Alternatívna charakterizácia synchronizovaného alternovania

Dana Pardubská[1⋆] Jiří Wiedermann[2⋆⋆]

[1] Katedra informatiky
Fakulta matematiky, fyziky a informatiky Univerzity Komenského
Mlynská dolina, 842 48 Bratislava, Slovensko
`pardubska@fmph.uniba.sk`
[2] Ústav informatiky Akademie věd České republiky
Pod Vodárenskou věží 2, 182 07 Praha 8, Česká republika
`jiri.wiedermann@cs.cas.cz`

Abstrakt V článku navrhneme nový výpočtový model, ktorého návrh je inšpirovaný základnými vlastnosťami bezdrôtových mobilných komunikačných sietí. V podstate ide o deterministický Turingov stroj obohatený o možnosť vytvárania nových komunikujúcich procesov, pričom „bezdrôtová“ komunikácia medzi procesmi prebieha prostredníctvom explicitne určených kanálov. Dokážeme, že výpočtovo sú tieto stroje časovo aj priestorovo ekvivalentné synchronizovaným alternujúcim Turingovým strojom. To ukazuje, že synchronizované alternovanie možno nahradiť deterministickým paralelizmom s možnosťou neobmedzenej komunikácie a naopak.

1 Úvod

Prirodzeným dôsledkom vývoja bezdrôtových a mobilných komunikačných technológií je zvýšenie záujmu o skúmanie algoritmických aspektov odpovedajúcich výpočtových a komunikačných mechanizmov. Zdá sa, že doterajší výskum sa sústredil najmä na otázky konkrétnych algoritmov, pričom sa takmer úplne zabúdalo na zložitostné aspekty týchto výpočtov. Jedným z dôvodov môže byť nedostatok formálnych modelov pre takýto typ výpočtov. Ak sa pokúsime identifikovať základné výpočtové a komunikačné vlastnosti príslušných výpočtov, po značnom zjednodušení skončíme pri dynamických rekonfigurovateľných sieťach mobilných procesorov, ktoré na vzájomnú komunikáciu používajú „rádio“, ktoré sa dá naladiť pre príjem či vysielanie na ľubovoľnom kanáli. Správu, vysielanú nejakým procesorom (vrcholom siete) na jednom kanáli, tak môžu počuť všetky na tento kanál naladené vrcholy.

Čo sa dá povedať o výpočtovej sile práve načrtnutého „výpočtového modelu“? Prináša — v porovnaní s klasickými výpočtami — bezdrôtové mobilné

⋆ Výskum bol čiastočne podporovaný grantom APVT-20-018902
⋆⋆ Práca vznikla v rámci výskumného zámeru AV0Z10300504 a bola čiastočne podporovaná grantom 1ET100300517.

počítanie do výpočtov novú kvalitu? Odpovede na tieto otázky sú hlavnou motiváciou tohto článku. Aby sme na ne mohli odpovedať, budeme definovať formálny model, ktorý zachytáva hlavné črty mobilných bezdrôtových výpočtov, dôležité z hľadiska ich funkčnosti. Snaha o zachovanie istej realističnosti modelu nás doviedla k návrhu deterministického zariadenia. V podstate to bude Turingov stroj, ktorý bude schopný deterministicky spúšťať paralelné procesy s možnosťou vytvárať komunikujúce skupiny. Pri jeho návrhu sme sa snažili o transparentnosť komunikačného mechanizmu. Aby sa mohlo uskutočniť spojenie v rámci rôznych skupín procesov, musia si tieto skupiny procesov alokovať rôzne kanály (avšak rovnaké pre danú skupinu). Rovnaký mechanizmus medziprocesorovej komunikácie použijeme aj na identifikáciu ukončenia výpočtu. Spomínaná transparentnosť tak zviditeľňuje (a robí merateľnými) aktivity, ktoré sa v prípade alternujúcich Turingových strojov dejú „za scénou“ a sú z pohľadu výpočtovej zložitosti zadarmo.

Ani zďaleka netvrdíme, že náš model je ideálnym modelom mobilných bezdrôtových výpočtov. Napriek tomu vidíme jeho dva prínosy. Po prvé, náš model charakterizuje výpočtovú silu istého typu bezdrôtových mobilných výpočtov. Druhým, a možno dokonca dôležitejším prínosom tohto modelu je alternatívna charakterizácia výpočtového mechanizmu synchronizovaného alternovania, čo prináša novú charakterizáciu pamäťových tried PSPACE a EXPTIME. Ukazuje sa, že synchronizované alternovanie je ekvivalentné deterministickému paralelizmu s explicitným mechanizmom komunikácie.

V našej práci sme vychádzali z princípu alternovania (napr. [1]), ktorý sa v teórii zložitosti považuje za najelegantnejší nástroj na štúdium paralelných výpočtov, a z nadväzujúcich prác [6], [3] a [2], ktoré rozširujú mechanizmus alternovanie o možnosti komunikácie medzi bežiacimi procesmi. Vnímať synchronizované alternujúce procesy ako bezdrôtovo komunikujúce mobilné procesy sa na prvý pohľad môže zdať prehnané; v skutočnosti to tak nie je. Už aj klasický alternujúci Turingov stroj v podstate využíva „bezdrôtovú“ komunikáciu medzi bežiacimi procesmi pri určovaní terminácie výpočtu. Mobilita procesov je v ich dynamickom vzniku a zániku a tiež v tom, že nie sú viazané konkrétnym umiestnením v priestore.

