



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Petra Vaňková

Bubble Blast 2

Katedra algebry

Vedoucí bakalářské práce: Doc. RNDr. Jiří Tůma, DrSc.

Studijní program: Matematika

Studijní obor: Matematické metody informační bezpečnosti

Praha 2018

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Děkuji Doc. RNDr. Jiřímu Tůmovi, DrSc. za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval.

Název práce: Bubble Blast 2

Autor: Petra Vaňková

Katedra: Katedra algebry

Vedoucí bakalářské práce: Doc. RNDr. Jiří Tůma, DrSc., Katedra algebry

Abstrakt: Tato práce se zabývá matematickou analýzou a řešitelností hry Bubble Blast 2. První část přibližuje pravidla hry a zavádí maticový zápis hrací plochy. Druhá část práce nejprve popisuje dynamiku dvojrozměrné verze a definuje důležité pojmy s ní související jako agent, čas nebo stav hry. Dále je vysvětleno, proč je obtížné se zabývat řešitelností dvourozměrné verze a je zavedena zjednodušená varianta hry na přímce. Hlavní částí je několik tvrzení o jednorozměrné verzi hry, která nakonec vedou k nutné a postačující podmínce řešitelnosti hry na přímce jedním kliknutím. Všechny výsledky práce jsou původní a ve velmi malé míře se odvoláváme na zdrojový kód hry.

Klíčová slova: model hry, stavová matice, transformace stavu, agent

Title: Bubble Blast 2

Author: Petra Vaňková

Department: Department of Algebra

Supervisor: Doc. RNDr. Jiří Tůma, DrSc., Department of Algebra

Abstract: This thesis deals with mathematical analysis and solvability of the Bubble Blast 2 game. The first part introduces rules of the game, and a matrix representation of the game. The second part at first describes the two-dimensional game dynamics, and also important terms such as agent, time, and state matrix are defined. It is explained why dealing with solvability of the two-dimensional game is difficult and an easier straight line version of the game is introduced. The main part consists of several theorems about the one-dimensional game that eventually lead to the necessary and sufficient condition of the game solvability with only one click given. All results of this thesis are original, only a minor part is based on the game source code.

Keywords: model of the game, state matrix, state transformation, agent

Obsah

Úvod	2
1 Dynamika jednoho kliknutí	4
1.1 Hra na přímce	7
1.1.1 Hra na přímce s jedním možným kliknutím	9
1.1.2 Rozšíření řešitelnosti pozice	15
1.1.3 Čas eliminace	20
Závěr	21

Úvod

Bubble Blast 2 je logická mobilní hra pro jednoho hráče z dílny studia Magma Mobile. V současné době existují verze pro operační systémy Android, iOS a Windows. Hra má několik tisíc úrovní rozdělených do packů, tedy balíčků. Konkrétně pro operační systém iOS je to 70 balíčků po 100 úrovních.

Každou úroveň tvoří obdélníková hrací plocha o 6 řádcích a 5 sloupcích. Dvojici (i, j) budeme značit konkrétní *místo* na i -tém řádku a j -tém sloupci hrací plochy. Na každém místě hrací plochy se může nacházet bublina určité barvy, a to buď červené, zelené, žluté, nebo modré.

Na začátku každé úrovně má hráč k dispozici předem daný počet kliků. Cílem hráče je pomocí tohoto počtu kliknutí získat prázdnou hrací plochu bez bublin. Každé kliknutí na bublinu způsobí změnu její barvy podle následujícího schématu:

modrá → žlutá → zelená → červená → bezbarvá

Změna barvy je možná pouze ve směru šipek a jedním kliknutím právě o jeden krok.

Kliknutím na červenou bublinu se již nezmění její barva, nýbrž bublina úplně zmizí, resp. vybuchne. Toto „vybuchnutí“ vysvětlíme o něco později.

Podle zmíněného pravidla změny barev můžeme bublinám přiřadit číselné hodnoty. Půjdeme-li ve směru šipek, poslední změnou je zmizení červené bubliny. Budeme tak přirozeně uvažovat, že místo na hrací ploše bez bublin má hodnotu 0. Kliknutí pak budeme chápat jako snižování hodnoty bublin. Barvám tedy podle schématu můžeme přiřadit sestupně hodnoty:

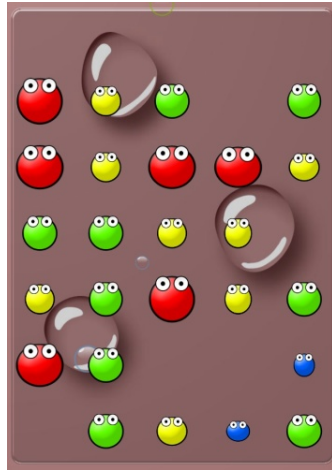
modrá	žlutá	zelená	červená	bezbarvá
4	3	2	1	0

Konkrétní *pozici* hrací plochy budeme nadále zapisovat maticí

$$A = (a_{ij})_{6 \times 5},$$

kde a_{ij} je barva bubliny na místě (i, j) , případně informace, že na daném místě matice se žádná bublina nenachází, pokud $a_{ij} = 0$.

