Referáty

Upozornění. Toto je verze z 2. týdne (konkrétně z 6.3.2008). V této verzi jsou již plná zadání referátů.

Poznámky. (Omlouvám se, že používám pojmy *vyučující*, *cvičící* apod. jen v mužském rodě, i když se v konkrétním případě může jednat o ženu.)

- Referáty budou přiděleny v zásadě na prvním cvičení. Studenti, kteří to nestihnou, se musí urychleně přihlásit cvičícímu (třeba emailem).
- Níže uvedené rozdělení referátů do týdnů odpovídá předběžnému plánu, je ale možné, že se podle potřeby změní. Cvičící si s vámi domluví způsob sdělení takové případné změny (pokud nebude hromadně oznámena v Aktuálních informacích na web-stránce předmětu). (Data odpovídající číslům týdnů jsou zřejmá z Průběhu výuky na web-stránce předmětu.)
- Na referátu je nejdůležitější vaše **osobní prezentace**, kdy prokážete, že jste tématu plně porozuměli a prezentaci si pečlivě připravili tak, aby jste podstatu věci stihli kolegům a vyučujícímu rozumně vysvětlit (ilustrovat) v čase **nejvýše 15 minut**.
- K tématu můžete čerpat informace a podklady z jakýchkoli veřejných zdrojů, ale musíte je sami za sebe srozumitelně a uceleně písemně zpracovat. Konzultovat můžete také se svými kolegy (přirozeně např. s těmi, kteří mají tentýž referát na jiném cvičení), **vyučující vám ale k tomuto poskytovat vysvětlující konzultace v zásadě nebudou**. Jistě vám neodmítnou krátkou radu, ale nebudou suplovat to, co máte prokázat vaší samostatnou prací. Něco jiného je ale konzultace zmíněná v dalším bodu, která je naopak žádoucí.
- Aby se předešlo nežádoucímu případu neuznání referátu (a tím neudělení zápočtu) z toho důvodu, že referující tématu ve skutečnosti nerozumí a/nebo nemá připravenou rozumnou (nanejvýš 15 minutovou) prezentaci, **je žádoucí**, aby se referující domluvil s cvičícím na schůzce nejméně týden před plánovanou prezentací, tam mu předložil (alespoň předběžnou) písemnou přípravu a **v (jen) několika minutách načrtl**, jak hodlá postupovat. Pokud se ukážou problémy (student tématu nerozumí, prezentaci nemá promyšlenou, apod.), cvičící ho na to upozorní a případně poskytne možnost další krátké schůzky (nebude ovšem téma studentovi vysvětlovat, jak už je zmíněno výše).

Týden 1

Přidělení referátů.

Týden 2

Referát č. 1

Vysvětlete operaci levého kvocientu pro jazyky, tedy operaci $L_2 \backslash L_1$ (definice, příklady, ...).
Vysvětlete, co znamená tvrzení "operace levého kvocientu je asociativní". Pak toto tvrzení dokažte či vyvraťte.

Referát č. 2

Vysvětlete, proč pro konečný automat $A = (Q, \Sigma, \delta, q_0, F)$ a jazyk $L \subseteq \Sigma^*$ je (levý kvocient) $L \backslash L(A)$ sjednocením jazyků L_q^{toAcc} pro vybrané stavy q. Pro které ?

Týden 3

Referát č. 3

Připomeňte, co je to homomorfismus a co je to izomorfismus mezi (relačními a algebraickými) strukturami (uveďte několik konkrétních příkladů). Pak vysvětlete, co to znamená, že dva konečné automaty jsou izomorfní.
Popište (rychlý) algoritmus testující, zda dané dva automaty jsou izomorfní.

Referát č. 4

Vysvětlete, jaký problém řeší tzv. Knuth-Morris-Prattův algoritmus (Knuth-Morris-Pratt algorithm) a ilustrujte jej např. na příkladu z přednášky z 1. týdne.

Týden 4

Referát č. 5

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 $L_q^{toAcc} = w \backslash L$.
Vysvětlete, proč ke každému regulárnímu jazyku L existuje právě jeden kanonický automat, a proč je kanonický automat minimálním.

Referát č. 6

Vysvětlete, proč pro každé n existuje nedeterministický automat A_n s n stavy takový, že minimální deterministický konečný automat přijímající $L(A_n)$ má 2^n stavů. (Ilustrujte např. na konkrétním příkladu pro $n = 5$. Ten můžete najít např. ve starším materiálu `http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`.)