Pri návrhu modelu sme upustili od nedeterminizmu, ktorý je základnou zložkou alternovania. Namiesto nedeterminizmu a synchronizácie sme zaviedli univerzálny mechanizmus deterministického vymieňania správ medzi procesmi. Hlavný výsledok práce je vcelku prekvapivý — tento nový mechanizmus nielenže dokázal nahradiť stratu nedeterminizmu a akceptačného mechanizmu alternujúcich Turingových strojov, ale zachoval aj výhodu, známu zo synchronizovaných Turingových strojov, a síce väčšiu silu synchronizovaného priestoru, ktorá je dokumentovaná ekvivalenciou tohoto priestoru so sychronizovaným exponenciálne väčším časom (pozri napr. [6], [3]). Dostali sme stroj, ktorý je ekvivalentný synchronizovaným alternujúcim Turingovým strojom.

Štruktúra článku je nasledovná. V časti 2 definujeme model, tzv. sieť Turingových strojov (WPTM — z anglického wireless parallel Turing machine), spolu s príslušnými zložitostnými mierami. V časti 3 sa venujeme základnej zložitostnej charakterizácii výpočtov takýchto strojov a nakoniec v časti 4 zhrnieme dosiahnuté výsledky.

2 Sieť Turingových strojov — WPTM

Pri vytváraní výpočtového modelu, v ktorom procesy komunikujú podobne ako prvky bezdrôtových mobilných sietí, využijeme schopnosť procesov alternujúcich Turingových strojov vytvárať v univerzálnom stave nové procesy. Túto schopnosť doplníme o špeciálny mechanizmus modelujúci bezdrôtový prenos informácie medzi procesmi naladenými na rovnaký kanál. Akceptačný mechanizmus alternujúcich strojov nahradíme flexibilným komunikačným mechanizmom.

Definícia 1 *Sieť k–páskových Turingových strojov (WPTM) s jednou vstupnou a jednou kanálovou páskou je systém* $M = (k, Q, R, \Sigma, \Gamma, \Delta, q_0, r_0, q_{accept}, q_{reject})$*, kde*

- *k je počet pracovných pások;*
- *$Q \times R$ je konečná množina stavov.*
  - *Q je množina pracovných stavov, q_0, $q_0 \in Q$ je počiatočný stav;*
  - *R je množina komunikačných stavov, ktorá obsahuje tri špeciálne stavy: počiatočný komunikačný stav r_0,* akceptujúci *stav q_{accept}* a zamietajúci *stav q_{reject};*
- *Σ je konečná vstupná abeceda (\$ $\notin \Sigma$ je tzv. zarážka);*
- *Γ je konečná pracovná abeceda ($\sharp \in \Gamma$ je prázdny symbol, $\sharp \notin \Sigma$);*
- *$\Delta \subseteq Q \times R \times (\Sigma \cup \{\$\}) \times \Gamma^{k+1} \times Q \times R \times (\Gamma - \{\sharp\})^{k+1} \times \{\leftarrow, \rightarrow\}^{k+2}$ je prechodová relácia.*

Prvkom Δ hovoríme prechody. Stroj má (iba čítaciu) vstupnú pásku so zarážkami, k pracovných pások a jednu kanálovú pásku; pracovné pásky spolu s kanálovou páskou budeme súhrnne nazývať pásky. Pásky sú jednosmerne nekonečné smerom doprava, číslovanie políčok začína od 0; na začiatku výpočtu sú prázdne.

Nech $\delta = \langle q, r, x, a_1, \ldots, a_{k+1}, q', r', a'_1, \ldots, a'_{k+1}, d_1, \ldots, d_{k+2} \rangle \in \Delta$ je prechod M. Podľa tohto prechodu stroj M, ktorý sa nachádza v pracovnom stave q, komunikačnom stave r, číta zo vstupu symbol x, z i-tej pásky a_i, $i = 1, 2, \ldots, k+1$, v jednom *kroku* dosiahne nový pracovný stav q', nový komunikačný stav r', napíše a'_i na i-tu pásku a posunie každú z $k+2$ hláv, $i = 1, 2, \ldots, k+1$, o jedno políčko v smere d_j (doľava alebo doprava), pre $j = 1, 2, \ldots, k+2$.

Konfigurácia M je prvok $Q \times R \times \Sigma^* \times ((\Gamma - \{\sharp\})^*)^{k+1} \times \mathbb{N}^{k+2}$, ktorý popisuje pracovný a komunikačný stav konečnostavovej riadiacej jednotky, vstup, neprázdne obsahy $k+1$ pások a $k+2$ pozícií hláv.

Hlavová konfigurácia stroja M je prvok $Q \times R \times (\Sigma \cup \{\$\}) \times \Gamma^{k+1}$, ktorý popisuje pracovný a komunikačný stav riadiacej jednotky spolu s obsahmi buniek snímanými jednotlivými hlavami.