Na obrázku 1a je zachycena úroveň 61 z prvního balíčku v uživatelském prostředí, tedy s barevnými bublinami, matice na obrázku 1b je pak příslušným maticovým zápisem pozice v této úrovni.



(a) Uživatelské prostředí

$$\begin{bmatrix} 1 & 3 & 2 & 0 & 2 \\ 1 & 3 & 1 & 1 & 3 \\ 2 & 2 & 3 & 3 & 0 \\ 3 & 2 & 1 & 3 & 2 \\ 1 & 2 & 0 & 0 & 4 \\ 0 & 2 & 3 & 4 & 2 \end{bmatrix}$$

(b) Maticový zápis pozice

Obrázek 1: Hrací plocha

Jinými slovy je cílem hráče vynulovat matici A . Nyní se již dostáváme k „vybouchnutí“ červené bubliny.

Kromě toho, že bublina zmizí, vytvoří se zároveň čtyři nové prvky. Kliknutí tímto vnáší do hry dynamiku, neboť tyto nové prvky se narozdíl od barevných bublin pohybují. Všechny čtyři se pohybují stejnou konstantní rychlostí, dva horizontálně po řádku, jeden doleva, druhý doprava a dva vertikálně po sloupci, jeden nahoru a druhý dolů.

Těmto pohybujícím se prvkům budeme říkat *agenti*. Agent se pohybuje po hrací ploše, dokud nenarazí na nenulový prvek. Pokud se tak stane, chová se jako kliknutí. Podle barvy bubliny tak dojde buď ke změně její barvy, nebo v případě červené bubliny k jejímu „vybouchnutí“ a vzniku nových agentů. Agent, který narazí do nějaké bubliny, zároveň v tu chvíli zmizí.

Jsou tedy právě dva způsoby, jak snížit hodnotu prvku a_{ij} . Prvním z nich je kliknutí, druhým pak náraz agenta. Nárazy agentů způsobují řetězové reakce, kdy agent, který vznikl vybouchnutím jedné bubliny, způsobí vybouchnutí další bubliny na hrací ploše atd.

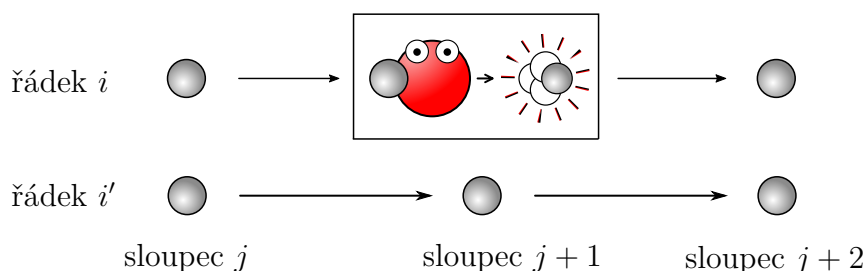
Definice 1. Řekneme, že hrací plocha (nebo její část) je řešitelná m kliky, pokud ji lze maximálně m kliky vynulovat.

Obtížnost hry je dána malým počtem kliknutí, které má hráč k dispozici, oproti součtu hodnot bublin na hrací ploše. Úkolem je tak kliknout na bubliny, jejichž smazáním dojde ke spuštění takových řetězových reakcí, které povedou k vynulování celé hrací plochy.

1. Dynamika jednoho kliknutí

Nadále budeme předpokládat, že hráč v průběhu hry vždy klikne a počká, dokud neskončí všechny řetězové reakce způsobené agenty, a teprve potom klikne znovu. Dynamiku hry tak v následujícím popisu budeme chápat jako dynamiku průběhu hry po jednom kliknutí.

Z pohybu agentů nyní odvodíme důležitou veličinu, a to *čas*. Bude se nám hodit nejen k získání informací, kdy došlo, popřípadě dojde k vynulování nějakého prvku hrací plochy, ale také k definici dalších pojmů.



Obrázek 1.1: Problematika času

Uvažujme dva agenty na místech (i,j) a (i',j) , tedy ve stejném sloupci, pohybující se oba směrem doprava jako na obrázku 1.1. Na místě $(i,j+1)$ uvažujme červenou bublinu, na místě $(i',j+1)$ žádnou. Agent pohybující se po řádku i tedy způsobí eliminaci červené bubliny, agent na řádku i' pouze pokračuje dál doprava. Pozorováním průběhu hry jsme zjistili, že náraz agenta do bubliny, její výbuch a vznik nových agentů nemá žádnou časovou náročnost a všechny tyto události proběhnou v jeden moment. Oba agenti tak dorazí na příslušné místo hrací plochy ve sloupci $j+2$ ve stejnou dobu.

