



KATEDRA INFORMATIKY
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO, BRATISLAVA

SYNCHRONIZÁCIA CELULÁRNYCH AUTOMATOV

(diplomová práca)

Autor: Marek Jančuška

Vedúci: RNDr. Borislav Šuster, CSc.

Bratislava, 2009

Čestne prehlasujem, že som túto diplomovú prácu
vypracoval samostatne s použitím citovaných zdro-
jov.

.....

Ďakujem vedúcemu mojej diplomovej práce RNDr. Borislavovi Šusterovi, CSc. za jeho rady a pripomienky pri písaní tejto práce.

Abstrakt

Celulárny automat je výpočtové zariadenie pozostávajúce zo vzájomne prepojených konečných automatov, nazývaných bunky. Synchronizácia vychádza z konfigurácie, keď je aktívna jediná bunka zvaná generál, a cieľom je, aby všetky bunky naraz vykonali určenú akciu.

V práci sú uvedené známe výsledky pre synchronizáciu úsečky, kruhu, jednosmerného kruhu, dvojrozmerných sietí a stromu. Ďalej uvádzame náš výsledok – synchronizáciu stromu pre špeciálnu pozíciu generála v čase $(2 + \frac{1}{2})r$. Nakoniec vyslovujeme hypotézu o možnom vylepšení dosiahnutého výsledku.

Kľúčové slová: synchronizácia, celulárne automaty, stromy

Abstract

The cellular automaton is a computational device consisting of interconnected finite automata, called cells. The synchronization starts in a configuration with the only active cell called general; the goal is that all cells do some action simultaneously.

In this thesis we present known results of synchronization of lines, rings, unidirected rings, two-dimensional networks and trees. Next we present our result – the synchronization of trees for a special position of the general, working in time $(2 + \frac{1}{2})r$. Finally we hypothesize about possible improvements of our result.

Keywords: synchronization, cellular automata, trees

Obsah

1	Úvod	7
2	Základné pojmy	9
2.1	Celulárny automat	9
2.2	Synchronizácia	12
2.3	Simulácia	12
2.4	Vnорiteľné riešenia	13
3	Známe výsledky	15
3.1	Jednorozmerné CA	15
3.1.1	Pôvodné riešenie	15
3.1.2	Riešenie s minimálnym časom	16
3.1.3	Riešenie s min. časom a 6 stavmi	18
3.1.4	Zovšeobecnenie - generál na ľubovoľnom mieste	18
3.1.5	Kruhy	21
3.1.6	Jednosmerné kruhy	22
3.2	Množina riešení FSSP nie je rekurzívna	27
3.3	Dvojrozmerné CA	28
3.3.1	Veta o dolnom časovom odhade	29
3.3.2	Štvorec	30
3.3.3	Obdĺžnik	31
3.3.4	Kocka	31
3.3.5	Ďalšie obrazce	32
3.3.6	Všeobecné obrazce	33
3.4	Synchronizácia všeobecného stromu	34

4	Synchronizácia stromov	38
4.1	Definície	38
4.2	Generál na kraji hlavnej vetvy, čas $(2 + \frac{1}{2})r$	39
4.2.1	Zaokrúhľovanie	44
4.3	Generál vnútri hlavnej vetvy	49
4.4	Vnoriťnosť	55
5	Možnosti vylepšenia	58
5.1	Synchronizácia stromu v čase blížiacom sa k $(2 + \frac{1}{4})r$	58
5.2	Synchronizácia stromu v čase $(2 + \frac{1}{4})r$	62
6	Záver	63
	Literatúra	64

Kapitola 1

Úvod

Významnou oblasťou informatiky sú paralelné výpočty. Praktickou motíváciou je napríklad vývoj v oblasti mikroprocesorov, kde sa výrobcovia už nesnažia zvýšiť taktovaciu frekvenciu, ale zvyšujú počet jadier procesora. Ďalšou aplikáciou paralelizmu je návrh VLSI obvodov.

Cieľom bude modelovať zariadenie s veľkým množstvom malých a rýchlych procesorov, pričom predpokladáme prítomnosť globálnych synchronizačných taktov. Od komunikačných protokolov sa očakáva veľká rýchlosť. Z tohto dôvodu nemôžu procesory použiť zdieľanú pamäť kvôli jej pomalosti. Teda procesor má k dispozícii len lokálnu pamäť a komunikačné linky so susedmi. Preto budeme jednotlivé procesory modelovať pomocou konečných automatov, takže množstvo použitej pamäte na jeden procesor bude konštantné.

Na takomto výpočtovom modeli budeme riešiť problém synchronizácie. Úlohou je, aby všetky procesory vykonali nejakú akciu súčasne, pričom na začiatku je prebudený len jeden procesor. Vyriešený problém synchronizácie môže slúžiť ako podprogram pri riešení zložitejších úloh.

Problém synchronizácie bol skúmaný pre rôzne topológie siete. Náš výsledok spočíva v nájdení lepšieho riešenia pre acyklické siete.

Zvyšok práce je členený nasledovne. V kapitole 2 zdefinujeme potrebné pojmy. V kapitole 3 zhrnieme známe výsledky o synchronizácii celulárnych automatov. Hlavným prínosom tejto práce je kapitola 4, kde sa venujeme

acyklickým sieťam. Ukážeme, že pre vhodnú pozíciu generála je možné dosiahnuť synchronizáciu rýchlejšie ako pri použití známeho všeobecného riešenie. V kapitole 5 načrtujeme, ako by sa dosiahnuté výsledky dali zlepšiť.

Kapitola 2

Základné pojmy

2.1 Celulárny automat

Celulárny automat je výpočtové zariadenie. Môžeme si ho predstaviť ako orientovaný graf, pričom v každom vrchole grafu je umiestnený konečný automat. Vrchol grafu spolu s príslušným konečným automatom budeme nazývať bunka celulárneho automatu. Namiesto „stav konečného automatu v bunke p ” budeme hovoriť len „stav bunky p ”. Celulárny automat je homogénny, teda konečný automat v každej bunke má rovnakú prechodovú funkciu. Prechodová funkcia λ konečného automatu má na vstupe stav bunky p a stav všetkých jej susedov v čase t , výstupom λ je stav bunky p v čase $t + 1$. Celulárny automat pracuje synchronne, teda prechodová funkcia je aplikovaná na všetky bunky naraz. Keďže v bunkách sú konečné automaty, obmedzíme zhora počet susedov bunky konštantou a ; budeme hovoriť, že celulárny automat je typu a .

Na zjednodušenie označovanie zavedieme špeciálny stav s_E (external); ak má bunka menej ako a susedov, doplníme bunke ako susedov „zarážky” v podobe automatov, ktoré sú trvalo v stave s_E .

Pri opise celulárneho automatu nebude stačiť pojem graf. V grafe môžeme pre každý vrchol hovoriť len o množine jeho susedov. Budeme však potrebovať týchto susedov aj očíslovať, teda pre daný vrchov u povedať, ktorý vrchol v je i -ty sused vrchola u . Kvôli tomu zavedieme pojem *sieť*. Sieť

sa dá chápať ako graf s pridanou informáciou o číslovaní susedov.

Budeme hovoriť, že bunka má vstupné porty. Ak bunka q je i -ty sused bunky p , tak bunka p vidí na svojom i -tom vstupnom porte stav bunky q . Ak bunka p nemá i -teho suseda, na svojom i -tom vstupnom porte vidí stav s_E .

Popis celulárneho automatu rozdelíme na popis siete (hovorí o tom, ako sú bunky prepojené) a popis konečného automatu (hovorí o tom, ako sa bunky správajú).

Definícia 2.1.1. Pod *sietou typu a* budeme rozumieť trojicu $N = (P, p_G, c)$, kde P je konečná množina buniek, $p_G \in P$ je generál a c je čiastočná funkcia $P \times \{0, \dots, a-1\} \rightarrow P$. Ak $c(p, i) = q$, tak i -ty vstupný port bunky p je spojený s bunkou q . Nedefinovaná hodnota $c(p, i)$ znamená, že i -ty vstupný port bunky p nie je zapojený. Množinu $\{c(p, i) \mid 0 \leq i \leq a-1 \wedge c(p, i) \text{ je definované}\}$ budeme nazývať *susedia* bunky p . Sieť nazveme *symetrická*, ak $c(p, i) = q \Rightarrow (\exists j)c(q, j) = p$.

Aby sme pri opise sietí mohli využiť pojmy z teórie grafov, ku každej sieti priradíme prislúchajúci graf. Neformálne povedané, zabudneme na očíslovanie susedov. Formálne, sieť $N = (P, p_G, c)$ prislúcha orientovaný graf $G = (V, E)$, kde $V = P$ a $(p_1, p_2) \in E \iff (\exists i)c(p_2, i) = p_1$. Ak sieť N je symetrická, má zmysel hovoriť o neorientovanom grafe G' prislúchajúcom N – ten dostaneme z G tak, že zabudneme orientáciu hrán. Takže napr. tvrdenie „sieť N je strom“ treba chápať ako „neorientovaný graf prislúchajúci sieti N je strom“.

Definícia 2.1.2. *Automat typu a* je šesticca $A = (S, s_Q, s_G, s_F, s_E, \lambda)$, kde S je konečná množina stavov, $s_Q \in S$ je pokojný stav (quiescent state), $s_G \in S \setminus \{s_Q\}$ je stav generála na začiatku, $s_F \in S \setminus \{s_Q, s_G\}$ je koncový stav, $s_E \notin S$ je externý stav (reprezentuje vstup automatu, ak príslušný vstupný port nie je zapojený), $\lambda : S^{a+1} \rightarrow S$ je prechodová funkcia. Pokojný stav s_Q má tú vlastnosť, že ak je bunka aj všetci jej susedia v stave s_Q , tak bunka ostáva v stave s_Q ; teda pre prechodovú funkciu musí platiť $\{s_1, s_2, \dots, s_a\} \subseteq \{s_Q, s_E\} \Rightarrow \lambda(s_Q, s_1, s_2, \dots, s_a) = s_Q$.

Definícia 2.1.3. Nech $N = (P, p_G, c)$ je sieť typu a a $A = (S, s_Q, s_G, s_F, s_E, \lambda)$ je automat typu a . *Celulárnym automatom typu a* budeme rozumieť dvojicu (N, A) .

Stav bunky $p \in P$ v čase $t \in \mathbb{N}$ v celulárnom automate (N, A) budeme značiť $s(p, t, N, A)$.

Prechod celulárneho automatu môžeme formálne opísať nasledovne:

$$s(p, t + 1, N, A) = \lambda(s(p, t, N, A), s_0, \dots, s_{a-1})$$

$$\text{kde } s_i = \begin{cases} s(c(p, i), t, N, A) & \text{ak } c(p, i) \text{ je definované} \\ s_E & \text{inak} \end{cases} \quad \text{pre } i \in \{0, \dots, a-1\}$$

Definícia 2.1.4. Prvý prechod bunky zo stavu s_Q do iného stavu nazveme *prebudenie* bunky. Teda bunka ostáva v stave s_Q , kým sa neprebudí. *Čas prebudenia* bunky p je najmenšie t_W také, že $s(p, t_W, N, A) \neq s_Q$.

Zavedieme označenie $\text{Grid}(P)$ pre sieť (typu 4) tvaru mriežky, kde P je konečná podmnožina \mathbb{Z}^2 a $(0, 0) \in P$:

$$\text{Grid}(P) = (P, p_g = (0, 0), c_{\text{Grid}})$$

$$c_{\text{Grid}}((x, y), i) = \begin{cases} c'_{\text{Grid}}((x, y), i) & \text{ak } c'_{\text{Grid}}((x, y), i) \in P \\ \text{nedefinované} & \text{inak} \end{cases}$$

$$c'_{\text{Grid}}((x, y), 0) = (x - 1, y)$$

$$c'_{\text{Grid}}((x, y), 1) = (x, y - 1)$$

$$c'_{\text{Grid}}((x, y), 2) = (x + 1, y)$$

$$c'_{\text{Grid}}((x, y), 3) = (x, y + 1)$$

Poznamenajme, že v definícii $\text{Grid}(P)$ sme implicitne zahrnuli zmysel pre orientáciu.

Pri synchronizácii stromov bude užitočné považovať generála p_G za koreň stromu, preto v definícii 2.1.5 zavedieme príslušné pojmy.

Definícia 2.1.5. Nech graf G je strom. Nech p je vrchol stromu. Nech $d(G, p) = l$. Potom tých susedov vrchola p , ktorých vzdialenosť od G je $l + 1$, budeme nazývať *synovia* p . Nech navyše $l \geq 1$. Potom ten sused vrchola p , ktorého vzdialenosť od G je $l - 1$, je *rodič* p . *Potomok* vrchola p je každý syn vrchola p a každý syn potomka vrchola p , tj. relácia „potomok” je tranzitívny uzáver relácie „syn”.

Nech P_p je množina obsahujúca p , všetkých potomkov vrchola p v grafe G a nič iné. Potom podgraf G indukovaný množinou P_p budeme nazývať *podstrom* prislúchajúci vrcholu p v grafe G a značiť $\text{subtree}_G(p)$. Keď bude jasné, o ktorom grafe hovoríme, budeme značiť len $\text{subtree}(p)$.

Hĺbku stromu G s koreňom p budeme značiť $\text{depth}(G, p)$. Špeciálne namiesto $\text{depth}(\text{subtree}(p), p)$ budeme písať len $\text{depth}(\text{subtree}(p))$.

2.2 Synchronizácia

Problém synchronizácie (resp. problém strelcov, firing squad synchronization problem, FSSP) je nasledujúca úloha. Je daná trieda sietí Γ . Treba navrhnúť konečný automat A tak, aby na každej sieti $N \in \Gamma$ celulárny automat (N, A) fungoval nasledovne. Na začiatku je práve jedna bunka (generál) v stave s_G , ostatné sú v pokojnom stave s_Q . Cieľom je, aby všetky bunky prešli naraz do stavu s_F , tento prechod budeme nazývať výstrel. Teda všetky bunky musia byť v čase t_F v stave s_F a žiadna bunka nesmie byť v čase $t < t_F$ v stave s_F . Formálne $t < t_f \Rightarrow s(p, t, N, A) \neq s_F, s(p, t_f, N, A) = s_F$.

Čas výstrelu pre celulárny automat (N, A) , tj. čas, keď všetky bunky prejdú do stavu s_F , budeme označovať $t_F(N, A)$. Čas výstrelu je nedefinovaný, ak A nie je riešením FSSP pre sieť N , teda ak niektoré bunky vystrelia a iné nie. Ak žiadna bunka nevystrelí, bude $t(N, A) = \infty$.

2.3 Simulácia

Aby sme mohli z jednoduchších riešení skladať zložitejšie, zavedieme pojem simulácie.

Definícia 2.3.1. Budeme hovoriť, že celulárny automat (N, A) *simuluje* celulárny automat (N', A') so začiatkom v čase t_0 , tj. že (N, A) je simulujúci (alebo reálny) a (N', A') je simulovaný (alebo virtuálny), ak existuje $d \in \mathbb{N}$, funkcia $f : P' \rightarrow P \times \{1, \dots, d\}$ a čiastočná funkcia $g : S \times \{1, \dots, d\} \rightarrow S'$ také, že

$$s(p', t, N', A') = g(s(p, t + t_0, N, A), i)$$

kde $f(p') = (p, i)$.

Funkcia f pre danú virtuálnu bunku $p' \in P'$ určuje, ktorá reálna bunka $p \in P$ ju bude simulovať. Funkcia g prekladá stavy reálneho automatu na zodpovedajúce stavy virtuálneho automatu. Keďže jedna reálna bunka môže simulovať viac virtuálnych buniek, druhým argumentom g je identifikátor požadovanej virtuálnej bunky. Tento identifikátor zodpovedá druhému členu dvojice, ktorú vracia funkcia f . Číslo d udáva, koľko najviac virtuálnych buniek môže simulovať 1 reálna bunka. (Např. nech bunka p simuluje bunky p'_1, p'_2 . Toto môžeme opísať nasledovne: $f(p'_1) = (p, 1)$, $f(p'_2) = (p, 2)$. Nech pre jednoduchosť $S = S' \times S'$. Potom jedna možnosť pre g je $g((s'_1, s'_2), 1) = s'_1$, $g((s'_1, s'_2), 2) = s'_2$. Stačí nám $d = 2$.)

Pri simulácii budeme vo všeobecnosti potrebovať, aby reálne bunky boli susedné, ak sú príslušné virtuálne bunky susedné. Teda ak $p'_1, p'_2 \in P'$ sú susedné v sieti N' a $f(p'_1) = (p_1, i_1)$, $f(p'_2) = (p_2, i_2)$, tak p_1 a p_2 sú susedné v sieti N .

2.4 Vnoriteľné riešenia

Ďalšou metódou na konštrukciu riešení FSSP je používanie vnoriteľných (embeddable) riešení. Vychádzame z článku [2]. Rovnaká myšlienka je použitá aj v [4], ale pod názvom pseudoriešenie (pseudosolution).

Majme konečné automaty A_1, \dots, A_n a triedy sietí $\Gamma_1, \dots, \Gamma_n$ také, že A_i je riešením pre Γ_i . Cieľom je pomocou automatov A_i skonštruovať riešenie A pre triedu $\Gamma = \bigcup_{i=1}^n \Gamma_i$. Pritom požadujeme, aby A bolo čo najrýchlejšie, teda aby $(\forall N \in \Gamma) t_F(N, A) = \min\{t_F(A_i, N) \mid N \in \Gamma_i\}$. Ako uvidíme, toto vieme za istých okolností dosiahnuť takým automatom A , ktorý bude simulovať

automaty A_1, \dots, A_n a vystrelí vtedy, keď aspoň jeden zo simulovaných automatov vystrelí.

Takto skonštruované A však vo všeobecnosti nebude riešením FSSP. Problém je v tom, že nevieme, ako sa automat A_i správa na sieťach $\Gamma \setminus \Gamma_i$. Preto môže nastať prípad, že v celulórnom automate (N, A_i) , kde $N \in \Gamma \setminus \Gamma_i$, niektoré bunky vystrelia a iné nie. Potom aj v celulórnom automate (N, A) môže nastať prípad, že niektoré bunky vystrelia a iné nie, a teda A nie je riešením pre N .

Aby sme sa tomuto problému vyhli, budeme pripúšťať len také riešenia A_i , ktoré na „zlých“ sieťach $\Gamma \setminus \Gamma_i$ vôbec nevystrelia. Teda budeme požadovať, aby pre každé $N \in \Gamma \setminus \Gamma_i$ bolo $t_F(N, A_i) = \infty$. Takéto riešenia budeme nazývať vnoriteľné.

Definícia 2.4.1. Označme Γ^a triedu všetkých silne súvislých sietí typu a . Nech $\Gamma \subseteq \Gamma^a$. Konečný automat A typu a nazveme *vnoriteľným riešením* pre Γ , ak

1. A je riešením pre Γ
2. $(\forall N \in \Gamma^a \setminus \Gamma) t_F(N, A) = \infty$

Ak horeuvedené automaty A_i sú vnoriteľnými riešeniami pre triedy Γ_i , tak A je vnoriteľné riešenie pre triedu Γ .