Hovoríme, že prechod s novým komunikačným stavom r' *vysiela* stav $r' \in R$. Na prechody kladieme jedno syntaktické obmedzenie, tzv. pravidlo súhlasného vysielania: *všetky prechody odpovedajúce rovnakej hlavovej konfigurácii vysielajú rovnaký komunikačný stav.* Líšiť sa môžu vo všetkých ostatných položkách – v novom pracovnom stave, v nových symboloch a v posunoch hláv. Ďalej hovoríme, že konfigurácia je *naladená na kanál* c ak obsahom kanálovej pásky naľavo od aktuálnej pozície hlavy je c; snímané políčko nie je súčasťou c. Ak c je neprázdny reťazec, hovoríme mu *číslo kanála/kanálové číslo*. Konfigurácia vysiela r' na kanáli c, ak je to konfigurácia naladená na kanál c, pričom na ňu aplikujeme prechod s novým komunikačným stavom r'.

Konfigurácia β je δ–nasledovník konfigurácie α vzhľadom na prechod $\delta \in \Delta$ (píšeme $\alpha \vdash^{\delta} \beta$), ak β vznikla z α aplikovaním prechodu δ. Prechodu $\alpha \vdash^{\delta} \beta$ hovoríme *jednoduchý krok* M.

Pripomíname, že konfigurácia môže mať viacero rôznych δ–nasledovníkov. Konfigurácia bez nasledovníka je *terminálna konfigurácia*.

Pri definovaní výpočtu stroja M nám pomôže niekoľko ďalších pojmov a definícií. Začneme definíciami projekcií, ktoré extrahujú z konfigurácie jej zložky.

Funkcia $Tuned : Q \times R \times \Sigma^* \times ((\Gamma - \{\sharp\})^*)^{k+1} \times \mathbb{N}^{k+2} \to (\Gamma - \{\sharp\})^*$ priraďuje konfigurácii číslo jej kanála. Miernym zneužitím tejto notácie môžme funkciu prirodzene rozšíriť na neprázdnu množinu konfigurácií $L \neq \emptyset$:

$$Tuned(L) = \begin{cases} c & \text{ak/práve vtedy keď } Tuned(\alpha) = c \text{ pre všetky } \alpha \in L \\ \perp & \text{ak existujú } \alpha, \beta \in L, \alpha \neq \beta \text{ a } Tuned(\alpha) \neq Tuned(\beta) \end{cases}$$

(symbol $\perp$ označuje nedefinovanú hodnotu.)

Analogicky definujeme funkciu $Broadcast : Q \times R \times \Sigma^* \times ((\Gamma - \{\sharp\})^*)^{k+1} \times \mathbb{N}^{k+2} \to R$, ktorá každej konfigurácii priradí stav, vysielaný prechodmi aplikovateľnými na túto konfiguráciu (vďaka pravidlu súhlasného vysielania je tento stav určený jednoznačne). Funkciu opäť rozšírime na neprázdnu množinu konfigurácií $L \neq \emptyset$:

$$Broadcast(L) = \begin{cases} b & \text{ak pre všetky } \alpha, \beta \in L, Tuned(\alpha) = Tuned(\beta) \\ & \text{a } Broadcast(\alpha) = b \\ \perp & \text{inak} \end{cases}$$

Poslednou projekciou definovanou na množine konfigurácií je $Comm : Q \times R \times \Sigma^* \times ((\Gamma - \{\sharp\})^*)^{k+1} \times \mathbb{N}^{k+2} \to R$, ktorá konfigurácii priraďuje jej komunikačný stav.

Nech $u, v \in R$ sú komunikačné stavy a α konfigurácia v komunikačnom stave u. Symbolom $\alpha|_{u:=v}$ označujeme tú konfiguráciu, ktorá vznikla nahradením (komunikačného) stavu u v pôvodnej konfigurácii α (komunikačným) stavom v.

Nech L je množina konfigurácií, $L_c \subseteq L$ podmnožina tých konfigurácií z L, ktoré sú naladené na c a nech $\alpha \vdash^{\delta} \beta$ je jednoduchý krok. Hovoríme, že konfigurácia γ je δ_L*–nasledovník* konfigurácie α vzhľadom na prechod δ *modifikovaný vysielaním* z L (označujeme $\alpha \vdash^{\delta_L} \gamma$,) ak je definovaná nasledovne:

$$\gamma := \begin{cases} \beta|_{Comm(\beta):=b} & \text{ak } L_{Tuned(\beta)} \neq \emptyset \text{ a } Broadcast(L_{Tuned(\beta)}) = b \\ \perp & \text{ak } L_{Tuned(\beta)} \neq \emptyset \text{ a } Broadcast(L_{Tuned(\beta)}) = \perp \\ \beta & \text{ak } L_{Tuned(\beta)} = \emptyset \end{cases}$$

Predchádzajúca konštrukcia popisuje vysielanie konfigurácií z L na tom kanáli, na ktorý je naladená konfigurácia β. Prvý prípad odpovedá korektnému vysielaniu, keď všetky konfigurácie súhlasne vysielajú rovnaký stav b. Druhý prípad popisuje situáciu, keď na sledovanom kanáli dochádza ku konfliktu vo vysielaní a nakoniec tretí prípad popisuje situáciu, keď sa na sledovanom kanáli nevysiela.