Chápání času tedy můžeme omezit na pohyb agentů po prázdné hrací ploše. K tomu budeme potřebovat ještě následující pojmy.

Definice 2. Mějme dvě místa na hrací ploše $(a_1,a_2), (b_1,b_2)$, pak vzdáleností těchto dvou míst rozumíme číslo $|a_1 - b_1| + |a_2 - b_2|$.

Sousedními prvky prvku na místě (i,j) budeme nazývat prvky ve vzdálenosti 1 od (i,j) .

Konečně *čas* budeme chápat jako diskrétní veličinu a jeho jednotkou bude právě doba, za kterou agent dojde na místo sousedního prvku, tzn. je-li agent pohybující se horizontálně v čase t na místě (i,j) , v čase $t+1$ se bude nacházet na místě $(i,j-1)$, nebo $(i,j+1)$ podle konkrétního směru.

Definice 3. Stav hry v čase t pak popíšeme dvojicí matic - stavovou maticí v čase t a maticí aktivních agentů v čase t .

Stavovou maticí hry v čase t rozumíme matici $\mathbf{a}^t = (a_{ij}^t)_{6 \times 5}$, kde a_{ij}^t je barva místa (i,j) v čase t .

Konečný stav pak budeme značit (\mathbf{e}, \mathbf{E}) .

Nenulový konečný stav je tedy zároveň počátečním stavem pro další kliknutí.

U agentů nás bude zpravidla zajímat jejich počet, popřípadě poloha v nějakém čase t . Uvažujme agenta, který vznikl vybuchnutím nějaké bubliny v čase $t' < t$ a který již do žádné bubliny hrací plochy nenarazí. Takovýto agent během hry určitě vznikne, ať už vybuchnutím bubliny na kraji hrací plochy, nebo způsobí-li vybuchnutí bublin v jeho směru jiní agenti. Agent se pak od určitého časového okamžiku může začít vzdalovat od hrací plochy. Bude-li nás pak zajímat například počet agentů v čase, kdy došlo k eliminaci celé hrací plochy, můžou se agenti nacházet jak na hrací ploše, tak v různé vzdálenosti od ní. Jelikož před hrou neumíme určit, jak dlouho bude eliminace celé hrací plochy trvat, budeme obecně uvažovat nekonečnou matici.

Definice 4. Maticí aktivních agentů v čase t budeme nazývat nekonečnou matici

$$\mathbf{A}^t = (A_{ij}^t)_{\mathbb{Z} \times \mathbb{Z}},$$

kde $A_{ij}^t = c_{ij}^t L + d_{ij}^t R + e_{ij}^t U + f_{ij}^t D$, kde $c_{ij}^t, d_{ij}^t, e_{ij}^t, f_{ij}^t \geq 0$.

Čísla $c_{ij}^t, d_{ij}^t, e_{ij}^t$ a f_{ij}^t v tomto pořadí vyjadřují počet agentů v čase t na místě (i, j) mířících doleva, doprava, nahoru a dolů. Jelikož náraz agenta do bubliny, její vybuchnutí a vznik nových agentů proběhne v jednom čase t , budeme předpokládat, že aktivní agenti v čase t jsou agenti vzniklí v čase t a agenti, kteří se pohybují beze změny. U agentů, kteří v čase t narazili do nějaké bubliny, tak v čase t budeme uvažovat, že již neexistují.

Definice 5. Necht (\mathbf{e}, \mathbf{E}) je konečný stav po kliknutí. Řekneme, že klik eliminuje prvek a_{ij} , pokud $a_{ij} > e_{ij} = 0$.

Dynamika hry začíná kliknutím na určité místo hrací plochy, čímž se hra dostane do stavu $(\mathbf{a}^0, \mathbf{A}^0)$.

Klikne-li hráč na místo (i, j) a je-li $a_{ij}^0 \geq 1$, pak $A_{ij}^0 = 0$. Je-li $a_{ij}^0 = 0$, pak $A_{ij}^0 = L + R + U + D$, neboť eliminací prvku a_{ij} vzniknou čtyři nové agenti.

Přechod $(\mathbf{a}^t, \mathbf{A}^t) \mapsto (\mathbf{a}^{t+1}, \mathbf{A}^{t+1})$ popisuje několik možných situací a jejich efekt na jednotlivé prvky obou matic.