Kapitola 3

Známe výsledky

3.1 Jednorozmerné CA

3.1.1 Pôvodné riešenie

Budeme predpokladať, že generál je na ľavom kraji. Pôvodné riešenie vyžaduje čas $3n$. Hlavnou myšlienkou riešenia je rozdeliť úsečku na dve polovice. V strede vznikne nový generál a ďalej sa postupuje rekurzívne – obe polovice na pokyn nového generála nájdu svoj stred, v bodoch $n/4$ a $3n/4$ vzniknú noví generáli atď. Keď každá bunka je generál, nastáva prechod do stavu s_F . Bunka vystrelí vtedy, keď je v stave s_G a jej susedia sú tiež v stave s_G .

Hľadanie stredu napravo od generála prebieha tak, že generál vyšle doprava dva signály S_1, S_2 . (To isté sa deje aj zrkadlovo obrátene, pri hľadaní stredu naľavo od generála.) Signál S_1 postupuje rýchlosťou 1 bunka za sekundu, signál S_2 je 3-krát pomalší. Keď S_1 dorazí na pravý okraj, „odrazí sa“, tj. vyšle sa doľava signál S'_1 , rovnako rýchly ako S_1 . S'_1 a S_2 sa stretnú v strede úsečky, lebo dráha, ktorú prešli signály S_1, S'_1 je rovná trojnásobku dráhy prejdenej signálom S_2 (lebo S_2 je trikrát pomalší).

Signál s rýchlosťou $1/3$ sa dá implementovať pomocou stavov s_1, s_2, s_3 , pričom z s_1 bunka prejde na stav s_2 , z s_2 na s_3 a v stave s_3 sa signál posunie doprava, tj. pravý sused prejde do stavu s_1 .

Uvážme ešte, ako parita čísla n ovplyvní riešenie. Je dôležité, aby po rozdelení mali vzniknuté úsečky presne rovnakú dĺžku, nesmú sa líšiť ani o 1.

Ak je n nepárne, vznikne 1 generál na pozícii $(n - 1)/2$ a úsečka sa rozdelí na $0, \dots, (n - 1)/2$ a $(n - 1)/2, \dots, n - 1$. (Stred $(n - 1)/2$ rátame do oboch nových úsečiek.) Pre párne n vzniknú dvaja generáli na pozíciách $n/2 - 1$ a $n/2$, nové úsečky sú $0, \dots, n/2 - 1$ a $n/2, \dots, n - 1$.

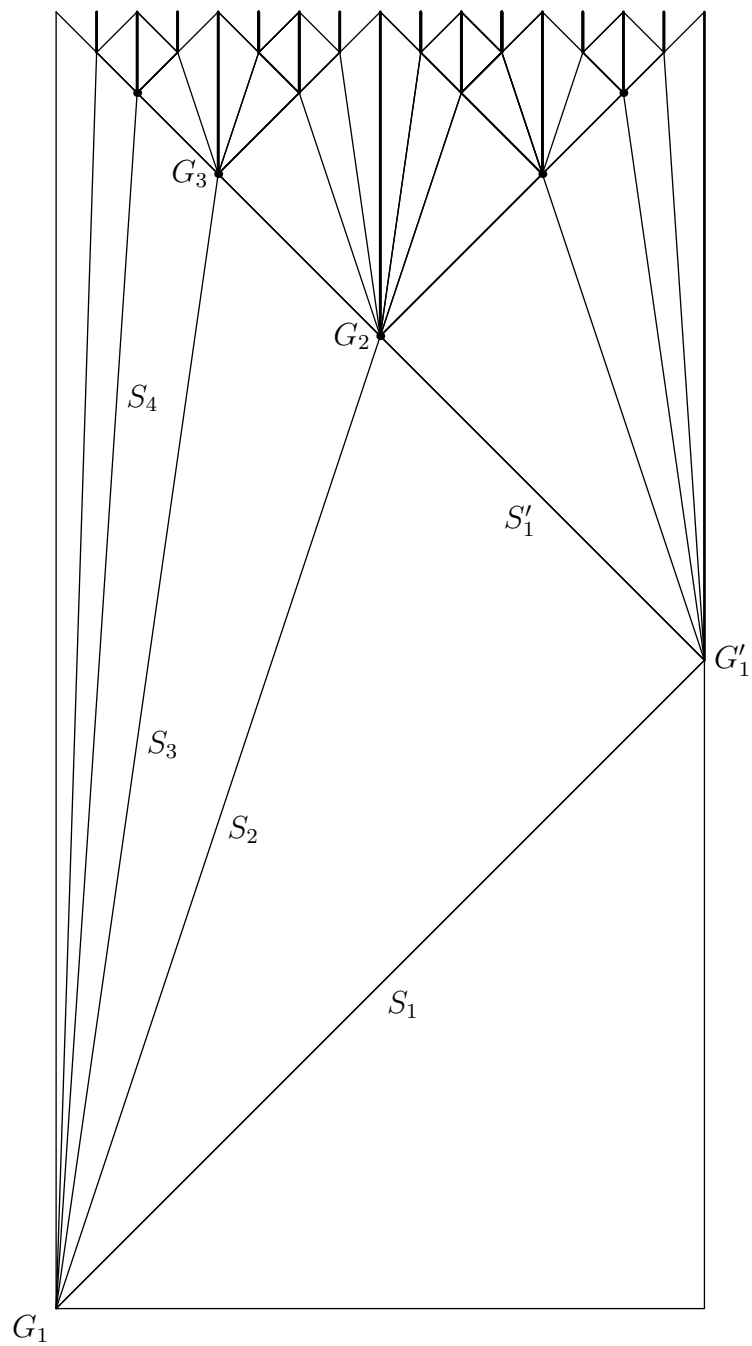
3.1.2 Riešenie s minimálnym časom

Pôvodné riešenie zrýchlime na čas $2n - 2$. Toto riešenie je uvedené v [11] a používa 16 stavov. Riešenie v [1] používa rovnakú myšlienku, ale redukuje počet stavov na 8. Generál bude vysielat' signály $S_1, S_2, \dots, S_i, \dots$ s rýchlosťami $1, 1/3, 1/7, \dots, 1/(2^i - 1), \dots$. Pre $i \geq 2$ sa signál S_i stretne s odrazeným signálom S'_1 v bode $n/2^{i-1}$ (obr. 3.1).

Ďalej vysvetlíme, ako generovať signály s rýchlosťou $1/(2^i - 1)$. Budeme používať pomocné signály, ktoré sa pohybujú doľava (na rozdiel od signálov, ktoré sa pohybujú doprava) rýchlosťou 1. Signál S_1 s rýchlosťou 1 vieme vygenerovať, ten sa pohybuje doprava sám od seba. Každý ďalší signál sa pohne doprava iba vtedy, keď dostane pomocný signál. Každý signál (aj S_1) generuje pomocný signál po každom druhom kroku doprava. Takže kým S_1 dorazí na pravý okraj, posunie sa o $n - 1$ krokov doprava a vyšle doľava $(n - 1)/2$ pomocných signálov. Preto sa signál S_2 od začiatku posunie o $(n - 1)/2$ krokov, teda vyšle doľava $(n - 1)/4$ pomocných signálov. Vo všeobecnosti S_i sa posunie o $(n - 1)/2^{i-1}$ krokov, čo sa dá ľahko ukázať indukciou na i . Keď sa signál S_1 odrazí, už nevysiela žiadne pomocné signály. Pritom odrazený signál S'_1 nemôže „predbehnúť“ žiadny pomocný signál, lebo všetky pomocné signály boli vyslané skôr ako S'_1 a majú rovnakú rýchlosť ako S'_1 . Teda keď sa stretnú S'_1 a S_i , tak S_i už dostal všetky pomocné signály a preto je na správnom mieste (body G_2, G_3 na obr. 3.1).

Na rozdiel od predchádzajúceho riešenia nový generál vznikne nielen v strede úsečky, ale aj na pravom kraji (bod G'_1 na obr. 3.1). Horeuvedené signály S_2, S_3, \dots totiž riešia len ľavú polovicu úsečky, napr. nájdenie bodu $n/4$. Dodatočný generál na pravom okraji pomôže nájsť napr. bod $3n/4$.

Paritu n (resp. vo všeobecnosti zvyšok $n \bmod 2^i$) riešime podobne ako v časti 3.1.1. Pritom stačí využiť informáciu o parite, ktorá je zakódovaná v



Obrázok 3.1: Riešenie v čase $2n - 2$

bunke, ktorá obsahuje signál S_i pre $i \geq 2$ (bunka musí vedieť, či nasledujúci pomocný signál pošle ďalej doľava alebo nie).

3.1.3 Riešenie s min. časom a 6 stavmi

Riešenie uvedené v časti 3.1.2 sa dá implementovať pomocou 8 stavov. Tento počet sa dá znížiť. Všimnime si, že v uvedenom riešení musí generál vo všeobecnosti posilať signály doľava aj doprava. Tieto signály musia byť reprezentované rôznymi stavmi, čo zbytočne zvyšuje počet stavov.

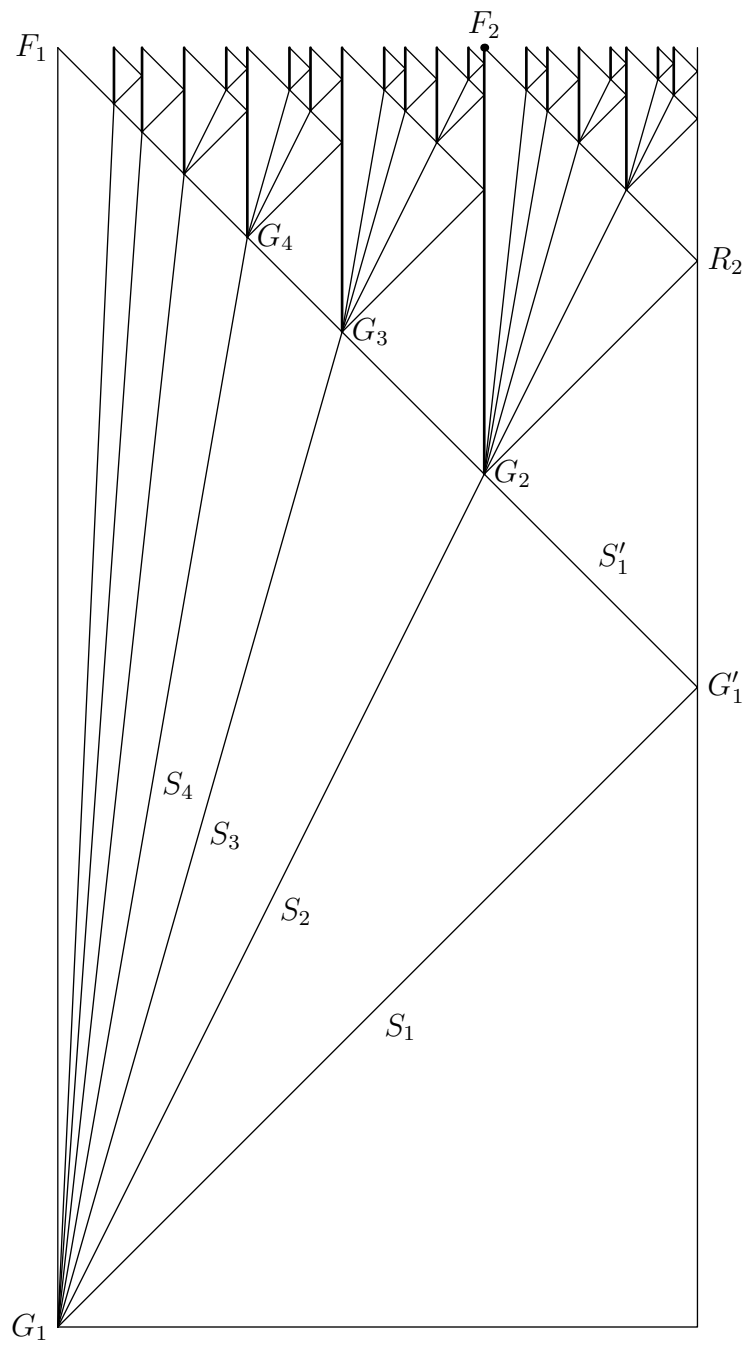
V tejto časti popíšeme odlišný prístup podľa [5], generál bude posilať signály len doprava a úsečka sa bude deliť v pomere 2:1 (nie 1:1). Na rozdiel od časti 3.1.2 každý signál vysiela pomocný signál doľava pri každom pohybe doprava. Signál S_1 sa pohybuje rýchlosťou 1. Každý ďalší signál sa pohne, len ak dostane pomocný signál, pritom ale každý tretí obdržaný pomocný signál ignoruje (tj. ani sa nepohne, ani nevysiela pomocný signál). Nakoniec sa signál S_i dostane na pozíciu $(\frac{2}{3})^i n$, kde sa stretne s odrazeným signálom S'_1 a vytvorí sa tam nový generál v čase $(2 - (\frac{2}{3})^i)n$. Ten rekurzívne zabezpečí zosynchronizovanie časti $A_i = \langle (\frac{2}{3})^{i+1}n, (\frac{2}{3})^i n \rangle$.

Aby sme videli, že všetky automaty vystrelia naraz, všimnime si na obr. 3.2 úsečku G_2F_1 a lomenú čiaru $G_2R_2F_2$. Im zodpovedajúce signály dorazia v rovnakom čase, preto bunky na pozíciách 0 a $(2/3)n$ vystrelia v rovnakom čase. Podobný argument platí pre všetky bunky na pozíciách $(2 - (2/3)^i)n$.

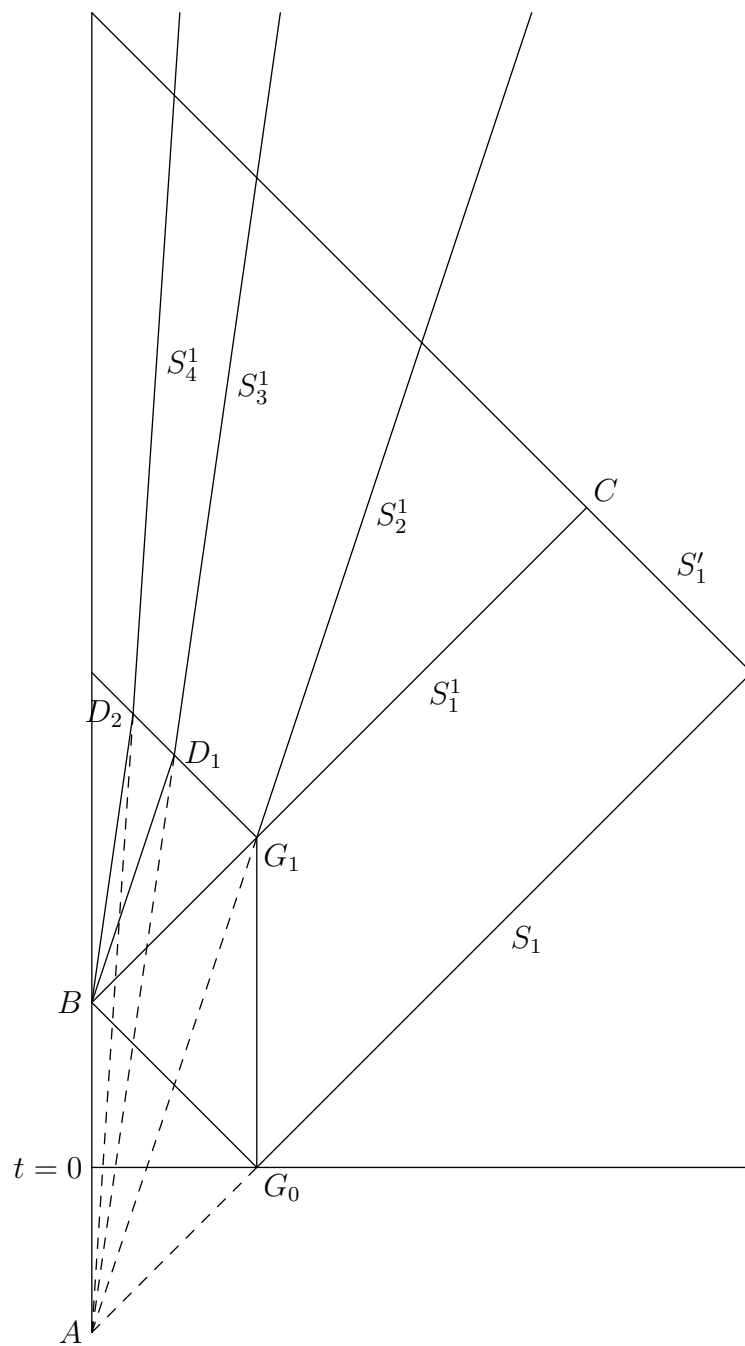
Ešte ostáva vyriešiť správanie automatu pre rôzne hodnoty $i = n \bmod 3$. Nový generál pri svojom vzniku vie zistiť i a začne byť aktívny až po uplynutí i kôl.

3.1.4 Zovšeobecnenie - generál na ľubovoľnom mieste

Nech generál je na k -tej bunke zľava (číslujeme od 0), pričom $k < n/2$. Triviálne riešenie by bolo, keby generál poslal signál doľava, najľavejšia bunka by sa stala novým generálom a ďalej by sa postupovalo ako v časti 3.1.2. To by vyžadovalo čas $2n + k - 2$. My však ukážeme riešenie s časom $2n - k - 2$, podľa [7]. Optimalitu tohto riešenia ukážeme vo vete 3.3.1.



Obrázok 3.2: Riešenie s min. časom a 6 stavmi



Obrázok 3.3: Generál na ľubovoľnom mieste

Budeme vychádzať z riešenia uvedeného v časti 3.1.2. Hlavnou myšlienkou je zrekonštruovať signály S_1, S_2, \dots , aby to vyzeralo tak, ako keby generál bol v bunke číslo 0 a poslal tieto signály v čase $-k$ (obr. 3.3, bod A). Signál S_1 môže generál vyslať hneď. Na vyslanie signálu S_2 musí počkať $2k$ krokov – bod G_1 na obr. 3.3 (keby bol S_2 vyslaný v čase $-k$ z bunky 0, prišiel by do bunky k v čase $-k + 3k = 2k$, viď úsečka AG_1). Preto pošle signál L doľava, tento sa odrazí od ľavého kraja (bod B) a vráti sa práve v čase $2k$. Keď bunka 0 dostane signál L , začne sa správať ako generál, tj. posiela signály S_1^0 (úsečka BG_1), S_2^0 (úsečka BD_1), \dots . Pritom poslaný signál S_1^0 je vlastne odraz L . Všetky ďalšie signály vyslané bunkou 0 treba spomaliť z rýchlosti $1/(2^i - 1)$ na rýchlosť $1/(2^{i+1} - 1)$, to sa deje v bodoch G_1, D_1, D_2 . Spomalenie nastane preto, lebo medzi doterajšie signály S_1^0 a S_2^0 (vyslané bunkou 0) sa v čase $2k$ vloží nový signál S_2^1 . Teda z S_1^0 sa stane S_1^1 (v bode G_1) a z S_2^0 sa stane S_3^1 (v bode D_1), lebo je tretí v poradí. Aby riešenie fungovalo, signál S_1^1 zanikne v bode C , keď sa stretne so signálom S_1^1 .