Týden 5

Referát č. 7 (Hltavý algoritmus 1)
Vysvětlete (na vhodně zvoleném případu), jak lze problém (ze studijního 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.
Ukažte myšlenku indukce podle počtu aktivit n prokazující, že uvedený přístup skutečně vede k optimálnímu řešení.
Referát č. 8 (Hltavý algoritmus 2)
Připomeňte na vhodně zvoleném případu, jak se řeší problém konstrukce minimální kostry grafu hltavým přístupem, a ilustrujte myšlenku důkazu toho, že tento přístup skutečně vede k optimu. (Můžete vyjít z popisu ve studijním textu a podle potřeby použít další materiály.)
Referát č. 9 (Dynamické programování)
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 ilustrací na vhodném příkladu.
(Další podklady je možno najít např. na
`http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm`, referát 4.)

Týden 6

Referát č. 10
Vysvětlete pojem regulárních gramatik (RG) a jejich vztah ke konečným automatům. Na vhodných příkladech ilustrujte převody mezi (N)KA a RG a naopak. (Můžete vyjít z materiálu
`http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`
a/nebo z jiných zdrojů.)

Referát č. 11 (Redukce bezkontextové gramatiky)
Vysvětlete, co dělá algoritmus popsaný v sekci 3 publikace

> J. Esparza, P. Rossmanith, and S. Schwoon. A uniform framework for problems on context-free grammars. EATCS Bulletin, 72:169-177, October 2000,

která by měla být přístupná na
`http://www7.in.tum.de/um/bibdb/author-esparza.shtml`.
(Ilustrujte na vhodném příkladu.)
Pak stručně vysvětlete, jak je možné tento algoritmus využít při redukci gramatiky (tj. při "Identifying useless variables" na str. 8 zmíněné publikace).
Referát č. 12 (Chomského normální forma bezkontextových gramatik)
Při převodu bezkontextové gramatiky do Chomského normální formy (který je naznačen např. v textu
`http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`)
se po odstranění pravidel typu $A \to \varepsilon$ provádí krok, který odstraňuje pravidla typu $X \to Y$ (neterminál se přepisuje na neterminál). (Pak tedy dostaneme jen pravidla typu $A \to \alpha$, kde α je terminál nebo $|\alpha| \geq 2$, přičemž generovaný jazyk se nezměnil.)
Na vhodném příkladu ilustrujte algoritmus odstraňující ona pravidla $X \to Y$.

Týden 7

Referát č. 13 (Pumping lemma pro regulární jazyky)
Vysvětlete tzv. pumping lemma pro regulární jazyky
(můžete vyjít např. z `http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`)
a naznačte na příkladu, jak jej lze využít pro důkaz neregularity nějakého jazyka.
Referát č. 14 (Pumping lemma pro bezkontextové jazyky)
Připomeňte pumping lemma pro bezkontextové jazyky (např. ze studijního textu) a vysvětlete souvislost se hrou dvou hráčů popsanou např. v
`http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`.
Referát č. 15 (Zásobníkové automaty)
Tvrzení „Ke každému ZA M lze sestrojit ZA M' s jedním stavem tž. $L(M) = L(M')$“
lze dokázat pomocí následující obecně popsané konstrukce.
Jednostavový ZA M', se stavem označeným s, bude mít zásobníkové symboly typu $\langle p, X, q \rangle$, kde p, q jsou stavy a X je zásobníkový symbol automatu M, a speciální počáteční zásobníkový symbol R.
Konkrétně pro $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ konstruujeme $M' = (\{s\}, \Sigma, \Gamma', \delta', s, R)$, kde $\Gamma' = (Q \times \Gamma \times Q) \cup \{R\}$ a δ' je určena následovně:

- $\delta'(s, \varepsilon, R) = \{(s, \langle q_0, Z_0, q \rangle) \mid q \in Q\}$,
- pro $(q', \varepsilon) \in \delta(q, a, X)$ $(a \in (\Sigma \cup \{\varepsilon\}))$ zařadíme do $\delta'(s, a, \langle q, X, q' \rangle)$ prvek (s, ε),

- pro $(q', A_1A_2 \ldots A_n) \in \delta(q, a, X)$ $(n \geq 1)$ zařadíme do $\delta'(s, a, \langle q, X, \overline{q}\rangle)$ prvek $(s, \langle q', A_1, q_1\rangle\langle q_1, A_2, q_2\rangle \ldots \langle q_{n-1}, A_n, \overline{q}\rangle)$ pro každé $\overline{q}, q_1, q_2, \ldots, q_{n-1} \in Q$.