1. Pohyb agentů na místo (i, j) bez nárazu do bubliny:

\mathbf{a}^{t+1} : Hodnota prvků stavové matice se nemění: $a_{ij}^{t+1} = a_{ij}^t = 0$.

\mathbf{A}^{t+1} : V matici aktivních agentů dojde k posunu agentů ze sousedních prvků v příslušném směru:

$$A_{ij}^{t+1} = c_{i,j+1}^t L + d_{i,j-1}^t R + e_{i+1,j}^t U + f_{i-1,j}^t D.$$

Tento vztah platí pro všechna $i, j \in \mathbb{Z}$. Mimo hrací plochu se již žádné bubliny nenacházejí, proto tato situace jako jediná může nastat v matici aktivních agentů i mimo rozměry hrací plochy.

2. Efekt kliknutí a nárazu jednoho agenta do bubliny na místě (i, j) v čase $t + 1$:

\mathbf{a}^{t+1} : Pro prvky stavové matice platí: $a_{ij}^{t+1} = a_{ij}^t - 1$.

\mathbf{A}^{t+1} : Je-li $a_{ij}^{t+1} \geq 1$, pak $A_{ij}^{t+1} = 0$.

Je-li $a_{ij}^{t+1} = 0$, pak $A_{ij}^{t+1} = L + R + U + D$, neboť $c_{ij}^{t+1} = d_{ij}^{t+1} = e_{ij}^{t+1} = f_{ij}^{t+1} = 1$.

3. Efekt nárazu více agentů do bubliny na místě (i,j) v čase $t + 1$:
Rozlišíme dva případy, z nichž pouze první je jednoduchý na popis:

(a) Je-li $c_{i,j+1}^t + d_{i,j-1}^t + e_{i+1,j}^t + f_{i-1,j}^t \leq a_{i,j}^t$:

$$\mathbf{a}^{t+1}: a_{ij}^{t+1} = a_{ij}^t - (c_{i,j+1}^t + d_{i,j-1}^t + e_{i+1,j}^t + f_{i-1,j}^t).$$

\mathbf{A}^{t+1} : Pro matici aktivních agentů platí, že je-li $a_{ij}^{t+1} \geq 1$, pak $A_{ij}^{t+1} = 0$. Je-li $a_{ij}^{t+1} = 0$, pak $A_{ij}^{t+1} = L + R + U + D$.

(b) Je-li $c_{i,j+1}^t + d_{i,j-1}^t + e_{i+1,j}^t + f_{i-1,j}^t > a_{i,j}^t$:

$$\mathbf{a}^{t+1}: a_{ij}^{t+1} = 0.$$

\mathbf{A}^{t+1} : Pohybují-li se všichni agenti ve stejném směru, například doleva, pak $\mathbf{A}^{t+1} = (c_{i,j+1}^t - a_{i,j}^t + 1)L$.

Přichází-li ale agenti alespoň ze dvou směrů, potřebujeme pro další vývoj hry vědět, který z agentů bude mít v eliminaci přednost a který bude pokračovat dál.

Ze zdrojového kódu verze pro operační systém Android jsme zjistili, že nelze jednoznačně říci, který agent způsobí vybuchnutí bubliny a kteří budou případně pokračovat dále. Vývojáři pro ukládání agentů využili seznam, který se aktualizuje v každém stavu hry. Agent je v seznamu popsán pomocí tří atributů, a to rádkového indexu, sloupcového indexu a směru. Atribut *směr* nabývá hodnot 1 až 4, kde každé číslo odpovídá jednomu směru, a pomocné hodnoty 0 například pro agenty, kteří zanikli. Algoritmus 1 popisuje průběh hry a aktualizaci seznamu s jedním klikem.

Algoritmus 1 Android

Vstup: místo kliknutí (klikáme na prvek hodnoty větší než 0)

Výstup: stavová matice e v konečném stavu po jednom kliknutí

- 1: vytvoříme agenta s indexy místa kliknutí a libovolným směrem;
 - 2: hodnotu prvku, na který klikneme, snížíme o 1 a nastavíme směr na 0;
 - 3: **if** nová hodnota prvku = 0 **then**
 - 4: vytvoříme 4 nové agenty v pořadí směrů doleva, nahoru, doprava a dolů a vložíme je na konec seznamu;
 - 5: **while** seznam agentů je neprázdný **do**
 - 6: **for** všechny agenty v seznamu **do**
 - 7: smažeme každý mimo hrací plochu;
 - 8: **for** všechny agenty v seznamu **do**
 - 9: **if** hodnota prvku matice na místě agenta je > 0 **then**
 - 10: snížíme její hodnotu o 1;
 - 11: nastavíme směr agenta na 0;
 - 12: **if** nová hodnota prvku = 0 **then**
 - 13: vytvoříme 4 nové agenty v pořadí směrů doleva, nahoru, doprava a dolů a vložíme je na konec seznamu;
 - 14: **else if** hodnota prvku = 0 **then**
 - 15: posuneme agenta v příslušném směru;
-