Zistíme, kde skončí signál S_i^1 pre $i \geq 3$. Od signálu S_i^0 dostane signál S_{i-1}^0 (ako predchodca S_i^1) $k/2^{i-2}$ pomocných signálov. Potom S_i^1 dostane $(n - 2k - 1)/2^{i-1}$ pomocných signálov od S_1^1 , lebo signál S_1^1 prejde z bodu G_1 do bodu C vzdialenosť $n + 2k - 1$. Spolu S_i^0 a S_i^1 dostanú $(2k + n - 2k - 1)/2^{i-1} = (n - 1)/2^{i-1}$ pomocných signálov, čo je správny počet.

Uvažovali sme s $k < n/2$, to však nie je nutné. Generál pošle signál L na obe strany a zistí, ktorý okraj (ľavý alebo pravý) je bližšie. Zistí to podľa toho, odkiaľ príde skôr odpoveď v podobe odrazeného signálu L .

3.1.5 Kruhy

Na rozdiel od úsečky, v kruhu dĺžky n susedí bunka 0 s bunkou $n - 1$. Teda formálne kruh je sieť typu 2, množina vrcholov je $P = \{0, \dots, n-1\}$, $c(i, 0) = (i - 1) \bmod n$, $c(i, 1) = (i + 1) \bmod n$, generál je vo vrchole 0. Využijeme riešenie z časti 3.1.2, aplikované na úsečku $\langle 0, n/2 \rangle$ a zrkadlovo obrátene na úsečku $\langle n/2, n - 1 \rangle$. To dosiahneme tak, že pôvodný generál vyšle na obe strany signály S_1, S_1^r rýchlosťou 1. Tieto signály sa stretnú v bunke číslo $n/2$,

tam sa vytvorí nový generál a oba signály sa odrazia (tj. vzniknú signály S_1'). Teda jediný rozdiel je v tom, že signál S_1 sa neodrazí od okraja úsečky, ale od signálu S_1^r . Preto má toto riešenie čas ako na úsečke dĺžky $n/2$, teda čas je $n + 1$.

V ďalšom ukážeme optimálnosť tohto riešenia (podľa [3]). Nech A je nejaké riešenie pre kruhy. Majme dva kruhy R_1 s n bunkami a R_2 s $2n + 1$ bunkami. Pre jednoduchosť nech n je nepárne. Je zrejmé, že ak $t \leq \lfloor n/2 \rfloor$ a $i \leq \lfloor n/2 \rfloor$, tak $s(i, t, R_1, A) = s(i, t, R_2, A)$. V čase $\lfloor n/2 \rfloor + 1$ sa stavy buniek $\lfloor n/2 \rfloor$ v R_1 a v R_2 môžu líšiť (tj. $s(\lfloor n/2 \rfloor, \lfloor n/2 \rfloor + 1, R_1, A) \neq s(\lfloor n/2 \rfloor, \lfloor n/2 \rfloor + 1, R_2, A)$). Potrvá však ďalších $\lfloor n/2 \rfloor$ krokov, kým sa táto odlišnosť prejaví aj v bunke 0. Presnejšie, pre $t \geq \lfloor n/2 \rfloor + 1$ a $i \leq 2\lfloor n/2 \rfloor - t$ je $s(i, t, R_1, A) = s(i, t, R_2, A)$ (analogicky môžeme argumentovať pre $i' = n - i$), teda aj $s(0, n, R_1, A) = s(0, n, R_2, A)$. Pritom ale kruh R_2 nemôže vystreliť skôr ako v čase $n + 1$ (dovtedy je bunka $n + 1$ v R_2 v stave s_Q). Preto ani R_1 nemôže vystreliť skôr ako v čase $n + 1$.

3.1.6 Jednosmerné kruhy

Formálne jednosmerný kruh je sieť typu 1 s množinou buniek $P = \{p_0, \dots, p_{n-1}\}$, $c(p_i, 0) = p_{i-1}$ pre $1 \leq i \leq n - 1$, $c(p_0, 0) = p_{n-1}$ a $p_G = p_0$. Takže signály sa môžu šíriť len jedným smerom, z bunky p_i do bunky p_{i+1} a z p_{n-1} do p_0 .

Pre jednoduchosť opíšeme riešenie len pre $n = 2^k$, podľa [9].

V riešení použijeme nasledovné série signálov:

1. P s rýchlosťou 0
2. BC s rýchlosťami $\frac{1}{3}, \frac{3}{7}, \dots, (2^i - 1)/(2^{i+1} - 1), \dots$
3. A_0 s rýchlosťou 1
4. RS s rýchlosťami $\frac{2}{3}, \frac{4}{7}, \dots, 2^i/(2^{i+1} - 1), \dots$

i -ty signál série BC resp. RS označíme $BC(i)$ resp. $RS(i)$ pre $i \geq 1$.

Signál P_{00} (patriaci do série P) zodpovedá stavu „generál“. Signál P_{00} hneď po svojom vzniku generuje uvedené 4 série signálov.

Myšlienka riešenia je nasledovná (obr. 3.4). Signál A_0 obíde celý kruh a príde do bunky p_0 v čase n . Potom sa stretáva so signálmi série BC . So signálom $BC(i)$ sa A stretne v čase $((2^{i+1} - 1) / 2^i) n$ v bunke číslo $((2^i - 1) / 2^i) n$. Pri týchto stretnutiach vzniknú noví generáli, ktorí hneď po svojom vzniku generujú signály P , RS , A_0 a BC .

Úlohou série signálov RS je synchronizovať existujúcich a novovzniknutých generálov. Napr. signál $RS(1)$ sa stretne so signálom P v bunke číslo 0 v čase $3n/2$; v tom istom čase sa stretnú A a $BC(1)$.

Nový generál sa vygeneruje po stretnutí signálu A a signálu P , signálu z RS a signálu P a pri stretnutí signálu A a signálu z BC .

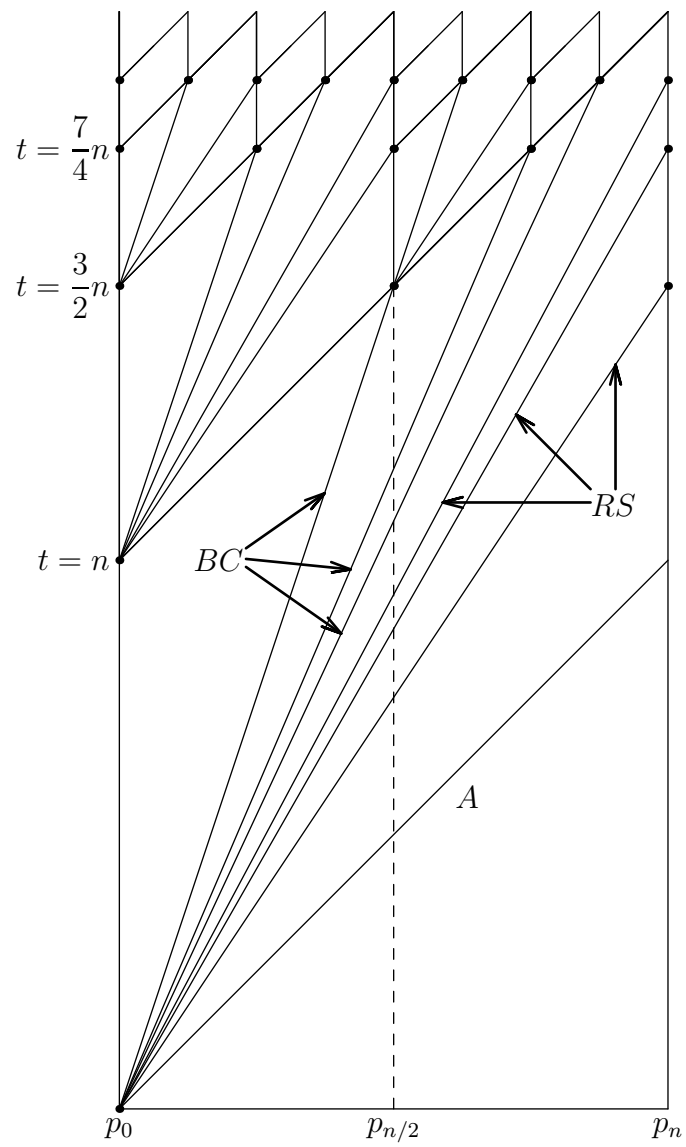
Takto dosiahneme, že v čase $((2^{i+1} - 1) / 2^i) n$ budú generáli vo všetkých bunkách číslo $nk/2^i$ pre $k = 0, 1, \dots, 2^i - 1$. Takže v čase $2n - 2$ bude každá bunka v stave „generál“ a v nasledujúcom takte všetky bunky vystrelia.

Ako implementujeme signály z RS a BC ? Signál A každý druhý takt generuje signál T s rýchlosťou 0. Signál RS ide rýchlosťou $1/2$, kým sa nestretne so signálom T . Pri stretnutí so signálom T signál RS dočasne zrýchli, ako keby vynechal jeden takt. (Tj. za prítomnosti T prejde RS za 1 takt do takého stavu, do akého by bez T prešiel za 2 takty.) Zároveň signál RS pri svojej ceste zruší každý druhý signál T . Takže $RS(1)$ zrýchli v každej druhej bunke, $RS(2)$ v každej štvrtej bunke, $RS(i)$ v každej 2^i -tej bunke. Podobná myšlienka sa dá použiť pri implementácii BC .

Kvázicyklické siete Uvidíme, že riešenie pre jednosmerné kruhy je riešením aj pre kvázicyklické siete (def. 3.1.1).

Definícia 3.1.1. Sieť $N = (P, p_G, c)$ nazveme *kvázicyklická sieť* dĺžky n , ak každá bunka z P má aspoň jedného suseda a existuje taký rozklad P_0, P_1, \dots, P_{n-1} množiny P , že $P_0 = \{p_G\}$ a každá bunka z P_i má všetkých svojich susedov v množine P_{i-1} pre $i = 0, \dots, n - 1$ pričom $P_{-1} = P_{n-1}$.

Kvázicyklická sieť má nasledovné vlastnosti. Množina P_i je množina všetkých buniek, ktoré sú vo vzdialenosti i od generála. V kvázicyklickej sieti dĺžky n musí existovať aspoň jeden cyklus, každý cyklus má dĺžku n a generál



Obrázok 3.4:

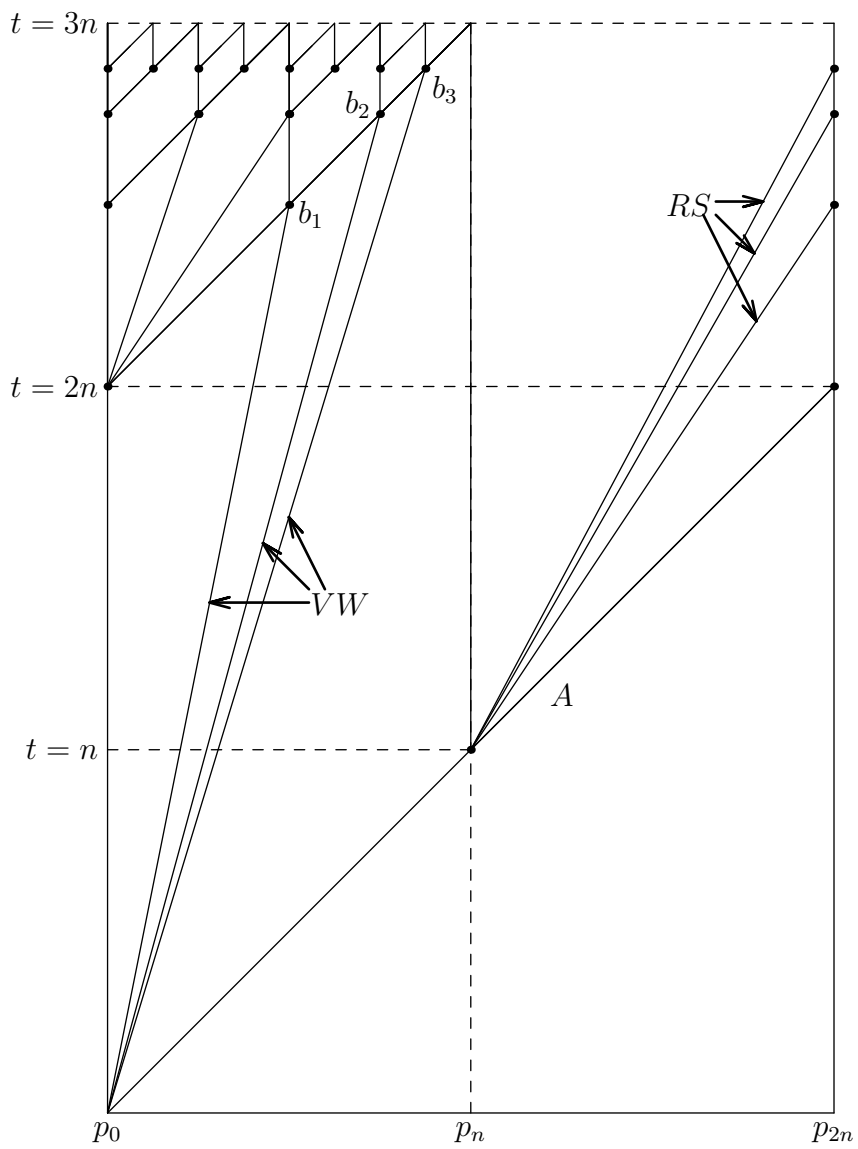
leží na každom cykle. Ak bunka $p \in P_i$, tak každá cesta z p_G do p má dĺžku i . Kruh je špeciálnym prípadom kvázicyklickej siete.

Nech A je riešenie pre jednosmerný kruh. Ako sa bude A správať na kvázicyklickej sieti N_1 dĺžky n ? Nech N_2 je jednosmerný kruh dĺžky n . Všetky bunky z P_i sa správajú rovnako ako i -ta bunka na kruhu dĺžky n , čo sa dá dokázať indukciou vzhľadom na čas. Preto celulárny automat (N_1, A) vystrelí v rovnakom čase ako celulárny automat (N_2, A) .

Synchronizácia polkruhu Synchronizáciu jednosmerného kruhu dĺžky $2n$ vieme urýchliť, ak je na kruhu vyznačená bunka v strede, teda p_n . Teda požadujeme, aby bunka p_n najneskôr pri svojom prebudení vedela, že je v strede. Zrýchlený automat vystrelí v čase $3n - 1$. V tomto prípade sa však nezosynchronizuje celý kruh, ale len polkruh, teda vystrelia len také bunky p , že $d(G, p) \leq n$. Vyslovujeme hypotézu, že existuje automat, ktorý s pomocou vyznačenej bunky v strede zosynchronizuje celý kruh v čase $3n$. Túto hypotézu využívame v kapitole 5.

Označme M_1 riešenie (konečný automat) pre kruh a M_2 riešenie pre polkruh. Chceme dosiahnuť, aby sa M_2 na bunkách p_0, \dots, p_{n-1} správali v čase $\langle 2n, 3n - 1 \rangle$ tak, ako sa správa M_1 na tých istých bunkách v čase $\langle n, 2n - 1 \rangle$, teda o n taktov skôr. Generál p_0 vyšle v čase 0 signál A s rýchlosťou 1. Keď signál A obehne celý kruh a vráti sa do p_0 , tak p_0 vie, že je čas $2n$. Vtedy vyšle signály A, BC, P a RS . Ďalej potrebujeme nahradiť signály RS , ktoré bunka p_0 očakáva v časoch $3n - n/2^i$ pre $i = 1, \dots, k$. Tieto signály vygeneruje bunka p_n v čase n , teda keď tam dorazí signál A . (Práve sme využili, že bunka p_n vie, že je v strede.) Ďalej treba nahradiť signály BC , ktoré vyznačia bunky $p_{n/2}, p_{3n/4}, \dots, p_{n-n/2^i}, \dots$ (body b_1, b_2, b_3 na obr. 3.5). Za týmto účelom vygeneruje p_0 v čase 0 sériu signálov VW s rýchlosťami $\frac{1}{5}, \frac{3}{11}, \dots, (2^i - 1) / (3 \cdot 2^i - 1), \dots$. Napokon pridáme signál Y (na obrázku nie je znázornený), ktorý štartuje v čase 0 z bunky p_0 a má rýchlosť $\frac{1}{3}$. Signál Y potrebujeme na to, aby bunka p_n vedela, kedy má vystreliť – keď do nej príde signál Y , čiže v čase $3n - 1$.

Vráťme sa ešte k vyslovenej hypotéze o synchronizácii celého kruhu v čase $3n - 1$. Pokúsme sa doplniť uvedený postup tak, aby sa bunky p_n, \dots, p_{2n-1}



Obrázok 3.5:

v čase $\langle n, 3n - 1 \rangle$ správali ako bunky na kruhu dĺžky n a teda aby vystrelili tiež v čase $3n - 1$. V prvom rade by musela bunka p_n v čase n vyslať aj signály BC . Ďalej by sme pridali novú sériu signálov Z tak, aby do bunky p_n prišiel signál $Z(i)$ v čase $3n - n/2^i$ – tieto signály by nahrádzali sériu RS . Signály Z by boli generované v bunke p_0 v čase 0. Tieto úvahy naznačujú, že uvedená hypotéza je pravdivá.

3.2 Množina riešení FSSP nie je rekurzívna

Ukážeme (podľa [6]), že množina všetkých riešení FSSP nie je rekurzívne vyčísliteľná. Podobne sa ukáže, že množina riešení v minimálnom čase nie je rekurzívne vyčísliteľná. Najprv si všimnime, že komplement množiny všetkých riešení je rekurzívne vyčísliteľný, Vieme totiž dokázať, že daná prechodová funkcia λ nie je riešením: konkrétny celulárny automat (s daným počtom buniek n) sa musí po konečnom počte krokov ($< |S|^n$) zacykliť. Stačí teda postupne simulovať celulárne automaty s prechodovou funkciou λ a počtom buniek 1, 2, 3, ... Simulácia prebieha dovtedy, kým sa daný celulárny automat buď nezacyklí alebo nezosynchronizuje. Ak existuje n , pre ktoré daná prechodová funkcia nezosynchronizuje automat, určite ho nájdeme. Pre prípad riešenia v minimálnom čase je to jednoduchšie, stačí simulovať automat do času $2n - 2$.

