakém pořadí.

Vstup 0 a 1, výstup 3 nebo více stejných symbolů bezprostředně za sebou.

Nápojový automat na limonádu, vstup 1, 2 nebo 5, cena limonády 5 Kč.

Vstupy a, b, c, každý ze 3 symbolů vstupní abecedy alespoň jednou.

Automyčka, 4 mycí programy: O-S, O-P-O, O-P-O-S, O-P-S-V, kde O je Oplach, P je Pěna, S je Sušení a V je Vosk.

Jiná verze:

Automyčka, 4 mycí programy: O-S, O-P-S, O-P-O-S, O-P-S-V, kde O je Oplach, P je Pěna, S je Sušení a V je Vosk.

Studenti odevzdávají příklady, za správný 1, za špatný 0. Musí mít 3 správně, aby dostali zápočet. Za každý špatný navíc musí přinést jeden správný. Nesmí být 2 chybné za sebou. Po splnění může odevzdat jakékoli příklady, i špatně vyřešené.

Studenti odevzdávají příklady, za správný 1, za špatný 0. Musí mít alespoň 2 správně, aby dostali zápočet. Každý špatný musí opravit správným. Nesmí být 2 chybné za sebou.

Přepis konečného automatu do Prologu:

```
delta(q1,1,q2).
```

```
delta(q1,2,q3).
```

```
...
```

```
delta(q4,1,q5).
```

```
koncovy(q5).
```

```
prijme(Stav,[]):-koncovy(Stav).
```

```
prijme(Stav1,[X|Zbytek]):-delta(Stav1,X,Stav2), prijme(Stav2,Zbytek).
```