-
- 16: **for** všechny agenty v seznamu **do**
17: smažeme všechny, jejichž směr je 0;
-

Pořadí agentů je tedy v seznamu kromě času dáno dvojím uspořádáním. První je uspořádání agentů podle jejich vzniku. Uvažujme agenty, kteří vznikli v čase t eliminací nějaké bubliny (agentovi, který způsobil její vybuchnutí, budeme říkat „rodič“, nově vzniklým agentům pak jeho „potomci“). Noví agenti jsou pak v seznamu uspořádání podle pořadí svých rodičů. Je-li jeden agent v seznamu před jiným agentem, pak potomci prvního agenta budou v seznamu před potomky druhého agenta, i když všichni vznikli v jeden čas t .

Druhé je samotné uspořádání potomků nějakého agenta. Do seznamu jsou vloženi podle směrů, kterými se pohybují, a to vždy v tomto pořadí - doleva, nahoru, doprava a dolů.

Je-li tedy počet agentů, kteří se setkají na místě (i,j) , větší než hodnota a_{ij} a přicházejí-li z více směrů, nemůžeme z celkového uspořádání seznamu jednoznačně určit pořadí agentů, v jakém budou snižovat hodnotu a_{ij} .

1.1 Hra na přímce

Z popisu hry s dvojrozměrnou hrací plochou je patrné, že dynamika je poměrně složitá na popis a mohou dokonce nastat situace, které popsat neumíme. Proto zavedeme variantu, ve kterém se omezíme na jednorozměrnou hrací plochu, a to jeden řádek s obecnou délkou. Pohyb agentů pak budeme uvažovat pouze v horizontálním směru. Budeme tedy předpokládat, že všichni agenti, kteří se ve hře objeví, vznikli eliminací prvků tohoto řádku. Tuto variantu by šlo také využít na analýzu dvojrozměrné hry, kdy v určité fázi hry se bubliny nacházejí pouze v jednom řádku, popřípadě sloupci.

Uvažujme tedy hrací plochu jako nekonečnou přímku, která obsahuje pouze konečně mnoho nenulových prvků. Tuto pozici hrací plochy pak snadno popíšeme vektorem obsahujícím všechny nenulové prvky.

Každá hra začíná kliknutím, prvek na který klikneme, označíme a_0 . Nejmenší index nenulové složky označíme $-M$, největší pak N .

$$\mathbf{a} = (a_{-M} \ \cdots \ a_0 \ \cdots \ a_N).$$

Úsekem vektoru $\mathbf{a} = (a_{-M}, \dots, a_N)$ délky k rozumíme složky $(a_{i+1}, \dots, a_{i+k})$ takové, že $-M \leq i+1$ a $i+k \leq N$. Nalezneme nejdelší úsek vektoru \mathbf{a} obsahující a_0 takový, že složky úseku nabývají pouze hodnot 0, nebo 1. Pak nejmenší index tohoto úseku označíme $-l$, největší pak r , tedy

$$\mathbf{a} = (a_{-M} \ \cdots \ a_{-l} \ \cdots \ a_0 \ \cdots \ a_r \ \cdots \ a_N).$$

Aby kliknutím došlo ke vzniku agentů, musíme předpokládat, že $a_0 = 1$.

Definice 6. Stavem hry v čase t v této variantě hry rozumíme dvojici vektorů $(\mathbf{a}^t, \mathbf{A}^t)$ - stavový vektor v čase t a vektor aktivních agentů v čase t .

Stavovým vektorem hry v čase t rozumíme vektor $\mathbf{a}^t = (a_{-M}^t, \dots, a_N^t)$, kde a_i^t je hodnota barvy bubliny na místě i v čase t .

Vektorem aktivních agentů v čase t rozumíme nekonečný vektor

$$\mathbf{A}^t = (\dots \ A_j^t \ \dots),$$

kde pro $\forall j \in \mathbb{Z}$ je $A_j^t = c_j^t L + d_j^t R$ a $c_j^t, d_j^t \geq 0$.

Čísla c_j^t , resp. d_j^t vyjadřují počet aktivních agentů v čase t na indexu j mířících doleva, resp. doprava.

Definice 7. Počátečním stavem budeme nazývat dvojici $(\mathbf{a}^0, \mathbf{A}^0)$, tedy stavový vektor a vektor aktivních agentů v čase 0.