(Chápeme-li δ' jako množinu 'instrukcí', pak lze říci, že δ' je minimální množina instrukcí splňující výše uvedené podmínky.)
(Dá se ověřit, že

$$(s, w, \langle p, X, q\rangle) \vdash^*_{M'} (s, \varepsilon, \varepsilon) \Longleftrightarrow (p, w, X) \vdash^*_M (q, \varepsilon, \varepsilon)\,, \tag{1}$$

a tedy každému přijímajícímu výpočtu automatu M nad slovem w odpovídá přijímající výpočet automatu M' nad w a naopak.)
Vašim úkolem je předvést aplikaci této konstrukce na zásobníkový automat M se vstupní abecedou $\{a, b\}$, zásobníkovou abecedou $\{A, B\}$, počátečním zásobníkovým symbolem A, množinou stavů $\{p, q, r\}$, počátečním stavem p a přechodovou funkcí δ definovanou následovně
$\delta(p, a, A) = \{(q, AA), (p, B)\}$,
$\delta(q, b, A) = \{(q, AA)\}$,
$\delta(p, \varepsilon, B) = \{(q, A)\}$,
$\delta(q, \varepsilon, A) = \{(r, \varepsilon)\}$,
$\delta(r, a, A) = \{(r, A)\}$,
$\delta(r, b, A) = \{(r, \varepsilon)\}$
(pro ostatní prvky def. oboru je funkční hodnota rovna $\emptyset$).
Přitom se snažte alespoň intuitivně přiblížit platnost uvedeného vztahu (1) v onom konkrétním případě.

Týden 8

Referát č. 16 (Simulace mezi různými variantami Turingových strojů 1)
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:
$Q' = \{q_0', q_1\} \cup \{q_x \mid x \in \Sigma\} \cup \{q_U \mid q \in Q\} \cup \{q_D \mid q \in Q\}$
$\Gamma' = \Sigma \cup (\Gamma \times \Gamma) \cup \{\not{c}, \Box\}$
$F' = \{q_U \mid q \in F\} \cup \{q_D \mid q \in F\}$
$\delta'(q_0', x) = (q_x, \not{c}, +1) \ldots$ pro $x \in \Sigma$
$\delta'(q_x, y) = (q_y, (x, \Box), +1) \ldots$ pro $x, y \in \Sigma$
$\delta'(q_x, \Box) = (q_1, (x, \Box), -1) \ldots$ pro $x \in \Sigma$
$\delta'(q_1, z) = (q_1, z, -1) \ldots$ pro $z \neq \not{c}$

$\delta'(q_1, \not{c}) = ((q_0)_U, \not{c}, +1)$
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. 17 (Simulace mezi různými variantami Turingových strojů 2)

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.

Stručně a srozumitelně popište, jak je možné simulovat tento "rovinný" stroj klasickým "lineárním" strojem. (Nápověda. Musíte tedy popsat, jak bude mít lineární stroj uložen na pásce obsah oné roviny; stačí mít v každém okamžiku zachycen jen obdélník obsahující všechna políčka roviny, která simulovaný stroj dosud navštívil. Pak musíte popsat, jak bude simulující stroj provádět analogii konkrétních instrukcí simulovaného.)

Týden 9

Referát č. 18 (Další otázka týkající se Turingových strojů)

Představme si, že navrhujeme Turingův stroj M, přičemž chceme používat již hotový M_1 jako proceduru, které lze předat vstup (parametr), tj. řetězec symbolů, a obdržet od ní příslušný výstup.

K tomu se hodí chápat M jako dvoupáskový (víme, že vícepáskový stroj lze simulovat jednopáskovým, je-li potřeba). Když M potřebuje výstup M_1 odpovídající vstupu w, neboli potřebuje zjistit řetězec $M_1(w)$, napíše w na druhou pásku, na ní pak nechá běžet M_1 a až M_1 skončí, překopíruje si M výsledný řetězec z druhé pásky na „základní" pásku a udělá si s ním, co potřebuje.

