Referát č. 1
Vysvětlete, co znamená tvrzení, že operace levého kvocientu je asociativní. Pak toto tvrzení pečlivě dokažte či vyvraťte.
Dále vysvětlete, proč pro konečný automat $A = (Q, \Sigma, \delta, q_0, F)$ a jazyk L je $L \backslash L(A)$ sjednocením jazyků $L_q^{out-to-F}$ pro vybrané stavy q. Pro které ?

Referát č. 2

- Připomeňte, co je to homomorfismus mezi (relačními a algebraickými) strukturami (uveďte více konkrétních příkladů), a také definujte homomorfismus mezi konečnými automaty. Dále připomeňte, co to znamená, že dvě (relační a algebraické) struktury jsou izomorfní. Pak konkrétně aplikujte na konečné automaty (vysvětlete, co to znamená, že dva automaty jsou izomorfní).
- Popište a na příkladu ilustrujte (rychlý) algoritmus testující, zda dané dva automaty jsou izomorfní.

Referát č. 3
K regulárnímu jazyku $L \subseteq \Sigma^*$ nazveme *kanonickým* takový KA A, $L(A) = L$, který je v normovaném tvaru (a tudíž jsou všechny stavy dosažitelné) a v němž pro každé $w \in \Sigma^*$ existuje právě jeden stav q takový, že $L_q^{out-to-F} = w \backslash L$.
Dokažte, že ke každému regulárnímu jazyku L existuje právě jeden kanonický automat. Vysvětlete, proč je kanonický automat minimálním.
Dále ukažte, že metoda minimalizace automatu připomenutá na přednášce sestrojí k jakémukoli KA A' kanonický automat pro jazyk $L(A')$.

Referát č. 4
Ukažte a podrobně vysvětlete, že pro každé n existuje nedeterministický automat A_n s n stavy takový, že kanonický automat (definice kanonického automatu viz. předchozí zadání) k jazyku $L(A_n)$ má 2^n stavů.

Referát č. 5
Připomeňte generalizovaný konečný automat (GKA), o němž jsme mluvili na přednášce 19.3.2007 v souvislosti s konstrukcí regulárního výrazu k danému automatu. (Hrany u tohoto GKA jsou obecně označeny regulárními výrazy, nejen jednotlivými znaky.) Exaktně definujte pojem GKA a jazyka jím přijímaného. Pak popište a ilustrujte na příkladu jádro konstrukce, která k danému konečnému automatu A sestrojí dvoustavový GKA s jedinou hranou označenou regulárním výrazem reprezentujícím $L(A)$. (Téma můžete najít např. v současných slidech k ÚTI, přístupných na webu ing. Kota. K ilustraci ovšem použijte jiný vhodný příklad než tam uvedený.)

Referát č. 6
Vysvětlete pojem regulárních gramatik (RG) a podrobně ukažte jejich vztah ke konečným automatům. Na vhodných příkladech ilustrujte převody mezi (N)KA a RG a naopak.

Referát č. 7
Popište pojem redukované bezkontextové gramatiky a algoritmus redukce; ilustrujte na vhodném příkladu. Poukažte zároveň na souvislost s algoritmem zjišťujícím, zda daná bezkontextová gramatika generuje neprázdný jazyk.

Referát č. 8
Popište (a na vhodném příkladu ilustrujte) převod gramatiky do Chomského normální formy.

Referát č. 9
Vysvětlete pumping lemma pro regulární jazyky a pumping lemma pro bezkontextové jazyky – střídající se kvantifikátory ve znění lemmat důkladně objasněte na hře dvou hráčů (ze studijního textu).

Referát č. 10
Předveďte algoritmický postup, který k zadanému vícestavovému zásobníkovému automatu zkonstruuje ekvivalentní jednostavový zásobníkový automat; ilustrujte na vhodném příkladu (alespoň) dvoustavového automatu a pak postup popište obecně.

Referát č. 11
Vysvětlete, co to znamená, že dvoupáskový (a obecně k-páskový) Turingův stroj je možné simulovat jednopáskovým Turingovým strojem, a podrobně předveďte, jak je taková simulace realizována.

Referát č. 12
V definici modelu RAM v základním studijním materiálu je uvedena hodnota operandu $*i$ jako číslo uložené na adrese, jež je dána součtem čísla i a čísla uloženého v indexregistru. Jiná užívaná možnost nepřímé adresace je, že hodnota $*i$ je chápána jako číslo uložené na adrese, která je uložena v buňce s adresou i.
Ukažte, jak lze RAM-program v jedné variantě simulovat RAM-programem v druhé variantě a naopak.