Konečným stavem budeme rozumět dvojici $(\mathbf{a}^T, \mathbf{A}^T)$, pokud T je nejmenší takové, že $\mathbf{a}^t = \mathbf{a}^T$ pro každé $t \geq T$. V takovém případě budeme konečný stav značit (\mathbf{e}, \mathbf{E}) .

Definice 8. Pro $t > 0$ definujme interval $I^t = \{i | a_i^t = 0\}$.

Definice 9. Řekneme, že pozice \mathbf{a} se eliminuje kliknutím na a_0 , pokud pro konečný stav (\mathbf{e}, \mathbf{E}) z počátečního stavu $(\mathbf{a}^0, \mathbf{A}^0)$ platí $\mathbf{e} = \mathbf{o}$.

Úsek (a_{-m}, \dots, a_n) se eliminuje kliknutím na a_0 , pokud $[-m, n] = I^t$ pro nějaké $t \geq 0$.

Druhá část definice je rozšířením první části, neboť pozice \mathbf{a} se eliminuje právě tehdy, když $I^t = [-M, N]$.

Příklad 1. Uvažujme-li pozici

$$\begin{pmatrix} 1 & 1 & 1 & 3 \\ -2 & -1 & 0 & 1 \end{pmatrix},$$

můžeme říci, že kliknutím na a_0 se eliminuje. Vezmeme-li ale úsek této pozice

$$\begin{pmatrix} 1 & 1 & 3 \\ -1 & 0 & 1 \end{pmatrix},$$

kliknutím na a_0 se neeliminuje $a_1 = 3$, neboť prvek a_{-2} v eliminaci úseku neuvažujeme.

Dynamika hry na přímce, tedy přechod $(\mathbf{a}^t, \mathbf{A}^t) \mapsto (\mathbf{a}^{t+1}, \mathbf{A}^{t+1})$, kde \mathbf{a}^t a \mathbf{A}^t jsou vektory, vychází z dvojrozměrné verze s tím rozdílem, že uvažujeme pouze agenty pohybující se vodorovně a místo matice typu 6×5 uvažujeme nekonečnou přímku s konečně mnoha nenulovými prvky.

Problém, kdy v dvojrozměrné verzi hry není možné určit pořadí agentů v seznamu, se zúží na rozhodnutí, zda mají přednost agenti zleva, nebo zprava. Pro jednoduchost zvolíme, že v seznamu budou vždy agenti mířící doprava před agenty mířícími doleva. Tedy pokud $c_{i-1}^t + d_{i+1}^t > a_i^t$, pak platí následující:

Pokud $c_{i-1}^t - a_i^t \geq 0$, pak $c_i^{t+1} = c_{i-1}^t - a_i^t + 1$ a $d_i^{t+1} = d_{i+1}^t + 1$. Pokud $c_{i-1}^t - a_i^t < 0$, pak $c_i^{t+1} = 1$ a $d_i^{t+1} = a_i^t - c_{i-1}^t - d_{i+1}^t + 1$.

Příklad 2. Je-li pro nějaké i $a_i^t = 2$ a $c_{i-1}^t = 2$, $d_{i+1}^t = 1$. Pak $a_i^{t+1} = 0$ a $A_i^{t+1} = 2L + R$.

Tuto volbu ale ve variantě na přímce nebudeme potřebovat, neboť využijeme tvrzení uvedeného níže.

1.1.1 Hra na přímce s jedním možným kliknutím

V následující části tedy budeme uvažovat, že máme k dispozici pouze jedno kliknutí. Dokážeme, že při hře na přímce s jedním klikem se nemusíme zabývat problémem eliminace složky více agenty zároveň. Zaměříme se tak hlavně na podmínky řešitelnosti hrací plochy a následně z toho plynoucí vlastnosti jako například čas eliminace jednotlivých složek.

Hra bez závislosti na čase

UVědomíme si, že nulové složky vektoru mají sice vliv na výsledný čas eliminace, ale ne na to, jaké složky budou eliminovány. V této části se budeme zabývat pouze řešitelností, proto můžeme nadále předpokládat, že $a_i > 0$ pro všechna $i \in [-M, N]$.

Tvrzení 1. *Indexy nulových prvků v \mathbf{a}^t tvoří interval I^t obsahující 0. Pokud $A_i^t \neq 0$ pro $i \in [-M, N]$, pak $i \in I^t$ a $c_i^t, d_i^t \leq 1$.*

Důkaz. Dokážeme indukci podle t .

Jediný nulový prvek v \mathbf{a}^0 je a_0^0 , z definice počátečního stavu hry plyne, že $A_0^0 = L + R$ a $A_i^0 = 0$ pro všechna $i \neq 0$. Tvrzení tak platí v čase $t = 0$.