Popište teď, jak byste řešili případ, kdy M potřebuje takto „volat" stroje $M_1, M_2, \ldots, M_k$, které se ovšem mohou také *rekurzivně volat navzájem.*

Referát č. 19

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. (Ilustrujte na jednoduchém příkladu.)

Týden 10

(1.5. se cvičení nekoná.)

Týden 11

(8.5. se cvičení nekoná.)

Týden 12

Referát č. 20 (Rozhodnutelnost a nerozhodnutelnost)
Uvažujme problém

Název: UHP (*Uniform Halting Problem*)

Vstup: Turingův stroj M.

Otázka: Zastaví se M na každý vstup?

Zjistěte, zda tento problém je rozhodnutelný či nerozhodnutelný a své zjištění prokažte. (Můžete např. vyjít ze stručné zmínky v textu
`http://www.cs.vsb.cz/jancar/TEORET-INF/teoret-inf.pdf`.)

Referát č. 21 (Převeditelnost mezi problémy)
Demonstrujte myšlenku převeditelnosti IPKP (iniciálního Postova korespondenčního problému) na PKP (Postův korespondenční problém). (Můžete vyjít např. z příslušné animace, zvolte si ale jiný příklad, na němž myšlenku srozumitelně předvedete a vysvětlíte.)

Referát č. 22 (Časová složitost algoritmů, asymptotická notace)
Podejte matematický důkaz (využívající např. l'Hospitalova pravidla) toho, že je-li $f(n) \leq p(n)$ pro nějaký polynom p a $g(n) \geq c^n$ pro nějakou konstantu $c > 1$, tak platí $f \in o(g)$.
Podobně to ukažte pro případ, kde $f(n) \leq (\log n)^k$ pro nějakou konstantu k a $g(n) \geq n^c$ pro nějakou konstantu $c > 0$.

Týden 13

Referát č. 23 (Polynomiální převeditelnost)
Jedna z animací k předmětu ukazuje polynomiální převeditelnost problému nezávislé množiny na problém hamiltonovského cyklu.
Je to technicky netriviální konstrukce (která vyšla z Referátu 6 na
`http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm`).
Prostudujte ji a prezentujte na co nejjednodušším konkrétním případu, který ještě umožňuje rozumnou demonstraci hlavní myšlenky.

Referát č. 24 (NP-úplnost)
Cookova věta říká, že SAT je NP-úplný problém. Zaměřme se na důkaz toho, že SAT je NP-těžký, tedy že pro každý problém $P \in \mathsf{NP}$ platí $P \lhd \mathrm{SAT}$ (P je polynomiálně převeditelný na SAT).
Uvažujme tedy libovolný, ale dále pevně daný, problém $P \in \mathsf{NP}$. Tou libovolností se rozumí to, že o P nemůžeme předpokládat nic jiného než $P \in \mathsf{NP}$ (nikoli to, že bychom se snad

mohli zaměřit jen na jeden konkrétní problém, který bychom si zvolili podle vlastní libosti). O P tedy víme, že je rozhodován nějakým nedeterministickým Turingovým strojem M s časovou složitostí $T_M(n) \leq p(n)$ pro nějaký polynom p. (O M a p zase nic bližšího nevíme, ale jsou už teď pro nás pevně dané.) Máme ukázat, že existuje polynomiální algoritmus, který k libovolnému vstupu w stroje M zkonstruuje booleovskou formuli $\mathcal{F}_w$ (v konjunktivní normální formě), která je splnitelná právě tehdy, když pro slovo w existuje přijímající výpočet stroje M.

Formule $\mathcal{F}_w$ má být tedy splnitelná právě tehdy, když existuje posloupnost konfigurací $C_0, C_1, C_2, \ldots, C_m$ stroje M, kde $m \leq p(n)$, C_0 je počáteční konfigurace odpovídající vstupu w, C_m je přijímající konfigurace a pro každé $i = 0, 1, \ldots, m-1$ platí $C_i \vdash_M C_{i+1}$ (tedy z C_i může stroj M jedním krokem přejít do C_{i+1}).

Obecný návod k sestrojení formule $\mathcal{F}_w$ zachycující schéma takového (potenciálního) výpočtu lze nalézt např. v podkladu k referátu č. 7 na
`http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm`.)