Referát č. 13
Důkladně promyslete, popište a vysvětlete následující příklad (s návodem).

Mějme standardní Turingův stroj (předpokládající oboustranně nekonečnou pásku) $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Sestrojme k němu Turingův stroj $M' = (Q', \Sigma, \Gamma', \delta', q_0', F')$, který předpokládá jen jednostranně (tj. pravostranně) nekonečnou pásku—tedy z nejlevější buňky (na níž stojí hlava na počátku) nemůže přejít doleva—a přitom simuluje stroj M.
Naznačíme možný způsob konstrukce:
$Q' = \{q_0', q_1\} \cup \{q_x \mid x \in \Sigma\} \cup \{q_U \mid q \in Q\} \cup \{q_D \mid q \in Q\}$
$\Gamma' = \Sigma \cup (\Gamma \times \Gamma) \cup \{\not{c}, \Box\}$
$F' = \{q_U \mid q \in F\} \cup \{q_D \mid q \in F\}$
$\delta'(q_0', x) = (q_x, \not{c}, +1)$ … pro $x \in \Sigma$
$\delta'(q_x, y) = (q_y, (x, \Box), +1)$ … pro $x, y \in \Sigma$

$\delta'(q_x, \square) = (q_1, (x, \square), -1)$ … pro $x \in \Sigma$
$\delta'(q_1, z) = (q_1, z, -1)$ … pro $z \neq \not\in$
$\delta'(q_1, \not\in) = ((q_0)_U, \not\in, +1)$
Obrázkem si znázorněte pásku a (na malém příkladu) počáteční fázi práce stroje M' (pozn.: asi vás napadne pojem 'dvoustopá páska'); doplňte pak instrukce stroje M' (tedy dodefinujte zobrazení δ') tak, aby skutečně simuloval M. (Ještě kousek nápovědy: U v indexu u stavu znamená 'up', D znamená 'down').

Referát č. 14
Představte si Turingův stroj pracující na "čtverečkované rovině" (místo lineární pásky). Vstupní slovo je zapsáno na začátku v jednom řádku, čtecí hlava stojí na jeho začátku (ostatní buňky=čverečky obsahují prázdný znak). Obor hodnot přechodové funkce je nyní rozšířen tak, že možné pohyby hlavy jsou Left, Right, Up, Down.
Přesně takový stroj definujte a precizně popište, jak je možné simulovat tento "rovinný" stroj klasickým "lineárním" (uveďte také konkrétní příklad).

Referát č. 15
Vysvětlete, jak lze problém (z pracovního textu)

> *Název problému*: *Výběr aktivit*
>
> *Vstup*: množina konečně mnoha aktivit $\{1, 2, \ldots, n\}$ s pevně určenými časovými intervaly (s_1, f_1), (s_2, f_2), …, (s_n, f_n), kde $(\forall i, 1 \leq i \leq n) : s_i < f_i$
> *Výstup*: množina obsahující největší možný počet vzájemně kompatibilních aktivit (tj. aktivit s vzájemně se nepřekrývajícími intervaly)

řešit hltavým (greedy) algoritmem.
Dokažte indukcí podle počtu aktivit n, že vámi uvedený přístup skutečně vede k optimálnímu řešení.

Referát č. 16
Sestavte důkaz použitelnosti hltavého přístupu při konstrukci minimální kostry grafu (důkaz je stručně nastíněn v pracovním textu).

Referát č. 17
Např. v prezentaci vas-predn-02-ho.pdf na web-stránce teoretické informatiky (u přednášky 16.4.2007) je naznačen algoritmus (se složitostí $O(n)$) pro hledání max. vzdálenosti v konvexním polygonu.
Zkonstruujte správný a přehledný, dobře okomentovaný pseudokód tohoto algoritmu (tedy ve formě blízké skutečnému programu), a dokažte, že algoritmus skutečně nalezne onu maximální vzdálenost.

Referát č. 18
Algoritmus (Cocke-Younger-Kasami) pro rozpoznávání bezkontextových jazyků (aplikace metody dynamického programování):

Mějme dánu bezkontextovou gramatiku G v tzv. Chomského normální formě, tedy s pravidly pouze typu $\boxed{X \longrightarrow YZ}$ a $\boxed{X \longrightarrow a}$. Algoritmus pro zadané (terminální) slovo w zjistí, zda $w \in L(G)$.