Dokazujeme sporom. Predpokladajme, že aj množina všetkých riešení FSSP je rekurzívne vyčísliteľná. Keďže komplement je rekurzívne vyčísliteľný, pôvodná množina je rekurzívna. Pomocou tohto predpokladu vyriešime problém zastavenia Turingovho stroja. Skonstruujeme prechodové funkcie celulárneho automatu $\lambda_1, \lambda_2, \dots$ také, že λ_i je riešením FSSP práve vtedy, keď sa i -ty Turingov stroj nezastaví na i -tom vstupe. Potom by sme mali algoritmus pre problém zastavenia, čo je spor.

Automat s prechodovou funkciou λ_i bude naraz vykonávať dve činnosti: synchronizuje bunky a zároveň simuluje i -ty Turingov stroj na i -tom vstupe. Nech CA má n buniek. Konštruovaný Turingov stroj bude spĺňať nasledovné podmienky:

- (1) Ak simulácia Turingovho stroja potrebuje viac ako n políčok pásky,

tak sa simulácia zastaví a synchronizácia sa dokončí.

(2) Ak sa Turingov stroj zastaví (pred dosiahnutím synchronizácie), tak sa automat nezosynchronizuje.

Z týchto dvoch podmienok vyplýva, že ak sa i -ty Turingov stroj zastaví na i -tom vstupe, tak pre dostatočne veľké n (aby nenastala podmienka (1)) sa celulárny automat nezosynchronizuje (kvôli podmienke (2)). Ak sa Turingov stroj nezastaví, celulárny automat sa zosynchronizuje pre každé n .

Ostáva opísať simuláciu Turingovho stroja T pomocou celulárneho automatu A . Nech Q_T je množina stavov T , A_T je pásková abeceda T . Potom množina stavov A bude $Q_A = (Q_T \cup \{q_n\}) \times A_T$, kde $q_n \notin Q_T$, teda i -te políčko A si pamätá, či je na zodpovedajúcom políčku T hlava (q_n znamená, že tam hlava nie je), ak tam hlava je, v akom je stave (prvý člen dvojice) a aký je symbol na príslušnom políčku pásky stroja T (druhý člen dvojice).

Simulovaný Turingov stroj na začiatku má na vstupe prázdnu pásku, ale cieľom je, aby mal na vstupe i -ty reťazec. Najjednoduchšie bude, ak celulárny automat bude simulovať dva Turingove stroje za sebou. Najprv odsimulujeme TS, ktorý zapíše na pásku i -ty vstup a vráti hlavu na začiatok, teda pripraví vstup. (Toto je možné napevno zakódovať do prechodovej funkcie.) Potom simulujeme i -ty TS.

Podmienku (2) naplníme tak, že ak sa Turingov stroj zastaví, synchronizáciu pokazíme (na tom mieste, kde je hlava) tým, že príslušná bunka prejde do nejakého špeciálneho stavu $\neq q_f$ a zotrvá v ňom.

3.3 Dvojrozmerné CA

Celulárny automat (CA) nazveme dvojrozmerný (2D), ak sú jeho bunky pospájané do dvojrozmernej mriežky. Pritom v niektorých miestach môže bunka chýbať, v takom prípade susedia chýbajúcej bunky vidia na vstupe stav s_E na mieste chýbajúcej bunky. Súvislú konečnú podmnožinu množiny \mathbb{Z}^2 budeme nazývať *obrazec* (figure). Formálne môžeme 2D CA chápať ako CA, ktorého sieť je $\text{Grid}(F)$ pre obrazec F . Budeme predpokladať, že generál je na pozícii $(0, 0)$, teda $s((0, 0), 0, F, A) = s_E$.

3.3.1 Veta o dolnom časovom odhade

Nasledujúca veta je prevzatá z [4].

Veta 3.3.1. *Nech M je množina obrazcov, nech $F \in M$. Ďalej nech P, Q, R sú po dvoch disjunktné podmnožiny \mathbb{Z}^2 také, že*

- (1) Q je neprázdna
- (2) $F = P \cup Q$
- (3) žiadna bunka z P nesusedí so žiadnou bunkou z R (teda Q oddeľuje P od R)
- (4) pre každé $p, p' \in F$ platí: spomedzi najkratších ciest z p do p' v $F \cup R$ existuje jedna cesta, ktorá je v F (inak povedané, prídanie R k F neskráti cesty v F)
- (5) $F \cup R$ je obrazec v M

Nech $a \in F$, $B \in F \cup R$ sú také bunky, že ak L_0 je dĺžka najkratšej cesty v F z $(0,0)$ do a idúcej cez nejakú bunku v Q a L_1 je vzdialenosť v $F \cup R$ z $(0,0)$ do b , tak $L_1 \leq L_0$.

Potom $t(F, A) \geq L_1$.

Dôkaz. Nasledujúce dve tvrdenia sú zrejmé:

(6) Ak $p \in P$ a $s(p, t, F, A) \neq s(p, t, F \cup R, A)$, tak pre nejakú bunku p' susediacu s p a nejaký čas $t' < t$ platí $s(p', t', F, A) \neq s(p', t', F \cup R, A)$. Pritom $p' \in F$ kvôli podmienke (3).

(7) Ak $p \in F$ (resp. $F \cup R$) a $s(p, t, F, A) \neq q_q$ (resp. $s(p, t, F \cup R, A) \neq q_q$), tak buď $p = (0,0)$ alebo pre nejakú bunku $p' \in F$ (resp. $p' \in F \cup R$) susediacu s p a nejaký čas $t' < t$ platí $s(p', t', F, A) \neq s_Q$ (resp. $s(p', t', F \cup R, A) \neq s_Q$).

Dokazujeme sporom. Predpokladajme, že $t_0 = t(F, A) < L_1$. Z definície t_0 máme $s(a, t_0, F, A) = s_F$. Pritom ale dĺžka ľubovoľnej cesty z $(0,0)$ do b v $F \cup R$ je viac ako t_0 , lebo L_1 je dĺžka najkratšej takej cesty a $t_0 < L_1$. Preto b nemôže vystreliť už v čase t_0 , teda $s(b, t_0, F \cup R, A) \neq s_F$. Preto ani a nemôže vystreliť už v čase t_0 v obrazci $F \cup R$, teda $s(a, t_0, F \cup R, A) \neq s_F$. (Keby bunka a vystrelila a bunka b nie, A by nebol riešením problému strelcov pre obrazec $F \cup R$, ale $F \cup R \in M$). Z (6) vyplýva, že existuje cesta $p_0 = a$,

p_1, \dots, p_u ($\in Q$) a časy $t_0 > t_1 > \dots > t_u$ také, že $p_i \in P$ pre každé i , $0 \leq i < u$ a $s(p_i, t_i, F, A) \neq s(p_i, t_i, F \cup R, A)$ pre každé i , $0 \leq i \leq u$. Preto aspoň jeden zo stavov $s(p_i, t_i, F, A)$, $s(p_i, t_i, F \cup R, A)$ nie je q_q . Podľa (2) preto existuje cesta $p_u, p_{u+1}, \dots, p_{u+v} = (0, 0)$ v F (resp. $F \cup R$) a časy $t_u > t_{u+1} > \dots > t_{u+v} \geq 0$ také, že $s(p_{u+j}, t_{u+j}, F, A) \neq s_Q$ (resp. $s(p_{u+j}, t_{u+j}, F \cup R, A) \neq q_q$), kde $0 \leq j \leq v$. Postupnosť $p_{u+v}, p_{u+v-1}, \dots, p_u, \dots, p_0$ je cesta dĺžky $u + v$ z $(0, 0)$ do a cez bunku z Q (konkrétne p_u) v obrazci $F \cup R$. Preto podľa (4) existuje cesta z $(0, 0)$ do a cez bunku z Q v obrazci F , pričom dĺžka tejto cesty je najviac $u + v$. Preto $u + v \geq L_0$ podľa definície L_0 . Na druhej strane $t_0 > \dots > t_u > \dots > t_{u+v} \geq 0$ implikuje $t_0 \geq u + v$. Takže dostávame spor: $L_0 \geq L_1 > t_0 \geq u + v \geq L_0$. \square

Dokázanú vetu použijeme na konkrétnych prípadoch:

Úsečka Nech generál je vzdialený od ľavého kraja o $k < n/2$ buniek. Máme teda množinu obrazcov $M = \{(x, 0) \mid x_0 \leq x \leq x_1 \mid x_0, x_1 \in \mathbb{Z}\}$, obrazec $F \in M$. Položme $P = \{(x, 0) \mid -k \leq x \leq n - k - 2\}$, $Q = \{(n - k - 1, 0)\}$, $R = \{(x, 0) \mid n - k \leq x \leq 2n - k - 2\}$, $a = (-k, 0)$, $b = (2n - k - 2, 0)$. Potom $L_0 = L_1 = 2n - k - 2$, preto podľa dokázanej vety $t(F, A) \geq 2n - k - 2$.

Obdĺžnik

Uvažujeme množinu obrazcov tvaru $F_{m,n} = \{(x, y) \mid 0 \leq x \leq m - 1 \wedge 0 \leq y \leq n - 1\}$, kde $m \geq 1, n \geq 1$ a aspoň jedno z m, n je viac ako 1. Predpokladajme $m \geq n$. Položíme $P = \{(x, y) \mid (0 \leq x \leq m - 2 \wedge 0 \leq y \leq n - 1)\}$, $Q = \{(m - 1, y) \mid 0 \leq y \leq n - 1\}$, $R = \{(x, y) \mid (m \leq x \leq 2m - 2 \wedge 0 \leq y \leq n - 1)\}$, $a = (0, n - 1)$, $b = (2m - 2, n - 1)$. Potom $t(F, A) \geq L_0 = L_1 = 2m + n - 3$. Keby bolo $m < n$, dostali by sme $t(F, A) \geq m + 2n - 3$. Tieto dva odhady môžeme zhrnúť do jedného: $t(F, A) \geq m + n + \max\{m, n\} - 3$

3.3.2 Štvorec

Časti 3.3.2, 3.3.3 a 3.3.4 sú prevzaté z [10].

Štvorec je formálne $F = \{(x, y) \mid 0 \leq x \leq n - 1 \wedge 0 \leq y \leq n - 1\}$, teda generál je v ľavom dolnom rohu štvorca. Synchronizácia prebehne tak, že každá bunka (i, i) , kde $0 \leq i \leq n - 2$, prejde do stavu s v čase $t_i = 2i$. Bunka zistí, že má pozíciu tvaru (i, i) podľa toho, že ľavý aj dolný sused sa prebudili v tom istom kole.

Úlohou bunky (i, i) je zosynchronizovať úsek napravo od nej – $\{(x, i) \mid i \leq x \leq n - 1\}$ a úsek nahor od nej – $\{(i, y) \mid i \leq y \leq n - 1\}$, čo je už vyriešený jednorozmerný prípad. Pre i -ty riadok aj i -ty stĺpec synchronizácia trvá $2(n - i) - 2$, pričom sa začala v čase $t_i = 2i$, teda sa dokončí v čase $2(n - i) - 2 + 2i = 2n - 2$. Špeciálne treba ošetriť bunku $(n - 1, n - 1)$, tá vystrelí jedno kolo po tom, čo sa obaja jej susedia prebudili (vynecháme prechod do stavu s_G).

Optimalita uvedeného riešenia vyplýva z toho, že $2n - 2$ je dĺžka najkratšej cesty medzi bunkami $(0, 0)$ a $(n - 1, n - 1)$.

3.3.3 Obdĺžnik

Formálne $F = \{(x, y) \mid 0 \leq x \leq m - 1 \wedge 0 \leq y \leq n - 1\}$. Budeme využívať riešenie jednorozmerného FSSP s generálom na ľubovoľnom mieste, ako sme ho uviedli v časti 3.1.4. Každá bunka (i, i) sa stane generálom v čase $t_i = 3i$, kde $0 \leq i \leq \min\{m, n\}$. Zjednotenie úseku napravo od generála a nahor od generála, teda množinu $F_i = \{(x, i) \mid i \leq x \leq m - 1\} \cup \{(i, y) \mid i \leq y \leq n - 1\}$ budeme chápať ako jednu úsečku. Úlohou generála je zosynchronizovať množinu F_i . Dĺžka úsečky je $n_i = (n - i) + (m - i) - 1$, vzdialenosť generála od najbližšieho okraja je $k_i = \min\{n - i - 1, m - i - 1\}$. Takže bunky z F_i vystrelia v čase $t = t_i + 2n_i - k_i - 2 = 3i + 2(n - i + m - i - 1) - \min\{n - i - 1, m - i - 1\} - 2 = n + m + \max\{m, n\} - 3$. To je zhodné s dolným odhadom, dokázaným vo vete 3.3.1, takže toto riešenie je časovo optimálne.

3.3.4 Kocka

Formálne $F = \{(x, y, z) \mid 0 \leq x \leq n - 1 \wedge 0 \leq y \leq n - 1 \wedge 0 \leq z \leq n - 1\}$. Riešenie bude fungovať tak, že každá bunka (i, i, i) sa v čase $t_i = 3i$ stane generálom, pričom $0 \leq i \leq n - 2$. Generál na pozícii (i_0, i_0, i_0) potom

zosynchronizuje steny (i_0, j, k) , (j, k, i_0) , (j, i_0, k) , kde $i_0 \leq j, k \leq n - 1$. Príslušné tri steny synchronizujeme ako obdĺžniky s $m_i = n_i = n - i$, na ich synchronizáciu použijeme postup pre obdĺžnik uvedený v časti 3.3.3. (Na synchronizáciu stien nemôžeme použiť riešenie pre štvorec, lebo toto riešenie je pre naše účely príliš rýchle. Napr. steny obsahujúce $(0, 0, 0)$ by vystrelili v čase $2n - 2$, avšak bunka $(n - 1, n - 1, n - 1)$ môže vystreliť až v čase $3n - 3$.) Bunky vystrelia v čase $t = t_i + 2m_i + n_i - 3 = 3n - 3$. Bunka $(n - 1, n - 1, n - 1)$ vystrelí vtedy, keď sa všetci jej (traja) susedia prebudia. Optimalita riešenia, podobne ako pri štvorci, vyplýva z toho, že $3n - 3$ je dĺžka najkratšej cesty medzi bunkami $(0, 0, 0)$ a $(n - 1, n - 1, n - 1)$.

3.3.5 Ďalšie obrazce

Budeme sa zaoberať triedou obrazcov $M_{\text{step}} = \{(x, y) \mid 0 \leq x \leq x_0 - 1 \wedge f(x) \leq y \leq g(x)\} \mid x_0 \geq 1, f \text{ a } g \text{ sú neklesajúce funkcie také, že } f(x) \leq g(x) \text{ pre každé } x, 0 \leq x \leq x_0\}$ (podľa [4]).

Problém pre M_{step} zredukujeme na jednorozmerný problém. Majme $F \in M$. Nech $(x_0 - 1, y_0 - 1)$ je pravý horný roh F , tj. $(x_0 - 1, y_0 - 1) \in F \wedge (\forall (x, y) \in F)(x \leq x_0 \wedge y \leq y_0)$. Budeme simulovať riešenie na úsečke F' dĺžky $x_0 + y_0 - 2$. Pritom i -tej bunke z F' zodpovedajú všetky také bunky $(x, y) \in F$, že $x + y = i$. Teda všetky bunky simulujúce i -tu bunku budú v danom čase t v tom istom stave. Riešenie bude mať tú vlastnosť, že v čase t bude postupnosť stavov na ľubovoľnej najkratšej ceste v F z $(0, 0)$ do $(x_0 - 1, y_0 - 1)$ rovnaká ako postupnosť stavov v čase t na F' , a to pre každé t .

Zabezpečíme to nasledovne. Majme bunku p . Ak nemá ani ľavého, ani dolného suseda, simuluje najľavejšiu bunku, správa sa ako generál. Ak má ľavého alebo dolného suseda, pri simulácii F' ho považuje za svojho ľavého suseda. (Ak má oboch susedov, ľavého aj dolného, ich stavy sa musia rovnať). Analogicky pre pravého suseda. Ak bunka nemá ani pravého, ani horného suseda, simuluje najpravejšiu bunku z F' .

Toto riešenie trvá tak dlho, ako riešenie jednorozmerného FSSP na F' , čo je $2(x_0 + y_0 - 1) - 2 = 2x_0 + 2y_0 - 4$ krokov.

Dolný odhad dokážeme pomocou vety 3.3.1. Za R položíme obrazec F posunutý o vektor $(x_0 - 1, y_0 - 1)$ bez bunky $(x_0 - 1, y_0 - 1)$, teda $R = \{(x + x_0 - 1, y + y_0 - 1) \mid (x, y) \in F\} \setminus \{(x_0 - 1, y_0 - 1)\}$. Ďalej $Q = \{(x_0 - 1, y_0 - 1)\}$, $P = F \setminus Q$, $a = (0, 0)$, $b = (2x_0 - 2, 2y_0 - 2)$. Dostávame $t(F, A) \geq L_1 = L_0 = 2x_0 + 2y_0 - 4$, teda uvedené riešenie je pre túto triedu optimálne.

3.3.6 Všeobecné obrazce

V tejto časti budeme predpokladať, že mriežka neobsahuje diery a že je potenciálne nekonečná. Každá bunka je na začiatku v jednom zo stavov s_0, s_1 . Obrazec F je tvorený bunkami, ktoré sú v čase 0 v stave s_1 . Úlohou je zosynchronizovať obrazec F , pričom môžeme využívať aj ostatné bunky, ale tie musíme nakoniec vrátiť do stavu s_0 . Na rozdiel od predchádzajúcich častí budeme za susedov bunky považovať 8 okolitých buniek: susedia (x_0, y_0) sú $\{(x, y) \mid |x - x_0| = 1 \vee |y - y_0| = 1\}$.

Riešenie ([8]) bude prebiehať v dvoch fázach. V prvej fáze ohraničíme F , teda nájdeme štvorec so stredom $(0, 0)$, v ktorého vnútri sú všetky bunky z F . To znamená, že bunky na hranici štvorca prejdú do špeciálneho stavu. V druhej fáze stačí zosynchronizovať každý riadok, čo je jednorozmerný FSSP, ktorý sme už vyriešili. Bude to fungovať, lebo všetky hraničné bunky prejdú do špeciálneho stavu naraz.

Označme $J_k = \{(x, y) \mid |x| \leq k \wedge |y| \leq k\}$. Pod rozširujúcou sa vlnou (expanding wave) budeme rozumieť signál, ktorý ak štartuje v bunke na pozícii x v čase t , dosiahne všetky bunky na pozíciách $y = x + z$ pre nejaké $z \in J_k \setminus J_{k-1}$ v čase $t' = t + w(k)$, kde w je rastúca funkcia. (Sčítovanie dvojíc chápeme po zložkách.) Zmršťujúca sa vlna (contracting wave) bude signál, ktorý štartuje v čase t zo všetkých buniek $y = x + z$ pre nejaké pevné x a pre všetky $z \in J_r \setminus J_{r-1}$ a dosiahne všetky bunky y' také, že $y' = x + z'$ pre každé $z' \in J_{r-k} \setminus J_{r-k-1}$ v čase $t' = t + w(k)$, kde w je rastúca funkcia. Ako w budeme používať lineárne funkcie. Ak $w(k) = ck$ pre $c \in \mathbb{N}$, povieme, že príslušná vlna má rýchlosť c . Najväčšiu rýchlosť má vlna s $c = 1$.