Podívejte se také na příslušnou animaci v seznamu animací odkazovaném na web-stránce předmětu.

Uvažujme teď tento konkrétní případ. Stroj M je dán následujícím seznamem instrukcí (q_1 je počáteční stav, q_{acc} je přijímající stav, q_{rej} je zamítající stav, vstupní abeceda je $\{a, b\}$, pracovní abeceda je $\{a, b, \Box\}$):

$$
\begin{array}{l}
(q_1, 0) \rightarrow (q_1, 0, +1),\ \ (q_1, 0) \rightarrow (q_2, 0, +1) \text{ (nedeterminismus)} \\
(q_1, 1) \rightarrow (q_1, 1, +1),\ \ (q_1, 1) \rightarrow (q_2, 1, +1) \text{ (nedeterminismus)} \\
(q_1, \Box) \rightarrow (q_{rej}, \Box, 0)
\end{array}
$$

$$
\begin{array}{l}
(q_2, 0) \rightarrow (q_{rej}, 0, 0) \\
(q_2, 1) \rightarrow (q_{acc}, 1, 0) \\
(q_2, \Box) \rightarrow (q_{rej}, \Box, 0)
\end{array}
$$

Je zřejmé, že časová složitost stroje M je $T_M(n) = n + 1$.

Předveďte a vysvětlete obecnou konstrukci formule $\mathcal{F}_w$ tak, že ji demonstrujete pro uvedený konkrétní M a vstupní slovo w délky 2, např. $w = 10$.

Referát č. 25 (Savitchova věta)

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`.)

Můžete se ovšem omezit na tento speciální případ:

> Je-li problém P rozhodován nedeterministickým Turingovým strojem s prostorovou složitostí n, pak je také rozhodován deterministickým Turingovým strojem s polynomiální prostorovou složitostí.

Máte tedy vysvětlit, jak lze k tzv. lineárně omezenému automatu (linear bounded automaton) M, tj. k nedeterministickému Turingovu stroji M, který při výpočtu na vstupním w, $|w| = n$, nenavštíví jiná políčka než ta, na nichž je zapsán vstup (a má tedy prostorovou složitost n)

navrhnout (deterministický) algoritmus A, který pro zadané w zjistí, zda M má přijímající výpočet pro w (tedy zda $w \in L(M)$). Algoritmu A přitom musí stačit polynomiálně omezená paměť.

(Připomenutí. Počet konfigurací délky n stroje M je omezen hodnotou c^n, kde konstantu c lze snadno spočítat z velikosti (stavové množiny a abecedy) stroje M. Délka nejkratšího přijímajícího výpočtu M nad w, $|w| = n$, (pokud takový existuje) je tedy také omezena oním c^n.)

(Bylo by dobré ukázat, že ta prostorová složitost A se dá omezit kvadraticky, je v $O(n^2)$, a naznačit, proč A lze přímočaře implementovat deterministickým Turingovým strojem s prostorovou složitostí $O(n^2)$.)

Týden 14

Referát č. 26 (Problém QBF; Quantified Boolean Formulas)
Uvažujme problém

Název: QBF (*problém pravdivosti kvantifikovaných booleovských formulí*)

Vstup: formule $(\exists x_1)(\forall x_2)(\exists x_3)(\forall x_4) \dots (\exists x_{2n-1})(\forall x_{2n})\mathcal{F}(x_1, x_2, \dots, x_{2n})$, kde $\mathcal{F}(x_1, x_2, \dots, x_{2n})$ je booleovská formule v konjunktivní normální formě.
Otázka: je daná formule pravdivá ?

Navrhněte algoritmus, který řeší problém QBF a má prostorovou složitost omezenou polynomem. (Tím ukážete, že QBF je v PSPACE.)
Návod. Řekneme, že formule $\mathcal{F}(x_1, x_2, \dots, x_{2n})$ je OK pro posloupnost booleovských hodnot $b_1, b_2, \dots, b_i$, kde $0 \le i \le 2n$, jestliže

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

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

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

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

Pak sestavte kýžený algoritmus (a prokažte, že jeho prostorová [tedy paměťová] složitost je polynomiální).

Referát č. 27 (Oblázková hra v PSPACE)
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.
(Jednou z motivací problému je problém přidělování paměti při výpočtu; stačí daný počet registrů k provedení určeného výpočtu ?)