Nech ϱ je konfigurácia; *výpočtový strom* $T(\varrho)$ vzhľadom na prechodovú reláciu Δ stroja M je orientovaný, potenciálne nekonečný strom s koreňom ϱ, ktorého vrcholy sú konfigurácie stroja M. Definujeme ho rekurzívne:

1. ϱ je koreň $T(\varrho)$ v hĺbke $d = 0$;
2. Nech C_d je množina konfigurácií v hĺbke $d \geq 0$. Potom pre všetky neterminálne $\alpha \in C_d$ množina C_{d+1} pozostáva zo všetkých δ_{C_d}–nasledovníkov α; t.j. $C_{d+1} = \{\gamma | \exists$ neterminálna konfigurácia $\alpha \in C_d : \alpha \vdash^{\delta_{C_d}} \gamma\}$. Ak je niektorý z δ_{C_d}–nasledovníkov α nedefinovaný, potom je celý strom nedefinovaný.
   V strome $T(\varrho)$ vedie hrana z každého vrchola $\alpha \in C_d$ do každého jeho δ_{C_d}–nasledovníka $\gamma \in C_{d+1}$.

Pripomíname, že podmienka („úplnosti“) na konci bodu 2 zaručuje, že strom $T(\varrho)$ obsahuje pre každý vrchol α v $T(\varrho)$ množinu všetkých jeho nasledovníkov, ktorých komunikačné stavy môžu byť modifikované vysielaním.

Z popisu vidno, že výpočtový strom je vytváraný plne deterministicky: všetky prechody medzi konfiguráciami rovnako ako všetky medzi–procesorové komunikácie sú definované jednoznačne, bez možnosti akejkoľvek voľby.

Majme (potenciálne nekonečnú) cestu v strome $T(\varrho)$, ktorá začína vo vrchole ϱ. Postupnosti konfigurácií na tejto ceste budeme hovoriť *výpočtová cesta* so začiatkom v ϱ. Výpočtová cesta prirodzene reprezentuje výpočet M na tejto ceste, ale tiež *proces* odpovedajúci tejto ceste.

Hovoríme, že výpočtový strom $T(\varrho)$ je *akceptuje vstup* w ak navyše spĺňa nasledujúce podmienky:

1. *Konečnosť:* $T(\varrho)$ je konečný strom;
2. *Počiatočná podmienka:* $\varrho = (q_0, r_0, w, \underbrace{\nu, \ldots, \nu}_{k+1}, \underbrace{0, \ldots, 0}_{k+2})$ je *počiatočná konfigurácia*, ν označuje prázdny reťazec.
3. *Dohoda o akceptovaní:* listy $T(\varrho)$ sú terminálne konfigurácie v rovnakej hĺbke stromu, v komunikačnom stave q_{accept} a naladené na rovnaký kanál.

Z praktického pohľadu dohoda o akceptovaní hovorí, že všetky procesy zdieľajú v rovnakom čase spoločnú informáciu o tom, že vstupné slovo bolo akceptované. Ak nepoznáme čas, v ktorom má výpočet skončiť, tak akceptovanie v rovnakej hĺbke zabezpečíme napr. pomocou bariérovej synchronizácie. Jej myšlienkou je periodické súčasné testovanie pripravenosti procesov na ukončenie výpočtu (pozri napr. [5]).

Analogicky sa definuje pojem výpočtového stromu zamietajúceho vstup.

Hovoríme, že M *akceptuje* w ak existuje výpočtový strom akceptujúci w. Množinu slov akceptovaných M budeme označovať $L(M)$.

Priestorová (kanálová) zložitosť konfigurácie je súčet dĺžok neprázdnych obsahov pracovných pások (dĺžka neprázdnej časti kanálovej pásky). Priestorová zložitosť výpočtového stromu T je maximum z priestorových zložitostí konfigurácií stromu T; analogicky definujeme kanálovú zložitosť výpočtového stromu T. Časová zložitosť výpočtového stromu T je jeho hĺbka.

Hovoríme, že M *pracuje v priestore* $S(n)$ (*s veľkosťou kanála* $C(n)$), ak pre každý reťazec $w \in L(M)$ dĺžky n existuje výpočtový strom akceptujúci w priestorovej zložitosti nanajvýš $S(n)$ (kanálovej zložitosti nanajvýš $C(n)$). Analogicky, M *pracuje v čase* $T(n)$, ak pre každý reťazec $w \in L(M)$ dĺžky n existuje výpočtový strom akceptujúci w časovej zložitosti nanajvýš $T(n)$.

Rozlišovanie priestorovej a kanálovej zložitosti je motivované bezdrôtovými mobilnými sieťami. Kanálová zložitosť odráža veľkosť príslušného komunikačného mechanizmu.

Na výsledný model sa môžeme pozerať aj ako na Π_1 stroj, ktorý je obohatený o mechanizmus medziprocesorovej komunikácie a má zmenený akceptačný mechanizmus alternujúcich Turingových strojov: kritérium akceptovania musí byť naprogramované v inštrukciách stroja. Vďaka svojej deterministickej povahe predstavuje WPTM do istej miery realistický model bezdrôtových mobilných výpočtov.

Všimnime si, že nahradením prechodovej relácie funkciou v definícii WPTM získame klasický Turingov stroj. Ľahko tiež vidno, že WPTM môže v lineárnom čase simulovať nedeterministický aj Π_1 Turingov stroj. Napriek tomu, že obe zariadenia majú odlišné akceptačné kritériá, WPTM sa ľahko prispôsobí obom. Ukážeme ešte silnejší výsledok: z hľadiska času sú WPTM polynomiálne ekvivalentné alternujúcim Turingovým strojom (a teda aj synchronizovaným Turingovým strojom).