Ďalej popíšeme, ako ohraničiť F . Generál vyšle v čase 0 rozširujúcu sa vlnu A s rýchlosťou c_1 . Keď vlna A zasiahne bunku p , p si zapamätá, odkiaľ

vlna prišla. Takto bude p schopná poslať signál generálovi. Ak $p \in F$, tak p pošle generálovi potvrdenie – signál B rýchlosťou c_2 , pričom B je rýchlejší ako A , teda $c_2 < c_1$.

Generál očakáva, že bude pravidelne každé $(c_1 + c_2)$ -te kolo dostávať signály B . Konkrétne, majme bunku p . Nech $m \in \mathbb{N}$ je najmenšie také, že $p \in J_m$. Trvá mc_1 jednotiek času, kým A zasiahne p a mc_2 jednotiek času, kým sa odpoveď B dostane späť ku generálovi. Preto generál očakáva signál B v časoch $i(c_1 + c_2)$ pre $1 \leq i \leq n$, kde $n \in \mathbb{N}$ je najmenšie také, že $F \subseteq J_n$. Keď generál očakávaný signál B nedostane (tj. v čase $(n + 1)(c_1 + c_2)$), vie, že vlna A už zasiahla všetky bunky z F . Preto počká k_1 krokov (kde k_1 je konštanta) a vyšle rozširujúcu sa vlnu C rýchlosťou c_3 , pričom $c_3 < c_1$ (C je rýchlejšia ako A). Cieľom je, aby C dobehla A . O ďalších k_2 krokov (kde k_2 je konštanta) vyšle generál rozširujúcu sa vlnu D s rýchlosťou c_4 , pomalšiu ako C .

Keď C dobehne A , obe sa zastavia. Bunky, zasiahnuté oboma vlnami, vyšlú zmršťujúcu sa vlnu E s rýchlosťou c_5 . (Bunky vedia určiť, ktorým smerom poslať vlnu E , lebo to je smer ku generálovi. Ten si zapamätali, keď dostali A .) Potom tieto bunky prejdú do stavu s_0 . Ostatné bunky, ktoré dostanú signál E , tiež prejdú do stavu s_0 , lebo E zasiahne iba bunky mimo obrazca F .

To zabezpečí vlna D . Tá sa stretne s E a obe sa zastavia. Pri vhodnej voľbe $c_1, \dots, c_5, k_1, k_2$ je sa dá dosiahnuť, aby sa D a E stretli na obvode najmenšieho štvorca so stredom $(0, 0)$, ktorý obsahuje obrazec F . (Tento obvod štvorca je množina $J_n \setminus J_{n-1}$). Jedna vhodná kombinácia je nasledovná:

A	B	C	D	E		
c_1	c_2	c_3	c_4	c_5	k_1	k_2
3	1	1	3	1	2	4

3.4 Synchronizácia všeobecného stromu

V tejto časti sa budeme zaoberať synchronizáciou všeobecných stromov. Vychádzame z článku [9], ale uvedieme len zjednodušené riešenie pre stromy.

Slovo „všeobecný“ zdôrazňujeme preto, lebo pripúšťame všetky stromy (s konštantným maximálnym stupňom vrchola) s ľubovoľnou pozíciou generála, na rozdiel od kapitoly 4. Rýchlosť riešenia nebudeme vyjadrovať podľa počtu buniek, ale podľa polomeru stromu (def 3.4.1).

Definícia 3.4.1. Polomer siete $N = (P, p_G, c)$ je $r = \max\{d(p, p_G) \mid p \in P\}$.

Takže ak r je polomer siete N , tak všetky bunky v sieti N sa prebudia najskôr v čase r . Každú bunku X , ktorá je najvzdialenejšia od G , tj. $d(G, X) = r$, budeme volať *radiálna bunka* a cestu GX budeme volať *radiálna cesta*.

Najskôr uvedieme jednoduchšie riešenie, ktoré potrebuje čas $4r - 1$. Potom popíšeme vylepšenie, ktorým znížime potrebný čas na $3r$.

Riešenie v čase $4r - 1$. Riešenie bude používať simuláciu kvázicyklickej siete. Pre každú radiálnu bunku X budeme na ceste GX simulovať jednosmerný kruh. Vo virtuálnej sieti nebudú žiadne iné cykly. Označme bunky jednosmerného kruhu po rade $p'_0, p'_1, \dots, p'_{n'-1}$, kde $n' = 2r$ bude počet buniek simulovaného kruhu. Ak bunka p leží na ceste GX , $d(G, p) = i$ a $0 < i < r$, tak p bude simulovať bunky p_i a p_{2r-i} . Takéto bunky budeme volať bunky 1. skupiny.

Bunka G simuluje iba bunku p'_0 a radiálne bunky simulujú iba bunku p'_r .

Všimnime si teraz ostatné bunky, teda také bunky p , ktoré neležia na žiadnej ceste dĺžky r začínajúcej v G . Takéto bunky budeme volať bunky 2. skupiny. Tieto bunky budú simulovať stromy visiace z virtuálneho kruhu. Ak p je bunka 2. skupiny, tak p bude simulovať bunku $p'_{d(G,p)}$. Okrem toho bude simulovať bunku p'_i , kde $i > d(G, p)$, to však nie je podstatné pre funkčnosť riešenia.

Teraz opíšeme simuláciu. Každá bunka okrem G a listov bude simulovať 2 virtuálne bunky. Treba si uvedomiť, že bunka p_i je v čase skoršom ako i v pokojnom stave. Nech p je bunka, p nie je list a $d(G, p) = i$. Bunka p začne simulovať 1. virtuálnu bunku p'_i zároveň so svojim prebudením, teda v čase i ; to je čas, kedy sa má prebudiť aj p'_i . Druhú virtuálnu bunku začne p simulovať až 1 takt po tom, keď každý syn p začal simulovať 2. virtuálnu

bunku alebo nastavil príznak listu. Tým je zaručené, že ak p je na radiálnej ceste, tak p začne simulovať bunku p'_{2r-i} v čase $2r - i$. Ak p je list, tak p si nastaví príznak listu pri svojom prebudení a simuluje jedinú bunku.

Reálna bunka p vystrelí, keď jej prvá simulovaná bunka vystrelí.

Simulovaná kvázicyklická sieť má dĺžku $2r$, takže vystrelí v čase $4r - 1$.

Riešenie v čase $3r$. Predchádzajúce riešenie sa dá vylepšiť využitím faktu, že kruh dĺžky n vieme zosynchronizovať v čase $\frac{3}{2}n$, ak je na kruhu vyznačená bunka p_M v strede, tj. $d(G, p_M) = n/2$. Bunku p_M stačí vyznačiť pri jej prebudení v čase $n/2$, lebo dovtedy je aj tak v pokojnom stave.

V našom prípade treba vyznačiť bunku p'_r . Pritom bunka p'_r je simulovaná práve radiálnymi bunkami, teda by sme mali vyznačiť práve radiálne bunky. Avšak nájdenie radiálnych buniek by trvalo príliš dlho, preto vyznačíme nadmnožinu radiálnych buniek – všetky listy okrem generála. Dôležité je, že z vyznačených buniek iba radiálne bunky ležia na virtuálnom cykle, teda iba radiálne bunky môžu ovplyvniť stav generála. Teda ak p je list a $0 < d(G, p) < r$, tak virtuálna bunka p' simulovaná bunkou p bude tiež vyznačená; to nevádi, lebo p' neleží na žiadnom virtuálnom cykle, teda neexistuje virtuálna cesta z p' do generála p'_0 .

Keďže niektoré vyznačené bunky nie sú radiálne, musíme synchronizáciu reálnej siete oneskoriť o 1 takt oproti virtuálnej sieti. (Inak by vyznačené bunky, ktoré nie sú radiálne, vystrelili príliš skoro.) Vyznačená bunka p vystrelí jeden takt potom, ako prvá simulovaná bunka aspoň jedného suseda bunky p vystrelila.

Všimnime si, že výstrel na reálnej sieti závisí len od prvých simulovaných buniek. Preto nám v tomto prípade stačí, aby vo virtuálnej sieti vystrelila len tá polovica buniek, ktorá je bližšie ku generálovi.

Dolný odhad. Uvažujme sieť N tvaru úsečky s bunkami p_0, \dots, p_{2r} a s generálom v strede, tj. v bunke p_r . Táto sieť má $n = 2r + 1$ buniek a polomer r . Použitím dolného odhadu pre úsečku (viď dôsledky vety 3.3.1) dostaneme $t_F(N, A) \geq 2n - r - 2 = 3r$.

Všimnime si, že N je aj strom. Tým sme ukázali, každé riešenie FSSP

pre stromy potrebuje v najhoršom prípade čas $3r$. (Najhorším prípadom je práve N .)

To však nevyklučuje možnosť, že pre užšiu triedu sietí existuje riešenie, ktoré je aj v najhoršom prípade rýchlejšie ako $3r$. Tejto otázke sa venujeme v kapitole 4.

Kapitola 4

Synchronizácia stromov

Ako sme ukázali v časti 3.4, všeobecný strom sa nedá v najhoršom prípade synchronizovať rýchlejšie ako v čase $3r$. Pritom úsečka s generálom na kraji, čo môžeme považovať za špeciálny prípad stromu, sa dá synchronizovať v čase $2r$. (V tomto prípade r sa rovná počtu buniek mínus 1.) Vzniká preto otázka, aká veľká je trieda tých sietí, ktoré sa dajú synchronizovať rýchlejšie ako $3r$. V tejto kapitole ukážeme, že do uvedenej triedy patria stromy so špeciálnou pozíciou generála.

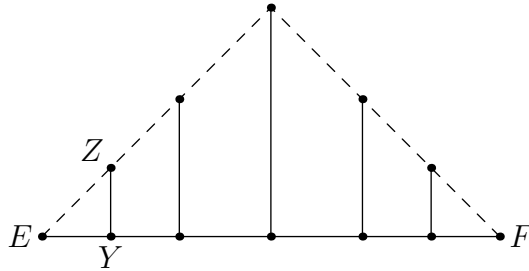
4.1 Definície

Definícia 4.1.1. Cestu EF nazveme *hlavnou vetvou* stromu T , ak $d(E, F) = \text{diam}(T)$.

Ak bod X leží na hlavnej vetve EF , budeme hovoriť, že EF je X -*hlavná vetva*.

Definícia 4.1.2. Nech EF je hlavná vetva, nech Y leží na EF . Nech Z je taký list, že cesty EF a YZ majú spoločný len bod Y . Potom cestu YZ budeme nazývať *vedľajšia vetva* vzhľadom k EF . (obr. 4.1)

Strom môže obsahovať aj viac hlavných vetiev. Preto nestačí povedať, že YZ je vedľajšia vetva. Treba rozlišovať, vzhľadom ku ktorej hlavnej vetve je YZ vedľajšia.



Obrázok 4.1:

V nasledujúcej vete ukážeme, ako definícia 4.1.1 obmedzuje dĺžku vedľajších vetiev. Toto obmedzenie je v obr. 4.1 znázornené čiarkovanými čiarami. Poznamenajme, že obr. 4.1 znázorňuje len dĺžky vedľajších vetiev – vrcholy vedľajších vetiev môžu mať stupeň väčší ako 2.

Veta 4.1.3. *Ak EF je hlavná vetva, Y leží na EF a YZ je vedľajšia vetva vzhľadom na EF , tak $d(Y, Z) \leq \min\{d(E, Y), d(F, Y)\}$.*

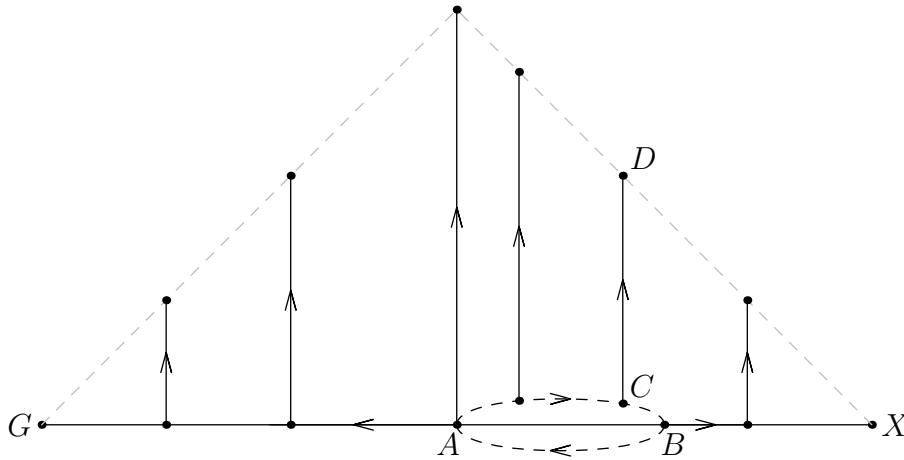
Dôkaz. Bez ujmy na všeobecnosti nech $d(E, Y) \leq d(F, Y)$. Potom dokazujeme, že $d(Y, Z) \leq d(E, Y)$. Sporom. Nech $d(Y, Z) > d(E, Y)$. Potom aj $d(Y, Z) + d(F, Y) > d(E, Y) + d(F, Y)$, teda $d(F, Z) > d(E, F)$. Takže E nie je najvzdialenejší bod od F , čo je spor s tým, že EF je hlavná vetva. \square

V tejto kapitole sa budeme zaoberať synchronizáciou stromov, pričom generál bude vždy na hlavnej vetve.

4.2 Generál na kraji hlavnej vetvy, čas $(2 + \frac{1}{2})r$

V tejto časti popíšeme riešenie pre prípad, že generál je na kraji hlavnej vetvy, teda v liste.

Synchronizácia bude prebiehať v 2 fázach. V prvej fáze nájdeme na hlavnej vetve bunky A, B tak, aby $d(G, A) = \frac{1}{2}r$, $d(G, B) = \frac{3}{4}r$ (obr. 4.2). Po skončení prvej fázy sa začne druhá fáza. V nej simulujeme (v zmysle def. 2.3.1) synchronizáciu virtuálneho kruhu ABA s generálom v bunke A . Presnejšie, na reálnej sieti simulujeme kvázicyklickú sieť (viď časť 3.1.6), pričom



Obrázok 4.2:

každá reálna bunka na úsečke AB simuluje 2 virtuálne bunky patriace kruhu ABA ; ostatné reálne bunky simulujú orientované stromy visiace z kruhu ABA . Virtuálna sieť sa od reálnej siete odlišuje len tým, že spojenia medzi bunkami sú jednosmerné v smere od bodu A a že sme pridali orientovanú cestu BA . Dĺžka virtuálneho kruhu ABA je $r' = \lfloor \frac{r+3}{2} \rfloor$.

Aby bolo možné použiť riešenie pre kvázicyklickú sieť, vo virtuálnej sieti musia byť všetky bunky vzdialené od generála menej ako r' . Aby táto podmienka platila, virtuálna cesta zodpovedajúca CD bude začínať v hornom polkruhu AB a nie v dolnom polkruhu BA (viď obr. 4.2) – cieľom je, aby bunky cesty CD boli (vo virtuálnej sieti) čo najbližšie k A . Viď lema 4.2.5.

Prvá fáza prebieha nasledovne. Bunka G vyšle v čase 0 signály S_0 s rýchlosťou 1, S_A s rýchlosťou $1/3$ a S_B s rýchlosťou $3/5$. Úlohou signálu S_0 je odraziť sa od konca hlavnej vetvy, potom sa stretnúť so signálom S_A (resp. S_B) a na mieste stretnutia vyznačiť bod A (resp. B). Signál vzniknutý odrazom S_0 budeme značiť S'_0

Signál S_0 sa šíri broadcastom. Navyiac, každá bunka si zároveň s prevzatím signálu S_0 zapamätá, od ktorého suseda signál S_0 prišiel – tento sused je rodičom danej bunky. Keď S_0 príde do listu, vygeneruje sa tu signál S'_0 .

Signál S'_0 sa šíri opačným smerom ako S_0 , teda smerom ku generálovi. Bunka prevezme signál S'_0 až vtedy, keď tento signál dorazí do všetkých synov bunky. (Toto platí aj pre bunky stupňa 2 – teda tie, ktoré majú jediného

syna. V tomto prípade sa signál S'_0 pohybuje bez zdržania.) Takže signál S'_0 čaká v bunke p dovtedy, kým rodič bunky p tento signál neprevezme.

Súčasne so šírením signálu S'_0 prebieha vyznačovanie G -hlavných vetiev. Majme bunky p, p' , kde p je rodič bunky p' . Nech v čase t dorazí do bunky p' signál S'_0 . Ak bunka p prevezme signál S'_0 hneď (tj. v čase $t + 1$), tak bunka p' si túto skutočnosť zapamätá – nastaví u seba príznak hlavnej vetvy. Význam tohoto príznaku objasní lema 4.2.1.

Signál S_B sa šíri broadcastom, podobne ako S_0 . (Netreba ošetrovať prípad, že S_B dorazí do listu, lebo taký prípad nemôže nastať. Ešte predtým sa totiž S_B stretne s S'_0 a S_B zanikne.)

Signál S_A sa bude šíriť len po hlavnej vetve (lema 4.2.4). Nech signál S_A dorazil do bunky p . Signál S_A si preberie len taký sused bunky p , v ktorom nečaká signál S'_0 . Ako dokážeme v leme 4.2.2, taký sused bude najviac jeden.

Lema 4.2.1. *Ak bunka p' má nastavený príznak hlavnej vetvy a jej rodič p je na G -hlavnej vetve, tak aj p' je na G -hlavnej vetve.*

Dôkaz. V dôkaze nebudeme využívať, že G je na konci hlavnej vetvy, aby sme túto lemu mohli využiť aj v časti 4.3.

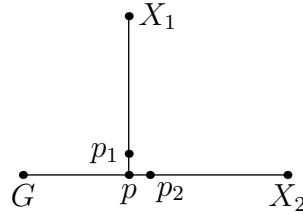
Keďže p' má nastavený príznak hlavnej vetvy, tak p' bol posledný spomedzi synov bunky p , ktorý dostal signál S'_0 . Preto najvzdialenejším potomkom bunky p je nejaká bunka zo subtree (p'); označme túto bunku X . Kvôli sporu predpokladajme, že existuje bunka Y taká, že $d(G, Y) > d(G, X)$. Potom Y nemôže byť potomkom p , takže každá cesta začínajúca úsekom Gp má dĺžku menšiu ako $d(G, Y)$. Teda žiadna z týchto ciest nemôže byť G -hlavnou vetvou. Preto p neleží na G -hlavnej vetve, čo je spor s predpokladom lemy.

Teda X je najvzdialenejšia bunka od G . Nech YZ je G -hlavná vetva. Potom $d(Y, Z) = d(G, Y) + d(G, Z)$. Bez ujmy na všeobecnosti nech $d(G, Y) < d(G, Z)$. Nech p_1, p_2 sú takí synovia G , že Y patrí do subtree (p_1) a Z patrí do subtree (p_2).