Předpokládejme, že tvrzení platí pro $t \geq 0$. Necht indexy nulových prvků v \mathbf{a}^t tvoří interval $I^t = [-m, n]$ a pro všechna $A_i^t \neq 0$, kde $i \in [-M, N]$ platí zároveň $i \in [-m, n]$.

Předpokládejme, že pro nějaké i platí $0 = a_i^{t+1} \neq a_i^t$. Pak zřejmě $i \notin I^t$. Bez újmy na obecnosti předpokládejme $i \leq -(m+1)$. Protože $a_i^{t+1} = 0$, pak musí platit $0 < a_i^t \leq c_{i+1}^t + d_{i-1}^t$. Z indukčního předpokladu víme, že se aktivní agenti nacházejí pouze v I^t , tedy $d_{i-1}^t = 0$. Tedy platí $0 < a_i^t \leq c_{i+1}^t$, z čehož plyne, že $i+1 \in I^t = [-m, n]$ a protože všechny prvky s indexy v I^t jsou nulové, tak $i+1 = -m$ a $i = -(m+1)$. Podobně $i = n+1$ pro $i \geq n+1$. V čase $t+1$ se tedy může interval rozšířit pouze o $-m-1$ nebo $n+1$ a platí $0 \in [-m-1, n+1]$.

Obdobně uvažujme, že $A_i^{t+1} > A_i^t = 0$. Právě jsme ukázali, že noví agenti v čase $t+1$ vzniknou pouze eliminací a_{-m-1} nebo a_{n+1} . Z indukčního předpokladu víme, že $c_i^t, d_i^t \leq 1$ pro $i \in I^t$, tedy do a_{-m-1} i a_{n+1} může v čase $t+1$ narazit pouze jeden agent, který následně zmizí. Případnou eliminací a_{-m-1} a a_{n+1} tak máme $A_{-m-1}^{t+1} = L + R$ a $A_{n+1}^{t+1} = L + R$. Zároveň agenti, kteří do žádného prvku nenarazí, z indukčního předpokladu zůstanou v I^t , tedy pro všechna $i \in [-M, N]$ platí, že pokud $A_i^{t+1} \neq 0$, pak $i \in I^{t+1}$ a $c_{-m-1}^t, d_{-m-1}^t \leq 1$ a $c_{n+1}^t, d_{n+1}^t \leq 1$. □

Důsledek 1. Pro každé $t \geq 0$ platí $I^t \subseteq I^{t+1}$.

Důsledek 2. Do prvku hrací plochy v čase $t > 0$ s indexem $i > 0$ může narazit agent pouze zleva, do prvku s indexem $j < 0$ pak pouze zprava.

Značení. Z tvrzení 1 víme, že pro $t > 0$ platí $c_i^t, d_i^t \leq 1$ pro všechna $i \in I^t$. Pro samotné označení agenta mířícího doleva (resp. doprava) budeme používat označení c_i^t (resp. d_i^t).

Definice 10. *Řekneme, že se agent spotřebuje v čase t na místě i , narazí-li na nenulovou složku, tedy je-li $c_{i+1}^{t-1} \neq 0$ nebo $d_{i-1}^{t-1} \neq 0$ a $a_i^{t-1} > 0$.*

Definice 11. Předpokládejme, že se pozice $\mathbf{a} = (a_{-M}, \dots, a_0, \dots, a_N)$ klikem na a_0 celá eliminuje. Necht t je čas, kdy $\mathbf{a}^t = \mathbf{e} = \mathbf{o}$. Pak o aktivních agentech v čase t řekneme, že je pozice \mathbf{a} vypustí.

Navíc řekneme, že pozice vypustí doleva agenty c_i^t a doprava agenty d_i^t .

Z definice vektoru aktivních agentů snadno nahlédneme, že celkový počet agentů, které pozice vypustí, je právě

$$\sum_{i \in \mathbb{Z}} (c_i^t + d_i^t). \quad (1.1)$$

Pouze doleva pak vypustí $\sum_{i \in \mathbb{Z}} c_i^t$ agentů a doprava $\sum_{i \in \mathbb{Z}} d_i^t$.

Tyto hodnoty lze vyjádřit i bez použití vektoru aktivních agentů:

Tvrzení 2. Mějme vektor $\mathbf{a} = (a_{-M}, \dots, a_0, \dots, a_N)$, který se celý eliminuje v čase t . Pak

$$\sum_{i \in \mathbb{Z}} c_i^t = M + N + 1 - \sum_{j=-1}^{-M} a_j \geq 1, \quad (1.2)$$

$$\sum_{i \in \mathbb{Z}} d_i^t = M + N + 1 - \sum_{j=1}^N a_j \geq 1, \quad (1.3)$$

kde alespoň jedna nerovnost je ostrá a dále

$$\sum_{i \in \mathbb{Z}} (c_i^t + d_i^t) = 2(M + N + 1) - \sum_{j=-M}^N a_j + 1 \geq 3. \quad (1.4)$$

Důkaz. Ze vztahů pro jednotlivé směry stačí dokázat 1.2, vztah 1.3 se dokáže analogicky.