3 Sila bezdrôtovej komunikácie

Začneme porovnaním WPTM s klasickými ATM.

Veta 1 *Nech $T(n)$ je časovo konštruovateľná funkcia, $T(n) \geq n$, A alternujúci Turingov stroj časovej zložitosti $T(n)$. Potom existuje WPTM M, ktorý simuluje A s časovou, priestorovou aj kanálovou zložitosťou $O(T(n))$.*

Náčrt dôkazu: Simuláciu spravíme v dvoch etapách. Najskôr vybudujeme výpočtový strom T_A a potom v jeho listoch spustíme výpočet akceptačného kritéria pre T_A. Kanálovú pásku pritom použijeme na identifikáciu vrcholov v T_A, resp. popis cesty, ktorou sa do nich dostaneme.

Môžeme predpokladať, že každý výpočtový strom T_A stroja A je binárny. M začína svoju prácu v počiatočnej konfigurácii stroja A. Realizuje všetky aplikovateľné prechody A nezávisle od toho, či bola konfigurácia existenčná alebo univerzálna; odpovedá to prechádzaniu T_A po úrovniach zhora nadol. V pracovnej pamäti si však procesy M pamätajú aj tzv. *trasu* — popis výpočtovej cesty v T_A z koreňa do konfigurácie c v hĺbke d, ktorá odpovedá príslušnému procesu v čase d. Vďaka predpokladu môžeme výpočtovú cestu reprezentovať binárnym číslom $(b_1, \ldots, b_d)_2,\ b_i \in \{0, 1\}$; niekedy mu hovoríme aj *číslo cesty* do c.

Súčasťou trasy je aj informácia o type jednotlivých vrcholov T_A(existenčný, univerzálny) na odpovedajúcej ceste. Kvôli druhej etape pridávame aj premennú $qual$, v ktorej budeme uchovávať *kvalitu* jednotlivých vrcholov na ceste. Kvalita vrchola v je prvok z množiny $\{\perp, Y, N\}$, ktorý zodpovedá nedefinovanej, akceptujúcej, resp. zamietajúcej hodnote. Hovorí o podstrome T_v výpočtového stromu T_A s koreňom v. Ak je T_v akceptujúci (zamietajúci) strom, kvalita v je Y (N); inak je kvalita v nedefinovaná, teda $\perp$ (pozri napr. [1]).

Trasa konfigurácie c uloženej vo vrchole u v hĺbke d je teda popísaná $d-$ticou $\langle(b_1, \Box_1, qual_1), \ldots, (b_d, \Box_d, qual_d)\rangle$, kde $b_i \in \{0, 1\}$, $\Box_i \in \{\exists, \forall, |, \}$ a $qual_i \in \{\perp, Y, N\}$. Kvantifikátor $\exists$ zachytáva existenčné vetvenie v u, $\forall$ univerzálne vetvenie (s dvomi nasledovníkmi) a $|$ znamená „žiadne vetvenie“ — deterministická konfigurácia s jedným nasledovníkom. Popis trasy sa zmestí do priestoru $O(T(n))$.

Po odsimulovaní $T(n)$ krokov ($T(n)$ je časove konštruovateľná funkcia) stroja A začnú procesy M overovať podmienku akceptovania v A. Táto podmienka vyjadruje, ako sa majú skombinovať „odpovede“ jednotlivých ukončených procesov, keď stúpame smerom ku koreňu T_A (pozri napr. [1]). Pri realizovaní „stúpania“ využijeme informáce z popisu trasy. Postúpiť z úrovne d na úroveň $d-1$ môžme len vtedy, ak majú všetky procesy M „vyplnenú“ hodnotu $qual_d$.

Odteraz $S_{\langle b_1,\ldots,b_d\rangle}$ označuje množinu procesov M, ktorých číslo cesty začína $(b_1, \ldots, b_d)_2$. Nezabúdajme, že hodnota $\Box_i$ (pre $i = 1, \ldots, d,$) je pre všetky procesy z $S_{\langle b_1,\ldots,b_d\rangle}$ rovnaká.

V každom procese M vypočítame hodnoty $qual_d,\ d = T(n), T(n)-1, \ldots, 1$ rovnakým algoritmom. Algoritmus teda pozostáva z $T(n)$ kôl, pričom na konci každého kola d platí invariant: každý proces z $S_{\langle b_1,\ldots,b_d\rangle}$, má určené všetky hodnoty $qual_i,\ i = T(n), T(n)-1, \ldots, d$ a navyše, hodnota $qual_d$ je vo všetkých procesoch z $S_{\langle b_1,\ldots,b_d\rangle}$ rovnaká. (Nezabúdajme, že všetky procesy WPTM vykonávajú jednotlivé kroky algoritmu samostatne, ale naraz.)

Výpočet algoritmu začína v hĺbke $d = T(n)$; hodnota $qual_d$ sa v každom procese M nastaví na Y, N, alebo $\perp$ podľa toho, či proces akceptoval, zamietal alebo ešte neskončil výpočet. Keďže množiny $S_{\langle b_1,\ldots,b_d\rangle}$ sú pre všetky čísla ciest $b_1, \ldots, b_d$ jednoprvkové, invariant platí.