Ak X patrí do subtree (p_1), a teda nepatrí do subtree (p_2), tak $d(X, Z) = d(X, G) + d(G, Z) \geq d(Y, G) + d(G, Z) = d(Y, Z)$, pričom nerovnosť $d(X, G) \geq d(Y, G)$ vyplýva z toho, že X je najvzdialenejšia od G . Teda $d(X, Z) \geq d(Y, Z)$, čiže ak YZ je hlavná vetva, aj XZ musí byť G -hlavná vetva.

Ak X nepatrí do subtree (p_1) , tak $d(X, Y) = d(X, G) + d(G, Y) \geq d(Z, G) + d(G, Y) = d(Y, Z)$, teda $d(X, Y) \geq d(Y, Z)$, teda XY je G -hlavná vetva.

Keďže $X = p'$ alebo X je potomok bunky p' , musí p' ležať na ceste GX , teda aj na G -hlavnej vetve XZ resp. YZ . \square



Obrázok 4.3:

Lema 4.2.2. Signál S_A je v každom okamihu najviac v 1 bunke.

Dôkaz. Použijeme indukciu vzhľadom na čas. V čase $t = 0$ tvrdenie zjavne platí, lebo generál je len jeden. Nech tvrdenie platí v čase t . Nech v čase t je signál S_A v bunke p (obr. 4.3). Stačí ukázať, že v čase t existuje najviac jeden syn p' bunky p taký, že v p' nečaká signál S'_0 . Sporom, nech p má aspoň dvoch takých synov p_1, p_2 . Potom z lemy 4.2.3 vyplýva, že G neleží na hlavnej vetve, čo je spor. \square

Lema 4.2.3. Nech p je bunka. Nech v čase t do p prišiel signál S_A . Nech p má aspoň dvoch takých synov p_1, p_2 , do ktorých v čase t ešte neprišiel signál S'_0 . Potom G neleží na hlavnej vetve.

Dôkaz. Aká je vzdialenosť $d(p, X_1)$, kde X_1 je najvzdialenejší potomok bunky p_1 (obr. 4.3)? Signál S'_0 vznikol v X_1 v čase $d(G, X_1) = d(G, p) + d(p, X_1)$, do bunky p dorazí o $d(p, X_1)$ taktov neskôr, teda v čase

$$t_1 = d(G, p) + 2d(p, X_1)$$

(Využili sme, že signál S'_0 sa na ceste z X_1 do p nikde nezdrží, vďaka výberu bodu X_1 . Inak by sme vedeli iba to, že $t_1 \geq d(G, p) + 2d(p, X_1)$.) Signál S_A prišiel do bunky p v čase

$$t_2 = 3d(G, p)$$

keďže sa pohybuje rýchlosťou $1/3$. Keďže signál S'_0 v čase t_2 ešte nedorazil do p_1 , platí $t_1 > t_2$. Po dosadení máme $d(G, p) + 2d(p, X_1) > 3d(G, p)$, po úprave $d(p, X_1) > d(G, p)$. Analogicky dokážeme $d(p, X_2) > d(G, p)$. Potom ale cesty pX_1, pX_2 sú obe dlhšie ako Gp , takže G neleží na hlavnej vetve. \square

Lema 4.2.4. *Signál S_A je vždy na G -hlavnej vetve.*

Dôkaz. Opäť použijeme indukciu vzhľadom na čas. Signál S_A štartuje v bode G , teda na G -hlavnej vetve. Ostáva ukázať, že S_A z G -hlavnej vetvy nikdy neodbočí. Nech S_A je v čase t v bunke p_1 a v čase $t + 1$ v bunke p_2 a nech $p_1 \neq p_2$. To znamená, že v čase t je p_2 jediný taký syn bunky p_1 , do ktorého ešte neprišiel signál S'_0 . (Teda vo všetkých ostatných synoch bunky p_1 už čaká signál S'_0 .) Z toho vyplýva, že p_2 si v budúcnosti nastaví príznak hlavnej vetvy (konkrétne 2 takty po tom, čo do p_2 príde signál S'_0). Použitím lemy 4.2.1 dostaneme, že aj p_2 leží na G -hlavnej vetve. \square

Druhá fáza synchronizácia sa začína, keď sa vyznačí bunka A , teda v čase $\frac{3}{2}r$. Simulácia virtuálneho kruhu sa šíri rýchlosťou 1 do buniek, ktoré sú potomkom A a majú nastavený príznak hlavnej vetvy – teda do G -hlavných vetiev. Do ostatných buniek sa šíri simulácia virtuálnych stromov. V bunke B sa šírenie virtuálneho kruhu „otočí“, teda sa začne šíriť aj opačným smerom (tj. aj k späť k bodu A , po virtuálnej ceste BA). Buniek B môže byť viac, ale „otočenie“ nastane iba v hlavných vetvách, teda len v tých bunkách B , ktoré majú nastavený príznak hlavnej vetvy.

Všetky hlavné vetvy sa správajú rovnako, takže druhá fáza synchronizácie prebehne korektne. Nech p je bunka na hlavnej vetve taká, že je to potomok A a že $d(A, p) < d(A, B)$. Označme $i = d(A, p)$. Potom bunka p simuluje dve virtuálne bunky: bunku číslo $i < d(A, p)$ a bunku číslo $j > d(A, p)$. (Kde $j = 2d(A, p) - i$ alebo $j = 2d(A, p) + 1 - i$ v závislosti od hodnoty $r \bmod 4$. Vid' časť 4.2.1.) Nech P_p je množina všetkých synov bunky p ležiacich na hlavnej vetve. Potom každá bunka z P_p simuluje bunky číslo $i + 1$ a $j - 1$. Teda všetky bunky z P_p robia to isté; správanie bunky na hlavnej vetve závisí len od vzdialenosti tejto bunky od A .

Teda máme tieto skupiny reálnych buniek:

1. tie bunky, ktoré sú na hlavnej vetve a zároveň sú potomkami A . Každá z týchto buniek simuluje 2 virtuálne bunky na kruhu.
2. ostatné bunky (tj. všetky bunky, ktoré neležia na G -hlavnej vetve plus bunky na ceste GA). Každá z týchto buniek simuluje 1 virtuálnu bunku. Ak p je taká bunka, tak p simuluje bunku číslo $d(A, p)$.
3. bunka A simuluje jediná bunku – generála
4. bunky B simulujú 1 alebo 2 bunky v závislosti od hodnoty $r \bmod 4$, viď časť 4.2.1

Správanie reálnej bunky p v druhej fáze závisí len od toho, do ktorej z uvedených skupín p patrí a aká je vzdialenosť p od A .

4.2.1 Zaokrúhľovanie

Podrobne preskúmame vývoj na ľubovoľnej G -hlavnej vetve. Nech hlavná vetva je tvorená bunkami p_0, p_1, \dots, p_r , pričom $G = p_0$, $d(p_0, p_i) = i$. Bunky virtuálneho kruhu budeme značiť $p'_0, p'_1, \dots, p'_{r'-1}$, kde r' je dĺžka (tj. počet buniek) virtuálneho kruhu a vo virtuálnej sieti platí $d(p'_0, p'_i) = i$.

Signál S s rýchlosťou $v = k/l$, $k, l \in \mathbb{N}$ implementujeme nasledovne. K signálu S bude priradené počítadlo nadobúdajúce hodnoty $0, 1, \dots, l - 1$. Signál S , ktorého počítadlo má hodnotu x , budeme značiť S^x . Nech v čase t je S^x v bunke p_i . Ak $x + k < l$, tak v čase $t + 1$ bude S^{x+k} v bunke p_i . (Teda signál sa nepohne a počítadlo sa zvýši o k .) Ak $x + k \geq l$ (tj. ak by nastalo pretečenie), signál sa posunie do bunky p_{i+1} a nová hodnota počítadla bude $x + k - l$. Teda ak bunka p_{i+1} vidí, že jej sused p_i obsahuje signál S^n , kde $x + k \geq l$, tak v ďalšom takte bunka p_{i+1} preberie signál S .

Ľahko sa dokáže, že ak S^0 štartuje v bunke p_0 , tak v čase t je $S^{kt \bmod l}$ v bunke $p_{\lfloor kt/l \rfloor}$

Každá bunka si musí pamätať, či v nej je signál S a ak áno, pamätá si aj hodnotu počítadla priradeného k S .

Uvedeným spôsobom implementujeme signály S_A (pre $k = 1, l = 3$) a S_B (pre $k = 3, l = 5$).

signál	$d_G(t)$	počítadlo	pre $t \in$
S_A	$\lfloor \frac{t}{3} \rfloor$	$t \bmod 3$	$\langle 0, t_A - 1 \rangle$
S_B	$\lfloor \frac{3t}{5} \rfloor$	$3t \bmod 5$	$\langle 0, t_B - 1 \rangle$
S_0	t	nemá	$\langle 0, r - 1 \rangle$
S'_0	$2r - t$	nemá	$\langle r, t_A - 1 \rangle$

Tabuľka 4.1:

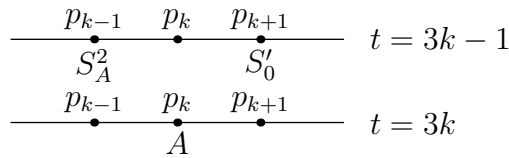
Signál S_0 je v čase t v bunke p_t , teda v čase r dorazí na koniec hlavnej vetvy p_r . V tom istom takte, teda v čase r , vznikne v bunke p_r signál S'_0 . Teda keď bunka p_r uvidí v čase $r - 1$ v bunke p_{r-1} signál S_0 , vygeneruje v čase r signál S'_0 . Signál S'_0 je v čase t v bunke p_{2r-t} , pre $t \geq r$.

Polohy signálov sú zhrnuté v tabuľke 4.1, v stĺpci $d_G(t)$ je vzdialenosť príslušného signálu od G v čase t , v stĺpci „počítadlo“ je hodnota počítadla príslušného signálu v čase t .

Teraz určíme, kde presne vzniknú body A a B . Tiež určíme, v ktorých reálnych bunkách budú umiestnené bunky virtuálneho kruhu. Budeme rozlišovať 4 prípady podľa hodnoty $r \bmod 4$. Bunka A bude v každom prípade simulovať jedinú bunku p'_0 . Bunka B bude simulovať 1 alebo 2 virtuálne bunky v závislosti od $r \bmod 4$; cieľom bude, aby dĺžka virtuálneho kruhu nebola menšia ako vzdialenosť najvzdialenejšej bunky od A .

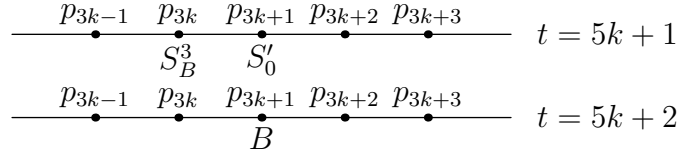
Pre bod A budeme rozlišovať 2 prípady:

1. $r = 2k$. Potom $A = p_{r/2} = p_k$. V čase $3k - 1$ je S'_0 v bunke p_{k+1} a S_A^2 je v p_{k-1} . V nasledujúcom takte sa na mieste S'_0 vyznačí bunka A .



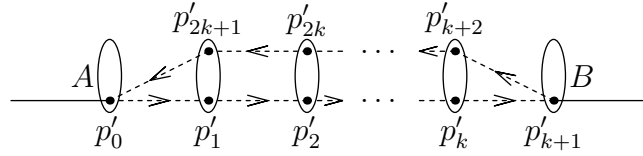
Obrázok 4.4:

2. $r = 2k + 1$. Potom $A = p_{(r-1)/2} = p_k$. V čase $3k + 1$ je S'_0 v bunke p_{k+1} a S_A^1 je v p_k . V nasledujúcom takte sa na mieste S_A vyznačí bunka



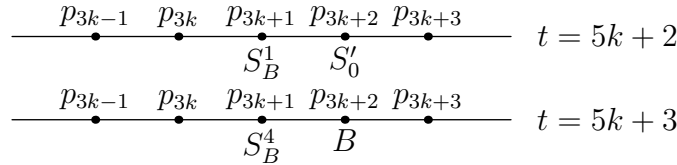
Obrázok 4.8:

Najvzdialenejšiou bunkou od A je p_r , $d(A, p_r) = r - 2k = (4k + 1) - 2k = 2k + 1$. Bunka B bude simulovať 1 bunku $-p'_{k+1}$. Bunka p_{2k+i} bude simulovať bunky p'_i a p'_{2k+2-i} (pre $1 \leq i \leq k$).



Obrázok 4.9:

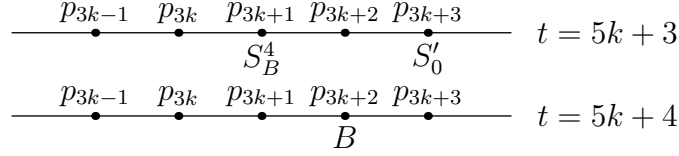
3. $r = 4k + 2$. Potom $A = p_{r/2} = p_{2k+1}$, $B = p_{(3r+2)/4} = p_{3k+2}$. V čase $5k + 2$ je S'_0 v p_{3k+2} a S_B^1 je v p_{3k+1} . V nasledujúcom takte (tj. v čase $5k + 3$) sa S_B^1 zmení na S_B^4 , S'_0 zanikne a v bunke p_{3k+2} sa vyznačí bod B . Signál S_B zanikne v ďalšom takte (tj. v čase $5k + 4$).



Obrázok 4.10:

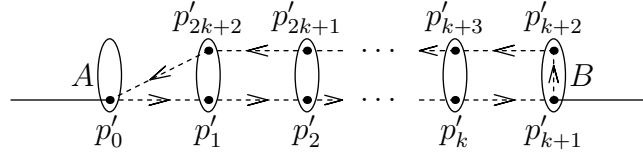
Najvzdialenejšiou bunkou od A je p_r , $d(A, p_r) = r - (2k + 1) = (4k + 2) - 2k - 1 = 2k + 1$. Bunka B bude simulovať 1 bunku $-p'_{k+1}$. Bunka p_{2k+1+i} bude simulovať bunky p'_i a p'_{2k+2-i} (pre $1 \leq i \leq k+1$). Virtuálny kruh vyzerá rovnako ako v predchádzajúcom prípade (teda bunka B bude simulovať jedinou bunku), len je posunutý o 1 bunku doprava.

4. $r = 4k + 3$. Potom $A = p_{(r-1)/2} = p_{2k+1}$, $B = p_{(3r-1)/4} = p_{3k+2}$. V čase $5k + 3$ je S'_0 v p_{3k+3} a S_B^4 je v p_{3k+1} . V nasledujúcom takte sa S_B neposunie, ale v bunke p_{3k+2} vyznačí bod B .



Obrázok 4.11:

Najvzdialenejšiou bunkou od A je p_r , $d(A, p_r) = r - (2k + 1) = 4k + 3 - 2k - 1 = 2k + 2$. Bunka B bude simulovať 2 bunky - p'_{k+1} a p'_{k+2} . Bunka p_{2k+1+i} bude simulovať bunky p'_i a p'_{2k+3-i} (pre $1 \leq i \leq k + 1$).



Obrázok 4.12:

r	A	B	t_A	r'	t_F
$4k$	p_{2k}	p_{3k}	$6k$	$2k + 1$	$10k + 1$
$4k + 1$	p_{2k}	p_{3k+1}	$6k + 2$	$2k + 2$	$10k + 5$
$4k + 2$	p_{2k+1}	p_{3k+2}	$6k + 3$	$2k + 2$	$10k + 6$
$4k + 3$	p_{2k+1}	p_{3k+2}	$6k + 5$	$2k + 3$	$10k + 10$

Tabuľka 4.2:

Zhrnutie je uvedené v tabuľke 4.2. t_A označuje čas nájdenia bunky A , t_F označuje čas synchronizácie. Platí $t_F = t_A + 2r' - 1$, lebo druhá fáza sa začína v čase t_A a synchronizácia virtuálneho kruhu s počtom buniek r' trvá čas $2r' - 1$.

Z predchádzajúceho rozboru sa dá určiť časť prechodovej funkcie týkajúca sa nájdenia vrcholov A , B . Napr. ak nejaká bunka p_X vidí, že v jej rodičovi je signál S_B^2 a v synovi je signál S'_0 , tak v nasledujúcom takte sa bunka

p_X označí za bunku B – toto zodpovedá prípadu $r = 4k$. Dôležité je, že jednotlivé prípady sa dajú rozlíšiť podľa hodnoty počítadla signálu S_B .

Lema 4.2.5. *Virtuálna sieť je kvázicyklická sieť dĺžky r' .*

Dôkaz. Dokážeme, že vo virtuálnej sieti nie je žiadna bunka vzdialená od A viac ako r' . Najprv to ukážeme pre ľubovoľnú G -hlavnú vetvu. Stačí preveriť, že v tab. 4.2 platí, že ak A je v bunke p_i , tak $i < r'$ (teda $d(p_0, A) \leq r'$) a $r - i \leq r'$ (teda $d(p_r, A) < r'$).

Ak by pre nejakú bunku p , ktorá neleží na žiadnej G -hlavnej vetve, platilo $d(A, p) \geq r'$, tak buď p_0p alebo $p_r p$ je dlhšia ako hlavná vetva, čo je spor. \square

4.3 Generál vnútri hlavnej vetvy

V tejto časti sa budeme zaoberať synchronizáciou stromov, pričom generál je na hlavnej vetve, ale nie nutne na kraji. Budeme sa snažiť prispôbiť riešenie z časti 4.2, aby fungovalo aj pre uvedenú triedu. Pritom druhú fázu spomínaného riešenia ponecháme bez zmeny, teda úlohou bude len vyznačiť bunky A a B .

Nech XY je ľubovoľná G -hlavná vetva taká, že $d(G, X) \leq d(G, Y)$. Budeme sa zaoberať len prípadom, keď $d(G, X) < \frac{1}{3}d(G, Y)$, v ostatných prípadoch je výhodnejšie použiť riešenie z časti 3.4. Bunky cesty XY označíme po rade p_0, \dots, p_{n-1} , kde $d(p_0, p_i) = i$, $X = p_0$, $G = p_l$, $Y = p_{n-1}$, $n = l + r$, r je polomer stromu. Keď opisujeme vývoj na G -hlavnej vetve, myslíme tým ľubovoľnú G -hlavnú vetvu XY , lebo všetky G -hlavné vetvy sa správajú rovnako. Tento postup je korektný vďaka lema 4.3.1. Tá hovorí, že ak XY je G -hlavná vetva, tak dĺžky $d(X, G)$ a $d(Y, G)$ nezávisia od výberu X a Y .

Lema 4.3.1. *Nech Z_1Z_2 a Z_3Z_4 sú G -hlavné vetvy, pričom $d(Z_i, G) \leq d(Z_{i+1}, G)$ pre $i = 1, 3$. Potom $d(Z_1, G) = d(Z_3, G)$, $d(Z_2, G) = d(Z_4, G)$.*

Dôkaz. Nech p_1, p_2, p_3, p_4 sú takí synovia bunky G , že Z_i leží v subtree (p_i) pre $i = 1, 2, 3, 4$.