Nástin: Označme $w = a_1 a_2 \ldots a_n$. Systematicky vyplňujeme (dvourozměrné) pole D tak, že na závěr bude $D[i,j]$ ($1 \leq i \leq n$, $0 \leq j \leq n-i$) obsahovat množinu právě těch neterminálů X, z nichž lze odvodit $a_i a_{i+1} \ldots a_{i+j}$.

Vysvětlete tento algoritmus, ilustrujte na příkladu a analyzujte jeho časovou složitost.
(Další podklady je možno najít např. na http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm, referát 4).

Referát č. 19 Předveďte algoritmus polynomiálního převodu problému nezávislé množiny na problém hamiltonovského cyklu.
(K nastudování můžete např. využít souboru IS-HC.pdf, který najdete u přednášky ze 7.5.2007 na http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.htm.)

Referát č. 20
Prokažte, že problém 3-SAT je polynomiálně převeditelný na problém obarvení grafu 3 barvami (3-SAT $\triangleleft$ 3-CG).
(K nastudování můžete např. využít příslušnou animaci v souboru Novak/index.htm, který naleznete u přednášky z 16.4.2007 na http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.htm.)

Referát č. 21
Popište konstrukci v důkazu Cookovy věty.
(K nastudování můžete např. využít podklad k referátu č. 7 na http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm.)

Referát č. 22
Popište konstrukci v důkazu Savitchovy věty.
(K nastudování můžete např. využít podklad k referátu č. 8 na http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm.)

Referát č. 23
Uvažujte problém

> *Název*: QBF (*problém pravdivosti kvantifikovaných booleovských formulí*)
>
> *Vstup*: formule $(\exists x_1)(\forall x_2)(\exists x_3)(\forall x_4)\ldots(\exists x_{2n-1})(\forall x_{2n})\mathcal{F}(x_1, x_2, \ldots, x_{2n})$, kde $\mathcal{F}(x_1, x_2, \ldots, x_{2n})$ je booleovská formule v konjunktivní normální formě.
> *Otázka*: je daná formule pravdivá ?

Navrhněte algoritmus, který řeší problém QBF a má prostorovou složitost omezenou polynomem; odvoďte zároveň stupeň tohoto polynomu.
Návod. Řekneme, že formule $\mathcal{F}(x_1, x_2, \ldots, x_{2n})$ je OK pro posloupnost booleovských hodnot $b_1, b_2, \ldots, b_i$, kde $0 \leq i \leq 2n$, jestliže

buď $i = 2n$ a $\mathcal{F}(b_1, b_2, \ldots, b_{2n}) = true$,

nebo $i < 2n$, i je liché a $\mathcal{F}$ je OK jak pro $b_1, b_2, \ldots, b_i, true$, tak pro $b_1, b_2, \ldots, b_i, false$,

nebo $i < 2n$, i je sudé a $\mathcal{F}$ je OK pro aspoň jednu z posloupností $b_1, b_2, \ldots, b_i, true$ a $b_1, b_2, \ldots, b_i, false$.

Ověřte nejprve, že formule $(\exists x_1)(\forall x_2)(\exists x_3)(\forall x_4) \ldots (\exists x_{2n-1})(\forall x_{2n})\mathcal{F}(x_1, x_2, \ldots, x_{2n})$ je pravdivá právě tehdy, když $\mathcal{F}$ je OK pro prázdnou posloupnost.
Pak sestavte kýžený algoritmus.

Referát č. 24
Uvažujme problém, jehož instancí je orientovaný graf s vybraným vrcholem v a dále k ‘oblázků’. Můžeme v jakémkoli pořadí provádět následující elementární kroky:

- na vrchol x můžeme položit oblázek, pokud v daný okamžik leží oblázky na všech vrcholech, z nichž vede hrana do x,
- oblázek položený na vrchol můžeme odebrat (a znovu použít později).

Otázkou je, zda existuje posloupnost kroků, při níž položíme oblázek na zadaný vrchol v.
Prokažte, že problém je v PSPACE.
Dále vysvětlete, jak je možné uvedenou hrou modelovat situaci přidělování paměti při výpočtu (stačí daný počet registrů k provedení daného výpočtu ? …).