Predpokladajme, že invariant platí v hĺbke $T(n) \geq d > 1$; popíšeme algoritmus, ktorý zachová jeho platnosť aj pre d rovné $d-1$.

Každý proces z $S_{\langle b_1,\ldots,b_d\rangle}$ skontroluje hodnotu $\Box_d$ na svojej trase.

Hodnota „|“ odpovedá deterministickému kroku; v tomto prípade proces nastaví $qual_d := qual_{d+1}$.

Hodnota „$\forall$" resp „$\exists$" poukazuje na univerzálne, resp. existenčné vetvenie. V rámci $S_{\langle b_1,\ldots,b_d\rangle} = S$ preto existujú dve „sesterské" podmnožiny procesov $S_{\langle b_1,\ldots,b_d,0\rangle} = S_1$ a $S_{\langle b_1,\ldots,b_d,1\rangle} = S_2$. Množiny S_1 a S_2 sú disjunktné, $S_1 \cup S_2 = S$, všetky procesy v $S_i,\ i = 1, 2$, majú rovnakú hodnotu $qual^i_{d+1}$. Aby mohol proces z S vypočítať hodnotu $qual_d = qual^1_{d+1} \wedge qual^2_{d+1}$, resp. $qual_d = qual^1_{d+1} \vee qual^2_{d+1}$, potrebuje obe hodnoty $qual^1_{d+1}, qual^2_{d+1}$ poznať. Preto obe množiny (postupne) odvysielajú hodnotu svojho $qual^i_{d+1}$ všetkým ostatným procesom v S. Vysielanie sa uskutoční na kanáli číslo $(b_1 \ldots b_d)_2$: najprv vysielajú procesy z S_1, potom procesy z S_2. Keďže sú hodnoty $qual^i_{d+1}$ v rámci S_1, resp. S_2, rovnaké, ku konfliktu vo vysielaní nedochádza. Každý proces z S má potrebnú informáciu a môže vypočítať kvalitu príslušného vrchola v hĺbke d v súlade s pravidlami vyhodnocovania alternujúcich stromov (pozri napr. [1]). Je zrejmé, že všetky procesy v S vypočítajú rovnakú hodnotu $qual_d$. Všimnime si, že pre rôzne množiny $S_{\langle b_1,\ldots,b_d\rangle}$ prebieha potrebná komunikácia na rôznych kanáloch. Nastavením hodnoty $d := d - 1$ ďalšie kolo algoritmu končí.

Takto vyhodnocujeme jednu úroveň za druhou až kým pre $d = 1$ nedosiahne $qual_1$ vo všetkých procesoch svoju výslednú hodnotu. Vysielaním tejto hodnoty na kanáli číslo 0 sa všetky procesy dostanú do rovnakého komunikačného stavu q_{accept} alebo q_{reject} a simulácia končí.

Ľahko vidno, že overenie podmienky akceptovania pre A realizuje M v $O(T(n))$ krokoch.

$\square$

Veta 2 *Nech $T(n)$ je časovo konštruovateľná funkcia, $T(n) \geq n$,a M WPTM časovej zložitosti $T(n)$. Potom existuje alternujúci Turingov stroj A, ktorý simuluje M v časovej aj priestorovej zložitosti $O(T^4(n))$.*

Náčrt dôkazu: Namiesto priamej simulácie M alternujúcim Turingovým strojom A ukážeme simuláciu M nedeterministickým Turingovým strojom N v priestore $O(T^2(n))$. Z toho na základe známych výsledkov o alternujúcich Turingových strojoch (napr. [1], Veta 3.1) vyplynie požadovaný výsledok. Navrhneme nedeterministický algoritmus, ktorý odsimuluje M v priestore $O(T^2(n))$.

Nech T_M je výpočtový strom stroja M so vstupom w. Popíšeme priestorovo efektívny nedeterministický peblovací algoritmus (stroj N) ktorý navštívi každý vrchol T_M. Uvedomme si, že peblovanie môžme použiť vďaka deterministickej povahe výpočtu M. V takých prípadoch je peblovanie jednou z možných stratégií modelovania priestorovo efektívneho prepočitávania prechádzaných konfigurácií (napr. [4]).

T_M je superpozíciou dvoch grafov: prvý z nich je binárny strom —budeme mu hovoriť *strom nasledovníkov*, ktorý znázorňuje históriu vzniku jednotlivých procesov. Druhý, tzv. *graf vysielania*, zachytáva vysielanie komunikačných stavov z konfigurácií do odpovedajúcich prijímajúcich konfigurácií.

Musíme zabezpečiť, že algoritmus navštívi každý vrchol T_M a v rámci tejto návštevy overiť, že príslušná konfigurácia bola vypočítaná správne. Tiež treba skontrolovať, že v T_M nedošlo k vysielacím konfliktom a že je splnená podmienka akceptovania/zamietania vstupu. K dosiahnutiu stanovených cieľov navrhneme nedeterministickú peblovaciu stratégiu.

Peblovanie pozostáva z dvoch častí: cieľom prvej fázy je položiť kameň na každý list T_M, v druhej časti sa kontroluje bezkonfliktnosť vysielania v T_M.