Skúmame hodnoty $\text{depth}(p_i)$. Platí $\text{depth}(\text{subtree}(p_i)) \geq d(G, Z_i)$, lebo Z_i leží v subtree (p_i) . Tiež platí $\text{depth}(\text{subtree}(p_i)) \leq d(G, Z_i)$ lebo inak

by sme vedeli predĺžiť príslušnú hlavnú vetvu. Uvažujme napr. Z_1 . Keby $\text{depth}(\text{subtree}(p_1)) > d(G, Z_1)$, tak v $\text{subtree}(p_1)$ existuje bunka Z taká, že $d(G, Z) > d(G, Z_1)$. Keďže G leží na ceste Z_1Z_2 , tak Z_2 neleží v $\text{subtree}(p_1)$ a teda $d(Z, Z_2) = d(Z, G) + d(G, Z_2) > d(G, Z_1) + d(G, Z_2) = d(Z_1, Z_2)$. Teda $d(Z, Z_2) > d(Z_1, Z_2)$, čo je spor s tým, že Z_1Z_2 je hlavná vetva. Analogicky sa dokazuje $\text{depth}(\text{subtree}(p_i)) \leq d(G, Z_i)$ pre $i = 2, 3, 4$. Spojením oboch nerovností dostaneme $\text{depth}(\text{subtree}(p_i)) = d(G, Z_i)$.

Kvôli sporu predpokladajme

$$d(Z_1, G) < d(Z_3, G) \quad (4.1)$$

Preto aj $\text{depth}(\text{subtree}(p_1)) < \text{depth}(\text{subtree}(p_3))$, teda $p_1 \neq p_3$.

Ak Z_1 leží v $\text{subtree}(p_4)$, tak $p_1 = p_4$. Preto $\text{depth}(\text{subtree}(p_1)) = \text{depth}(\text{subtree}(p_4))$, teda

$$d(G, Z_1) = d(G, Z_4) \quad (4.2)$$

Z predpokladov lemy vieme, že $d(G, Z_3) \leq d(G, Z_4)$, dosadením 4.2 dostaneme $d(G, Z_3) \leq d(G, Z_1)$, čo je spor s 4.1.

Ak Z_2 leží v $\text{subtree}(p_3)$, tak $p_2 = p_3$. Preto $\text{depth}(\text{subtree}(p_2)) = \text{depth}(\text{subtree}(p_3))$, teda

$$d(G, Z_2) = d(G, Z_3) \quad (4.3)$$

Pritom Z_4 neleží v $\text{subtree}(p_3) = \text{subtree}(p_2)$, lebo cesta Z_3Z_4 ide cez G . Preto $d(Z_1, Z_2) = d(G, Z_1) + d(G, Z_2) = d(G, Z_1) + d(G, Z_3) < d(G, Z_3) + d(G, Z_3) \leq d(G, Z_3) + d(G, Z_4) = d(Z_3, Z_4)$. Druhá rovnosť vyplýva z 4.3, ostrá nerovnosť vyplýva z 4.1. Teda $d(Z_1, Z_2) < d(Z_3, Z_4)$, čo je spor s tým, že Z_1Z_2 je hlavná vetva.

Ak Z_2 neleží v $\text{subtree}(p_3)$, tak $d(Z_2, Z_3) = d(Z_2, G) + d(Z_3, G) > d(Z_2, G) + d(Z_1, G) = d(Z_1, Z_2)$, pričom nerovnosť vyplýva z 4.1. Teda $d(Z_2, Z_3) > d(Z_1, Z_2)$, čo je spor s tým, že Z_1Z_2 je hlavná vetva.

Teda 4.1 neplatí. Analogicky odvodíme spor aj pre $d(Z_1, G) > d(Z_3, G)$. Preto $d(Z_1, G) = d(Z_3, G)$. Keďže všetky hlavné vetvy majú z definície

rovnakú dĺžku, platí aj $d(Z_2, G) = d(Z_4, G)$. \square

Hlavná myšlienka prvej fázy riešenia je podobná ako v časti 3.1.4. Vytvoríme signály S_A a S_B tak, ako keby boli vyslané z bunky X v čase $-l$.

Generál v čase 0 vyšle signál S_0 , ktorý sa správa tak, ako je opísané v časti 4.2. Keď sa S'_0 dostane do syna bunky G , tak v nasledujúcom takte zanikne.

Signál S'_0 tiež nastavuje príznak hlavnej vetvy, v súlade s lemov 4.2.1. Vďaka tomu dokáže G poslať správu len bunkám na G -hlavných vetvách. Ale okrem toho je potrebné, aby G dokázala poslať signál smerom len k X alebo len k Y . Na to musí každý syn bunky G vedieť, či leží na ceste do X , resp. do Y .

Za týmto účelom bude mať bunka G počítadlo, kde si pamätá, z koľkých susedov ešte nedostala signál S'_0 . Na začiatku je toto počítadlo nastavené na počet susedov bunky G a pri príchode signálu S'_0 do syna bunky G počítadlo klesne o 1. Keď syn p bunky G uvidí, že tento počet klesol na 0 (resp. 1) jeden takt po tom, ako bol v p signál S'_0 , tak p vie, že $\text{depth}(\text{subtree}(p)) + 1 = d(G, Y)$ (resp. $\text{depth}(\text{subtree}(p)) + 1 = d(G, X)$).

Pri rekonštrukciu signálov S_A resp. S_B použijeme signál S''_0 resp. S'_B . Signál S''_0 vznikne v bunke G v čase $2l$, teda pri príchode S'_0 z X do G . Signál S''_0 si môžeme predstaviť ako pokračovanie signálu S'_0 . Signály S'_0 a S''_0 rozlišujeme najmä preto, že S''_0 vie, že už prešiel cez bunku G .

Šírenie signálu S''_0 . Signál S''_0 je vygenerovaný v bunke G vtedy, keď počítadlo bunky G klesne na 1. Syn p bunky G preberie signál S''_0 v nasledujúcom takte, ale len vtedy, ak v p ešte nebol signál S'_0 . Ďalej sa S''_0 šíri broadcastom.

Nech v čase $2l - 1$ má počítadlo bunky G hodnotu x . V čase $2l$ má toto počítadlo hodnotu 1. To vyplýva z lemy 4.3.2, lebo v čase $2l - 1$ musel do nejakého syna G doraziť signál S'_0 z každého podstromu s hĺbkou najviac l . Teda ak do času $2l - 1$ signál S'_0 nedorazil do p a p je syn G , tak $\text{depth}(\text{subtree}(p)) + 1 > l$, a podľa lemy 4.3.2 existuje práve jedno také p . Signál S''_0 je v čase $2l + 1$ prevzatý touto bunkou p a ďalej sa šíri len v $\text{subtree}(p)$.

Šírenie signálu S'_B . Signál S'_B je vygenerovaný v bunke G v čase 0 a šíri sa broadcastom. Pohyb S'_B na ceste GY je opísaný v tabuľke 4.3. Skúmame pohyb S'_B na ceste GZ , kde Z je list a $d(G, Z) < d(G, Y)$. Na cestu GZ sa nikdy nedostane signál S''_0 , preto sa na ceste GZ nevytvorí signál S_B . Signál S'_B príde až do Z a tam zanikne.

Lema 4.3.2. *Ak $d(G, X) < d(G, Y)$, tak existuje práve jeden taký syn p bunky G taký, že $\text{depth}(\text{subtree}(p)) + 1 > d(G, X)$.*

Dôkaz. Existencia ukážeme tak, že zoberieme také p , že Y leží v subtree(p).

Jednoznačnosť dokážeme sporom. Nech existujú dvaja rôzni takí synovia p_1, p_2 . Nech Z_i je najvzdialenejší potomok p_i pre $i = 1, 2$. Bez ujmy na všeobecnosti nech p_1 je taký, že Y neleží v subtree(p_1), a teda cesta Z_1Y ide cez G . Keďže $\text{depth}(\text{subtree}(p_1)) + 1 > d(G, X)$, tak $d(G, Z_1) > d(G, X)$. Potom $d(Z_1, Y) = d(Z_1, G) + d(G, Y) > d(G, X) + d(G, Y) = d(X, Y)$, teda $d(Z_1, Y) > d(X, Y)$, čo je spor s tým, že XY je hlavná vetva. \square

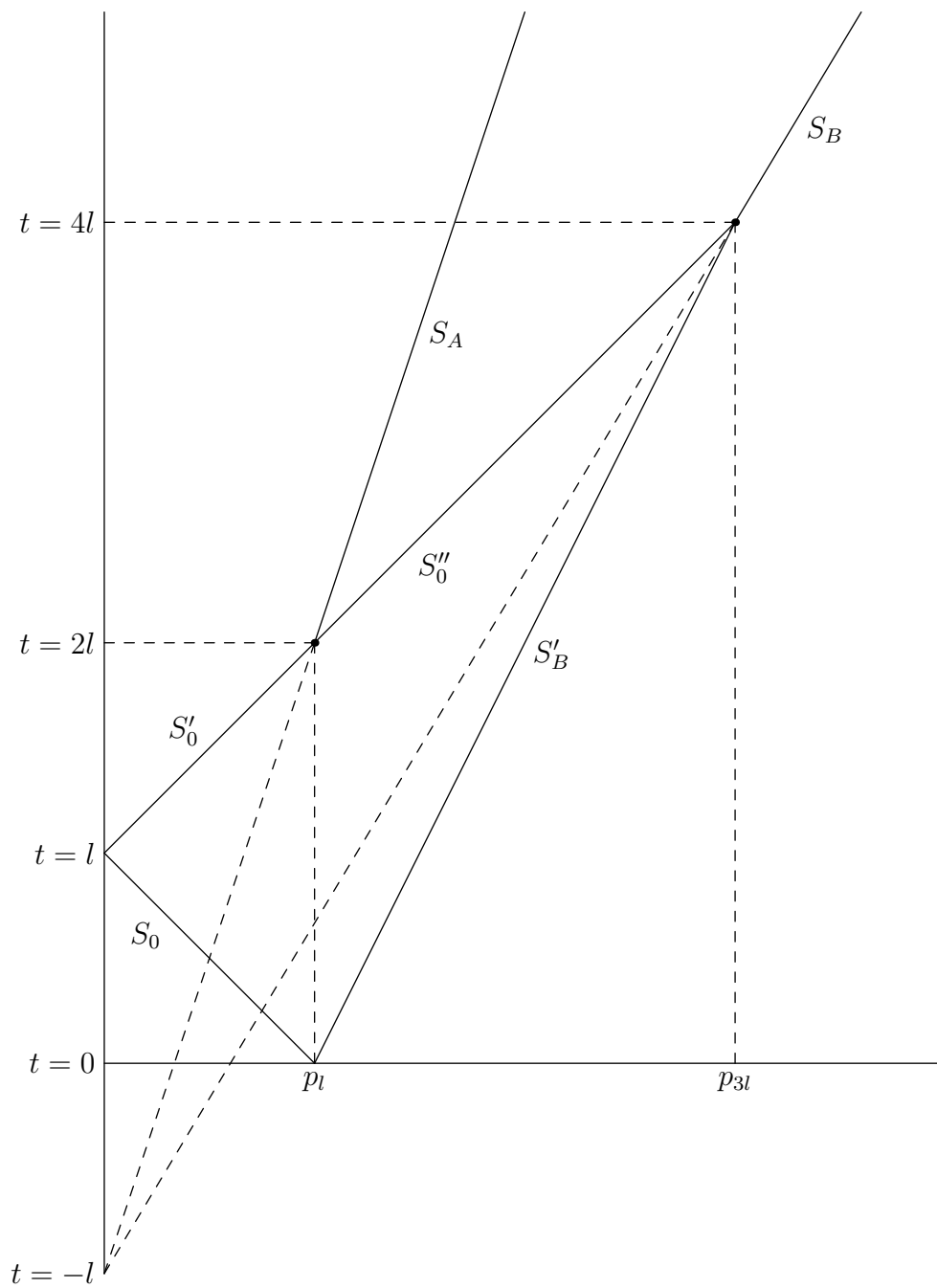
Rekonštrukcia S_A . Signál S_A vznikne v bunke $G = p_l$ v čase $2l$, teda pri príchode signálu S'_0 do G .

Rekonštrukcia S_B . Signál S_B vznikne v bunke p_{3l} v čase $4l$, teda pri stretnutí signálov S''_0 a S'_B . Za stretnutie považujeme, keď sú S''_0 a S'_B v tej istej bunke a hodnota počítadla S'_B je 0. Viď tabuľka 4.3 a obr. 4.13.

Všimnime si, že keďže $l < 3r$, tj. $l < 4n$, tak signál S_B vznikne včas, teda skôr ako by sme mali nájsť bunku B .

signál	poloha	počítadlo	pre $t \in$
S_0	$l - t$	nemá	$\langle 0, l - 1 \rangle$
S'_0	$t - l$	nemá	$\langle l, 2l - 1 \rangle$
S''_0	$t - l$	nemá	$\langle 2l, 4l - 1 \rangle$
S_A	$\lfloor \frac{t+l}{3} \rfloor$	$(t + l) \bmod 3$	$\langle 2l, t_A \rangle$
S'_B	$\lfloor \frac{t}{2} \rfloor + l$	$t \bmod 2$	$\langle 2l, 4l - 1 \rangle$
S_B	$\lfloor \frac{3}{5}(t + l) \rfloor$	$3(t + l) \bmod 5$	$\langle 4l, t_B \rangle$

Tabuľka 4.3: Polohy signálov t_A resp. t_B označuje čas nájdenia bunky A resp. B

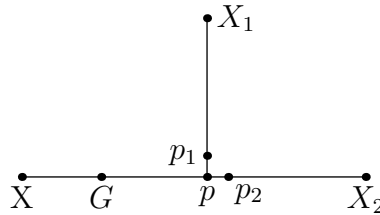


Obrázok 4.13:

Signály S_A a S_B sa po svojom vzniku šíria podľa rovnakých pravidiel ako v časti 4.2. Z lemm 4.3.3 a 4.2.2 vyplýva, že signál S_A je v každom okamihu najviac v jednej bunke na G -hlavnej vetve. Preto sa vyznačí len jedna bunka A .

Zaoberali sme sa len vývojom na G -hlavných vetvách. Vývojom na ostatných bunkách sa nemusíme zaoberať. Druhá fáza synchronizácie potrebuje len vyznačenie jedinej bunky A a buniek B_i na G -hlavných vetvách a naštartovanie druhej fázy v bunke A . Tieto požiadavky opísaná prvá fáza spĺňa.

Lema 4.3.3. *Nech p je bunka. Nech v čase t do p prišiel signál S_A . Nech p má aspoň dvoch takých synov p_1, p_2 , do ktorých v čase t ešte neprišiel signál S'_0 . Potom G neleží na hlavnej vetve.*



Obrázok 4.14:

Dôkaz. Sporom. Nech XY je G -hlavná vetva, nech $d(G, X) \leq d(G, Y)$. Nech X_i je najvzdialenejší potomok bunky p_i pre $i = 1, 2$. Aká je vzdialenosť $d(p, X_1)$ (obr. 4.14)? Signál S'_0 vznikol v X_1 v čase $d(G, X_1) = d(G, p) + d(p, X_1)$, do bunky p dorazí o $d(p, X_1)$ taktov neskôr, teda v čase

$$t_1 = d(G, p) + 2d(p, X_1)$$

(Využili sme, že signál S'_0 sa na ceste z X_1 do p nikde nezdrží, vďaka výberu bodu X_1 . Inak by sme vedeli iba to, že $t_1 \geq d(G, p) + 2d(p, X_1)$.) Signál S_A prišiel do bunky p v čase

$$t_2 = 2d(X, G) + 3d(G, p)$$

keďže sa pohybuje rýchlosťou $1/3$ a začína v čase $2d(X, G)$. Keďže signál S'_0 v čase t_2 ešte nedorazil do p_1 , platí $t_1 > t_2$. Po dosadení máme $d(G, p) +$

$2d(p, X_1) > 2d(X, G) + 3d(G, p)$, po úprave $d(p, X_1) > d(X, G) + d(G, p)$. Analogicky dokážeme $d(p, X_2) > d(X, G) + d(G, p)$.

Bez ujmy na všeobecnosti môžeme p_1 a p_2 zvoliť tak, aby Y neležala v subtree(p_1). Potom $d(X_1, Y) = d(X_1, p) + d(p, Y) > d(X, G) + d(G, p) + d(p, Y) = d(X, G) + d(G, p) + d(p, Y) = d(X, Y)$, teda $d(X_1, Y) > d(X, Y)$. Teda XY nemôže byť hlavná vetva. \square

Čas synchronizácie. Riešenie v tejto časti sme konštruovali tak, aby fungovalo ako riešenie z časti 4.2 s generálom na kraji hl. vetvy naštartované v čase $-l$. Takže čas riešenia bude $(2 + \frac{1}{2})n - l = (2 + \frac{1}{2})(r + l) - l = (2 + \frac{1}{2})r - \frac{3}{2}l$. Je zrejmé, že pre $l < r/3$ je toto riešenie rýchlejšie ako všeobecné riešenie z časti 3.4.

4.4 Vnoriťnosť

V tejto časti ukážeme, že riešenia uvedené v častiach 4.2 a 5.1 sú vnoriťné v zmysle definície 2.4.1. Vďaka vnoriťnosti môžeme paralelne spustiť riešenie pre všeobecný strom (uvedené v časti 3.4) aj riešenie pre strom s generálom na konci hlavnej vetvy (uvedené v časti 5.1). Takýmto spôsobom získame riešenie, ktoré vystrelí v čase $(2 + \frac{1}{2})r$ (resp. $(2 + \frac{1}{4})r$ pri použití riešenia z časti 5.1), ak je generál na konci hlavnej vetvy a inak vystrelí v čase $3r$.

Postup bude nasledovný. Označme Γ triedu prípustných sietí, teda do Γ patrí každý strom s generálom na hlavnej vetve. Ak nejaká bunka p zistí, že daná sieť nepatrí do Γ , tak p prejde do chybového stavu s_X . Stav s_X sa bude šíriť broadcastom. Teda ak bunka p_1 zistí, že jej susedná bunka p_2 je v stave s_X , v nasledujúcom takte prejde p_1 do stavu s_X . Stav s_X je stabilný, teda ak je bunka v stave s_X , tak v ňom zotrúva bez ohľadu na stav susedov.

Bunka môže prejsť do stavu s_X v nasledujúcich prípadoch.

Detekcia cyklov

Najprv uvažme prípad, že daná sieť nie je strom, teda obsahuje cyklus. Použijeme prehľadávanie do šírky. Ak sa nájde cyklus, bunky na konci cyklu

prejdú do stavu s_X .