V prvej fáze kladieme kamene na vrcholy T_M podľa nasledujúcich pravidiel. Kameň sa vždy dá položiť na koreň T_M. Kameň vždy môžeme odstrániť z ľubovoľného vrchola. Kameň sa dá položiť na vrchol, ak je kameň položený na jeho bezprostrednom predchodcovi v strome nasledovníkov; v prípade, že je do pokrývaného vrchola vysielaný komunikačný stav, musí navyše obsahovať kameň aj jeden z jeho predchodcov v grafe vysielania. Cieľom tejto fázy je pokryť každý list T_M.

V druhej fáze skontrolujeme, že pri vysielaní nedošlo v T_M ku konfliktom. Spravíme to takto: pre všetky obsahy kanálovej pásky dĺžky $c : 0 \leq c \leq T(n)$ a pre každú hĺbku overíme, či všetky vysielajúce konfigurácie v T_M naladené na c vysielajú rovnaký komunikačný stav. Cieľom hry v tejto fáze je prechádzať stromom T_M jednu úroveň za druhou a podľa vyššie naformulovaných pravidiel postupne pokrývať všetky ich vrcholy.

Jedine v prípade, že prvá fáza ohlásila akceptujúci stav v každej terminálnej konfigurácii a všetky tieto konfigurácie sú naladené na rovnaký kanál a v priebehu druhej fázy sa nezistil žiaden vysielací konflikt, dochádza k tomu, že N akceptuje w.

Nedeterministická peblovacia stratégia je kombináciou dvoch algoritmov: deterministického, ktorý systematicky prechádza stromom nasledovníkov a nedeterministického, ktorý kladie kamene na vrcholy stromu s cieľom položiť kameň na konkrétny vrchol v T_M, ktorý bol určený deterministickým algoritmom. Kroky nedeterministického peblovacieho algoritmu sa generujú on–line, nie je potrebné si ich pamätať.

V prvej fáze prechádzame strom nasledovníkov prehľadávaním do hĺbky, v druhej zas prehľadávaním do šírky. Ak v tomto prehľadávaní neboli navštívené všetky vrcholy, prehľadávací algoritmus jednoznačne určí nový ďalší nenavštívený vrchol v, ktorý sa potom musí pokryť nedeterministicky.

Odhadnime počet kameňov, postačujúcich na realizáciu popísanej hry. Aby sme položili kameň na vrchol v stromu T_M v hĺbke d, musíme položiť kameň na dvoch jeho priamych predchodcov. Prvým je vrchol v_1 v strome nasledovníkov, z ktorého, s výnimkou komunikačného stavu, vieme vypočítať konfiguráciu uloženú vo vrchole v. Tento chýbajúci stav môžme doplniť prostredníctvom druhého vrchola v_2 v grafe vysielania. Oba vrcholy v_1 a v_2 ležia v hĺbke $(d-1)$. Označme $P(d)$ počet kameňov postačujúcich na pokrytie vrchola v stromu T_M v hĺbke d. Najprv — použitím $P(d-1)$ kameňov — pokryjeme vrchol v_1, necháme kameň na v_1 a odstránime (pre ďalšie použitie) $P(d-1)-1$ kameňov použitých pri pokrývaní v_1. Potom pokryjeme pomocou $P(d-1)$ kameňov vrchol v_2. Tentokrát však môžeme využiť $P(d-1)-1$ kameňov, ktoré sme uvoľnili po pokrytí vrchola v_1. Na pokrytie v_2 teda stačí jeden kameň navyše. Keďže jeden kameň je umiestnený na v_1, $P(d) \leq P(d-1)+1$ a $P(1) = 1$, platí $P(d) = d$. Preto $T(n)$ kameňov v každej fáze pokrývacej hry stačí.

Keďže kameň v pokrývacej hre popísaného algoritmu je v skutočnosti konfigurácia M veľkosti $O(T(n))$, je priestorová zložitosť popísaného nedeterministického simulačného algoritmu $O(T^2(n))$.

□

V nasledujúcej časti sa budeme venovať zaradeniu WPTM do hierarchie známych zložitostných tried. Navyše k štandardným deterministickým triedam ako LOGSPACE, PSPACE a EXPTIME budeme používať ich analógie pre ATM, resp. WPTM. Označenie príslušných zložitostných tried vznikne pridaním prefixu A resp. W k už spomínaným triedam.

Dôsledok 1 *Pre každú konštruovateľnú funkciu $T(n)$, $T(n) \geq n$,*

$$\bigcup_{k>0} \mathsf{WTIME}(\mathsf{T}^{\mathsf{k}}(\mathsf{n})) = \bigcup_{k>0} \mathsf{ATIME}(\mathsf{T}^{\mathsf{k}}(\mathsf{n}))$$

t.j.,

$$\mathsf{WPTIME} = \mathsf{APTIME} = \mathsf{PSPACE}$$

Ďalej sa budeme zaoberať vzťahom medzi deterministickým a bezdrôtovým mobilným priestorom.

Veta 3 *Nech D je deterministický Turingov stroj priestorovej zložitosti $S(n)$. Potom D sa dá simulovať WPTM M kanálovej zložitosti $O(\log S(n))$ a priestorovej zložitosti $O(1)$.*

Náčrt dôkazu: Dôkaz postupuje podobne ako dôkazy podobných tvrdení o synchronizovanom alternovaní (napr. [2], [6]). Obsahy jednotlivých buniek jedinej pracovnej pásky DTM D sa udržiavajú v procesoch M. Vždy, keď hlava D dosiahne na pracovnej páske prázdny symbol, vytvorí M nový proces. Procesy sú číslované pozíciou bunky, aktuálny obsah ktorej udržiavajú. Na svojej pracovnej páske si teda proces číslo i pamätá symbol zapísaný v i-tej bunke, prítomnosť hlavy D na nej (yes / no), a stav stroja D v prípade, ak hlava sníma i–tu bunku. Číslo i (zapísané binárne) si i–ty proces uchováva na svojej kanálovej páske. Simulácia jedného kroku výpočtu D znamená aktualizáciu informácie (obsah, resp. stav) v tých procesoch M, ktoré odpovedajú bunkám zúčastneným v tomto kroku. „Pohyb hlavy" z i–teho procesu (bunky) sa realizuje vyslaním príslušnej „správy" ľavému, resp. pravému susedovi bunky na kanáli, ktorého číslo sa zhoduje s indexom príslušnej bunky — $(i-1)$ resp. $(i+1)$. Preladenie kanála teda spočíva v pripočítaní alebo odpočítaní 1 od aktuálnej hodnoty čísla kanála. Kanálová zložitosť je zrejme $O(\log S(n))$, priestorová $O(1)$.

□

Veta 4 *Nech $S(n) \geq \log n$, M WPTM priestorovej a kanálovej zložitosti $O(S(n))$. Potom existuje $c > 0$ také, že M sa dá simulovať deterministickým Turingovým strojom D v priestore $O(c^{S(n)})$.*

Náčrt dôkazu: Majme výpočtový strom T_M WPTM M so vstupom w. Ak M akceptuje w, tak M môže mať nanajvýš $O(c_1^{S(n)})$ rôznych procesov. Keďže priestorová zložitosť každého z nich je ohraničená $O(S(n))$ (uvedený priestor je postačujúci aj pre pozíciu hlavy na vstupe), tak pre vhodnú konštantu $c > c_1$ má D dosť miesta na to, aby zapísal všetky potrebné konfigurácie a krok za krokom simuloval výpočet M. D akceptuje, ak jeho výpočet odpovedá výpočtovému stromu M akceptujúcemu w.

$\square$

Dôsledok 2 *Pre každú funkciu $S(n) \geq \log n$*

$$\mathsf{WSPACE}(\mathsf{S}(\mathsf{n})) = \bigcup_{\mathsf{c}>0} \mathsf{DSPACE}(\mathsf{c}^{\mathsf{S}(\mathsf{n})})$$

$\mathsf{WLOGSPACE} = \mathsf{PSPACE} = \mathsf{WPTIME}$, $\mathsf{WPSPACE} = \mathsf{EXPSPACE} = \mathsf{WEXPTIME}$

Dôsledky 1 a 2 ukazujú, že základné zložitostné triedy — logaritmický priestor a polynomiálny čas — sú pre synchronizované alternujúce Turingove stroje (podľa [6] alebo [3]) a sieť Turingových strojov totožné. Zvlášť si všimnime, že podobne ako synchronizované Turingove stroje, WPTS používa svoj priestor optimálnym spôsobom — napr. logaritmický „bezdrôtový“ priestor je ekvivalentný „bezdrôtovému“ polynomickému času. Vzhľadom na základné triedy klasickej priestorovej zložitosti sú teda obidva modely ekvivalentné.

4 Záver

Okrem charakterizácie výpočtovej sily bezdrôtových sietí sme získali aj nový pohľad na podstatu klasického výpočtového zdroja — alternácie, resp. synchronizovanej alternácie. Hoci (synchronizované) alternujúce stroje so svojím neobmedzeným využívaním nedeterminizmu a zvláštnym mechanizmom akceptovania pôsobia ako pomerne nerealistický výpočtový model, získané výsledky ukazujú, že sú vlastne ekvivalentné deterministickým paralelným zariadeniam so schopnosťou šírenia správ na neobmedzenom počte rôznych kanálov. Tieto výsledky stavajú bezdrôtové mobilné siete medzi veľmi silné výpočtové zariadenia.

Literatúra

1. Chandra, A., Kozen, D., Stockmeyer, L.: Alternation. Journal of the ACM, Vol. 28, pp. 114–133, 1981.
2. Geffert, V.: A Communication Hierarchy of Parallel Computations. Theor. Comput. Sci., Vol 198, No.1–2, pp. 99–130, 1998
3. Hromkovič, J., Karhumäki, J., Rovan, B., Slobodová, A.: On the power of synchronization in parallel computations. Discrete Appl. Math. 32 (1991), 155-182
4. Hopcroft, J. E., Paul, J.W., Valiant, L.:. On time versus space. Journal of ACM, Vol. 15, 1968, 414–427.
5. Valiant, L. G.: A Bridging Model for Parallel Computation, Communication of the ACM, Vol. 33 No. 8, pp. 103–111, 1990
6. Wiedermann, J.: On the Power of Synchronization. Elektronische Informationsverarbeitung und Kybernetik 25(10): 499–506 (1989)