Uvažujme najprv cyklus nepárnej dĺžky. V tomto prípade sa dve bunky cyklu, ktoré sú najvzdialenejšie od generála, prebudia v rovnakom čase. Nech t_W je čas prebudenia bunky p . p si pri prebudení (tj. pri prechode z času $t_W - 1$ do času t_W) zapamätá, ktorý sused p_1 ju prebudil. Ak v čase t_W je prebudený aj iný sused bunky p ako p_1 , tak p zistila cyklus a prejde do stavu s_X .

V prípade cyklu párnej dĺžky bunka zistí, že aspoň dvaja jej susedia sa prebudili v rovnakom čase a v takom prípade prejde do stavu s_X .

Takže ak sieť obsahuje cyklus, nejaká bunka prejde do stavu s_X najneskôr v čase $r + 1$. Stav s_X sa rozšíri do celej siete najneskôr v čase $2r + 1$. Keďže naše riešenia pracujú dlhšie ako $2r + 1$, tak žiadna bunka v cyklickom grafe nevystrelí.

Detekcia hlavnej vetvy

Prípad, keď generál nie je v liste, sa dá ošetriť triviálne – ak generál vidí, že má aspoň 2 susedov, tak prejde do stavu s_X .

Nakoniec predpokladajme, že daná sieť je strom, generál je v liste, ale generál nie je na hlavnej vetve. Treba ukázať, že S_A sa niekedy „rozdvojí“. To vyplýva z lemy 4.4.1. Keď totiž signál S_A príde v čase t do bunky p (ktorej existencia je zaručená lemov 4.4.1), tak signál S'_0 ešte nestihol doraziť z X_1 do p ani z X_2 do p . Takže p má aspoň dvoch takých synov p_1, p_2 , v ktorých nečaká signál S'_0 . V tomto prípade S_A nepostúpi do všetkých takých synov, ale namiesto toho bunka p prejde do stavu s_X . Synovia bunky p nepreberú signál S_A , lebo počítadlo signálu má hodnotu 0. Takže sa nevyznačí bunka A , nezačne sa druhá fáza synchronizácie a preto žiadna bunka nevystrelí.

Z lemy 4.4.1 vyplýva, že takýmto spôsobom zrušíme synchronizáciu pre každý neprípustný strom. Z lemy 4.2.3 vyplýva opačná implikácia, teda že takýmto spôsobom zrušíme synchronizáciu len pre neprípustné stromy. (V tomto odseku sa zaoberáme len prípadom, keď sieť je strom. Prípad, keď sieť nie je strom, sme už vyriešili vyššie.)

Lema 4.4.1. *Nech sieť je strom, nech bunka G neleží na žiadnej hlavnej vetve. Potom existujú také bunky p, X_1, X_2 že cesty Gp a X_1X_2 majú spoločnú len bunku p a $d(G, p) < d(X_1, p)$ a $d(G, p) < d(X_2, p)$. (obr. 4.3)*

Dôkaz. Stačí za X_1 a X_2 zobrať také bunky, že X_1X_2 je hlavná vetva. Tým je jednoznačne určené p , keďže cesty Gp a X_1X_2 majú spoločnú len bunku p .

Predpokladajme kvôli sporu, že $d(G, p) > d(X_1, p)$. Potom $d(G, X_2) > d(X_1, X_2)$, a teda X_1X_2 nemôže byť hlavná vetva, čo je spor.

Predpokladajme, že $d(G, p) = d(X_1, p)$. Potom $d(G, X_2) = d(X_1, X_2)$, teda aj GX_2 je hlavná vetva. Dostali sme spor s tým, že G neleží na hlavnej vetve.

Preto musí platiť $d(G, p) < d(X_1, p)$ a analogicky sa dokáže $d(G, p) < d(X_2, p)$. □

Kapitola 5

Možnosti vylepšenia

V tejto kapitole ukážeme, že riešenie z časti 4.2 sa pravdepodobne dá zrýchliť. Zrýchlené riešenie však nie je úplne popísané. Využívame totiž hypotézu, že ak v kruhu dĺžky n je vyznačená bunka v strede, celý kruh sa dá synchronizovať v čase $\frac{3}{2}n$. V časti 3.1.6 sme uviedli automat, ktorý pracuje v požadovanom čase, ale synchronizuje len polovicu kruhu. Dá sa predpokladať, že tento automat je možné upraviť tak, aby v rovnakom čase $\frac{3}{2}n$ synchronizoval celý kruh.

Z uvedeného dôvodu nemôžeme tvrdiť, že máme riešenie s časom blížiacim sa k $(2 + \frac{1}{4})r$. Ukážeme však, že ak požadované riešenie pre kruh existuje, tak riešenie uvedené v časti 5.1 funguje.

5.1 Synchronizácia stromu v čase blížiacom sa k $(2 + \frac{1}{4})r$

V tejto časti zrýchlime riešenie uvedené v časti 4.2. Pritom predpokladáme, že jednosmerný kruh sa dá synchronizovať v čase $\frac{3}{2}r$, ak je na kruhu vyznačená bunka M na pozícii $n/2$, teda oproti generálovi.

Takisto ako v časti 4.2 nájdeme v prvej fáze body A , B a v druhej fáze simulujeme kvázicyklickú sieť s kruhom na ceste AB . Navyiac v prvej fáze vyznačíme na virtuálnom kruhu bunku v strede. Táto bunka bude práve v bunke B . Keďže však nesynchronizujeme len kruh, ale kvázicyklickú sieť,

je potrebné, aby vo virtuálnej sieti na každej orientovanej ceste z A bola vo vzdialenosti $d(A, B)$ značka. Teda treba nájsť všetky body B_i také, že $d(A, B_i) = d(A, B)$.

Ako nájdeme body B_i ? Hneď ako sa vyznačí bunka B na hlavnej vetve, vyšľú sa z bunky B dva signály S_1, S_2 s rýchlosťami 1, resp. v . Tieto signály sa budú šíriť z bunky do jej rodiča, teda smerom k bunke A . Keď signál S_1 (resp. S_2) príde do A , vygeneruje sa tu signál S'_1 (resp. S'_2) s rýchlosťou v (resp. 1). S'_1 a S'_2 sa šíria broadcastom. Za predpokladu $v < 1$ sa signály S'_1, S'_2 stretnú vo vzdialenosti $d(A, B)$ od A , čiže práve v bunkách B_i . Čím väčšie bude číslo v , tým rýchlejšie sa nájdú bunky B_i . Pre v blížiac sa zdola k 1 sa bude potrebný čas (od nájdania bunky B po nájdanie buniek B_i) blížiť zhora k $2d(B, B_i) = 2d(A, B) = r/2$.

Bunka A spustí simuláciu virtuálneho kruhu až vtedy, keď do A dorazí signál S_2 . Toto oneskorenie je nutné, lebo bunky B_i je potrebné vyznačiť najneskôr vtedy, keď sa virtuálne bunky simulované bunkami B_i prebudia.

Nech t_B je čas nájdania bunky B , nech $v = \frac{m-1}{m}$, $m \in \mathbb{N}$. Signály S_2, S'_1 (šíriace sa rýchlosťou v) implementujeme podobne ako S_A a S_B , teda budú mať počítadlo nadobúdajúce hodnoty $0, \dots, m-1$ a inkrementujúce sa v každom takte o $m-1$. Rozdiel oproti S_A a S_B je v smere a spôsobe šírenia – S_2 sa šíri smerom k A a S'_1 sa šíri broadcastom. Pri svojom vzniku si signál S'_2 zapamätá hodnotu počítadla signálu S_2 . Táto hodnota sa pri šírení S'_2 nemení. Stretnutie signálov S'_1 a S'_2 nastane (a bunka B_i sa vyznačí) až vtedy, keď S'_1 a S'_2 sú v tej istej bunke a majú rovnakú hodnotu počítadla.

V tabuľke 5.1 sú zhrnuté polohy signálov. $d_A(t)$ označuje vzdialenosť daného signálu od bunky A v čase t , v ďalšom stĺpci je hodnota počítadla v čase t . Označili sme $d = d(A, B)$. Pri výpočte hodnoty počítadla sme využili, že $k(m-1) \bmod m = -k$. Operáciu mod chápeme tak, že vždy dáva nezáporný výsledok.

Z tabuľky 5.1 vyplýva, že S'_1 vznikne v čase $t_B + d$, čiže o 1 takt neskôr, ako sa S_1 dostane do suseda bunky A .

signál	$d_A(t)$	počítadlo	pre $t \in$
S_1	$d - (t - t_B)$	nemá	$\langle t_B, t_B + d - 1 \rangle$
S'_1	$\lfloor (t - t_B - d)v \rfloor$	$(t_B + d - t) \bmod m$	$\langle t_B + d, t_B + d + \lceil \frac{d}{v} \rceil \rangle$
S_2	$d - \lfloor (t - t_B)v \rfloor$	$(t_B - t) \bmod m$	$\langle t_B, t_B + \lceil \frac{d}{v} \rceil - 1 \rangle$
S'_2	$t - t_B - \lceil \frac{d}{v} \rceil$	$(-\lceil \frac{d}{v} \rceil) \bmod m$	$\langle t_B + \lceil \frac{d}{v} \rceil, t_B + d + \lceil \frac{d}{v} \rceil \rangle$

Tabuľka 5.1:

Signál S'_2 vznikne v čase $t_B + \lceil \frac{d}{v} \rceil$, čiže o 1 takt neskôr, ako sa S_2 dostane do suseda bunky A (viď lema 5.1.2). Signál S'_2 si svoje počítadlo nastaví na $(-\lceil \frac{d}{v} \rceil \bmod m)$ – toto je hodnota, ktorú by malo počítadlo signálu S_2 v čase $t_B + \lceil \frac{d}{v} \rceil$ (vtedy však už S_2 neexistuje). Takže keď A vidí vo svojom susedovi signál S'_2 , tak A v nasledujúcom takte vytvorí signál S'_2 a jeho počítadlo nastaví na $(n + m - 1) \bmod m$.

Lema 5.1.1. $(\forall d \in \mathbb{N}, v \in (0, 1)) \lfloor \lceil \frac{d}{v} \rceil v - v \rfloor = d - 1$

Dôkaz. Z definície hornej celej časti vyplýva

$$\frac{d}{v} \leq \left\lceil \frac{d}{v} \right\rceil < \frac{d}{v} + 1$$

$$d \leq \left\lceil \frac{d}{v} \right\rceil v < d + v$$

$$d - v \leq \left\lceil \frac{d}{v} \right\rceil v - v < d$$

Z toho vidieť, že $\lfloor \lceil \frac{d}{v} \rceil v - v \rfloor = d - 1$. □

Lema 5.1.2. V čase $t_B + \lceil \frac{d}{v} \rceil - 1$ je signál S_2 vo vzdialenosti 1 od A .

Dôkaz. V čase t je S_2 vo vzdialenosti $d - \lfloor (t - t_B)v \rfloor$ od A . Pre $t = t_B + \lceil \frac{d}{v} \rceil - 1$ platí $d - \lfloor (t - t_B)v \rfloor = d - \lfloor (t_B + \lceil \frac{d}{v} \rceil - 1 - t_B)v \rfloor = d - \lfloor (\lceil \frac{d}{v} \rceil - 1)v \rfloor = d - \lfloor \lceil \frac{d}{v} \rceil v - v \rfloor = d - (d - 1) = 1$, pričom v predposlednej rovnosti sme použili lemu 5.1.1. □

Kde sa stretnú S'_1 a S'_2 ? Najprv zistíme, kedy budú v tej istej bunke, teda pre aké t platí

$$\lfloor (t - t_B - d)v \rfloor = t - t_B - \left\lceil \frac{d}{v} \right\rceil \quad (5.1)$$

Z definície dolnej celej časti platí:

$$\begin{aligned} (t - t_B - d)v - 1 &< \lfloor (t - t_B - d)v \rfloor \\ tv - t_B v - dv - 1 &< t - t_B - \left\lceil \frac{d}{v} \right\rceil \\ t_B(1 - v) - dv + \left\lceil \frac{d}{v} \right\rceil - 1 &< t(1 - v) \\ t &> t_B + \frac{\left\lceil \frac{d}{v} \right\rceil - dv - 1}{1 - v} \end{aligned} \quad (5.2)$$

(delenie oboch strán nerovnosti výrazom $1 - v$ je v poriadku, lebo $v \in (0, 1)$ a teda $v > 0$)

Kedy majú S'_1 a S'_2 rovnakú hodnotu počítadla?

$$\begin{aligned} \overbrace{(t_B + d - t) \bmod m}^{\text{počítadlo } S'_1} &= \overbrace{- \left\lceil \frac{d}{v} \right\rceil \bmod m}^{\text{počítadlo } S'_2} \\ (t_B + d + \left\lceil \frac{d}{v} \right\rceil - t) \bmod m &= 0 \\ t = km + d + t_B + \left\lceil \frac{d}{v} \right\rceil &\text{ pre } k \in \mathbb{N} \end{aligned} \quad (5.3)$$

Ľahko sa overí, že $t_2 = t_B + d + \left\lceil \frac{d}{v} \right\rceil$ spĺňa podmienky 5.1 aj 5.3. Navyše t_2 je najmenšie také prirodzené číslo, keďže najväčšie riešenie rovnice 5.3 menšie ako t_2 je $(t_2 - m)$ a $(t_2 - m)$ nespĺňa nerovnosť 5.2. Takže S'_1 a S'_2 sa prvýkrát stretnú v čase t_2 vo vzdialenosti $d = d(A, B)$ od A .

Čas synchronizácie. Odhadneme čas synchronizácie, zanedbávajúc aditívne konštanty. Druhá fáza štartuje v čase $t_B + \left\lceil \frac{d}{v} \right\rceil$, čo je zhruba $\frac{5}{4}r + \frac{r}{4v}$.

Druhá fáza trvá zhruba $\frac{3}{2} \left(\frac{r}{2}\right) = \frac{3}{4}r$. Celkovo to je $2r + \frac{r}{4v}$, čo pre v blížiacu sa zdola k 1 dáva čas blížiaci sa zhora k $(2 + \frac{1}{4})r$.

5.2 Synchronizácia stromu v čase $(2 + \frac{1}{4})r$

V tejto časti naznačíme, ako sa dá urýchliť hľadanie buniek B_i bez neobmedzeného nárastu počtu stavov.

Predpokladajme najprv, že v každej bunke máme k dispozícii automat s nekonečným počtom stavov. V takom prípade môžeme vyslať z bodu B smerom do bodu A signál S_3 , ktorý si pamätá prejdenú vzdialenosť. Keď S_3 príde do A , vie, že prešiel vzdialenosť $d(B, A)$. Vtedy sa vygeneruje signál S'_3 , ktorého úlohou je prejsť (zapamätanú) vzdialenosť $d(B, A)$ a potom vyznačiť body B_i . Teda S_3 má počítadlo a pri svojom pohybe ho inkrementuje. Po príchode do bodu A sa S_3 zmení na S'_3 a pri svojom pohybe od bodu A bude počítadlo dekrementovať. Keď počítadlo dosiahne hodnotu 0, signál S'_3 vyznačí bod B_i . Signál S_3 sa šíri smerom k bodu A , teda k rodičom. Signál S'_3 sa šíri broadcastom.

Uvedený postup vieme realizovať aj s konečnostavovými automatmi. Keďže hodnota počítadla môže byť neobmedzene veľká, táto hodnota bude uložená vo viacerých susedných bunkách. Počítadlo môžeme implementovať podobne ako v [12], ale potrebujeme navyše, aby sme počítadlo vedeli aj dekrementovať a testovať na 0. Dôležité je, že pri dekrementácii sa dĺžka počítadla (tj. počet buniek, v ktorých je hodnota počítadla uložená) znižuje. Keď počítadlo dosiahne hodnotu 0, bude mať dĺžku 2.

Kapitola 6

Záver

V tejto práci sme ukázali, že problém synchronizácie sa dá pre stromy riešiť v čase $(2 + \frac{1}{2})r$, ak je generál na špeciálnej pozícii – na kraji hlavnej vetvy. Navyše sme ukázali vnoriteľnosť tohto riešenia, čo umožňuje skombinovať naše riešenie s inými riešeniami. Napokon sme uviedli zovšeobecnenie nášho riešenia, kde pripúšťame generála aj vnútri hlavnej vetvy. Tu sme dosiahli čas $(2 + \frac{1}{2})r + \frac{3}{2}l$, kde l je vzdialenosť generála od bližšieho kraja hlavnej vetvy. Pre $l < \frac{1}{3}r$ je tento výsledok lepší ako všeobecné riešenie s časom $3r$.

Sformulovali sme hypotézu o synchronizácii jednosmerných kruhov, ktorej platnosť by umožnila ďalšie zrýchlenie až na čas $(2 + \frac{1}{4})r$.

Ostala otvorená otázka, či je možné strom s generálom na kraji hlavnej vetvy synchronizovať rýchlejšie ako $(2 + \frac{1}{4})r$. Triviálny dolný odhad, totožný s dolným odhadom pre úsečku, je $2r$. Ostáva teda úloha nájsť rýchlejšie riešenie, alebo (čo je nepravdepodobné) lepší dolný odhad.

Ďalšia zaujímavá otázka je, či sa dajú postupy z tejto práce použiť aj na všeobecných grafoch so špeciálnou pozíciou generála.

Literatúra

- [1] Robert Balzer. An 8-state minimal time solution to the firing squad synchronization problem. *Information and Control*, 10(1):22–42, January 1967.
- [2] John J. Grefenstette. Network structure and the firing squad synchronization problem. *Journal of Computer and System Sciences*, 26(1):139–152, February 1983.
- [3] Gruska, La Torre, and Parente. Optimal time and communication solutions of firing squad synchronization problems on square arrays, toruses and rings. In *International Conference on Developments in Language Theory (DLT), LNCS*, volume 8, 2004.
- [4] Kojiro Kobayashi. The firing squad synchronization problem for two-dimensional arrays. *Information and Control*, 34(3):177–197, July 1977.
- [5] Jacques Mazoyer. A six-state minimal time solution to the firing squad synchronization problem. *Theoretical Computer Science*, 50(2):183–238, 1987.
- [6] Jacques Mazoyer. On optimal solutions to the firing squad synchronization problem. *Theoretical Computer Science*, 168(2):367–404, 20 November 1996.
- [7] F. R. Moore and G. G. Langdon. A generalized firing squad problem. *Information and Control*, 12(3):212–220, March 1968.

- [8] H. B. Nguyen and V. C. Hamacher. Pattern synchronization in two-dimensional cellular spaces. *Information and Control*, 26(1):12–23, September 1974.
- [9] Yasuaki Nishitani and Namio Honda. The Firing Squad Synchronization Problem for graphs. *Theoretical Computer Science*, 14(1):39–61, April 1981.
- [10] Ilka Shinahr. Two- and three-dimensional firing-squad synchronization problem. *Information and Control*, 24:163–180, 1974.
- [11] Abraham Waksman. An optimum solution to the firing squad synchronization problem. *Information and Control*, 9(1):66–78, February 1966.
- [12] Dušan Závodský. Systolické výpočty. Diplomová práca, Matematicko-fyzikálna fakulta, Univerzita Komenského, Bratislava, 1997.