Eliminací všech složek vektoru vznikne právě $M + N + 1$ agentů pohybujících se doleva. Z důsledku 2 plyne, že z tohoto počtu se jich právě $\sum_{j=-1}^{-M} a_j$ spotřebuje, neboť do složek se záporným indexem mohou narazit pouze agenti mířící doleva. Zřejmě agent $c_{-M}^{t'}$, který vznikl eliminací a_{-M} v čase t' , na žádnou bublinu nenarazí, a tedy vektor vypustí doleva vždy alespoň tohoto agenta. Doprava pak vypustí alespoň agenta $d_N^{t''}$, který vznikl eliminací a_N v čase t'' .

Zároveň z důsledku 2 plyne, že a_{-M} nebo a_N bude eliminováno jako poslední prvek hrací plochy. Pak se zřejmě nevyužije ani jeden agent, který vznikl eliminací tohoto prvku, a tedy alespoň jedna z nerovností musí být ostrá. Součtem vztahů 1.2 a 1.3 dostaneme $\sum_{i \in \mathbb{Z}} (c_i^t + d_i^t) = 2(M + N + 1) - \sum_{j=-M}^N a_j + 1$. Z ostrosti alespoň jednoho ze vztahů 1.2 a 1.3 pak získáme nerovnost v 1.4. □

V průběhu hry se může stát, že budeme potřebovat právě uvedené vztahy pro kratší úsek vektoru \mathbf{a} . Jednoduše pak můžeme použít definici 11 a tvrzení 2 pouze pro eliminovaný úsek.

Značení. Počet agentů, které úsek $(a_{-m}, \dots, a_0, \dots, a_n)$ vypustí doprava, resp. doleva označíme $R_{(-m,n)}$, resp. $L_{(-m,n)}$.

Tvrzení 3. Z tvrzení 2 plyne, že úsek samých jedniček $(a_{-l}, \dots, a_0, \dots, a_r)$ vektoru \mathbf{a} vypustí právě $r + l + 2$ agentů a platí

$$L_{(-l,r)} = r + 1, \quad R_{(-l,r)} = l + 1. \quad (1.5)$$

Předpokládejme nyní, že $M > m$ nebo $N > n$ a klikem na a_0 lze eliminovat celý úsek $(a_{-m}, \dots, a_0, \dots, a_n)$. Prvky a_{-m-1} a a_{n+1} se mohou eliminovat pouze pomocí agentů, které vypustí úsek $(a_{-m}, \dots, a_0, \dots, a_n)$.

Tvrzení 4. Je dán vektor \mathbf{a} a necht $(a_{-m}, \dots, a_0, \dots, a_n)$ je jeho podúsek, který se celý eliminuje klikem na a_0 .

Je-li $n < N$, pak se úsek (a_{-m}, \dots, a_{n+1}) celý eliminuje právě tehdy, když je splněna podmínka $a_{n+1} \leq R_{(-m,n)}$ a pak platí

$$R_{(-m,n+1)} = R_{(-m,n)} - a_{n+1} + 1, \quad L_{(-m,n+1)} = L_{(-m,n)} + 1 \quad (1.6)$$

Podobně je-li $m < M$, pak se eliminuje celý úsek (a_{-m-1}, \dots, a_n) právě tehdy, když je splněna podmínka $a_{-m-1} \leq L_{(-m,n)}$ a pak platí

$$L_{(-m-1,n)} = L_{(-m,n)} - a_{-m-1} + 1, \quad R_{(-m-1,n)} = R_{(-m,n)} + 1 \quad (1.7)$$

Důkaz. Stačí dokázat pro rozšíření eliminovaného úseku o prvek a_{n+1} vpravo, rozšíření o prvek vlevo se dokáže analogicky.

Předpokládejme, že $a_{n+1} \leq R_{(-m,n)}$. Z předpokladu, že se úsek (a_{-m}, \dots, a_n) eliminuje a z tvrzení 2 máme $R_{(-m,n)} = m + n + 1 - \sum_{j=1}^n a_j$. Po úpravě dostaneme

$0 \leq m + n + 1 - \sum_{j=1}^{n+1} a_j$. Zřejmě tedy $m + n + 1$ je dostatečný počet agentů k eliminaci úseku (a_{-m}, \dots, a_{n+1}) .

Dokážeme opačnou implikaci, tedy eliminuje-li se úsek (a_{-m}, \dots, a_{n+1}) , pak platí $a_{n+1} \leq R_{(-m,n)}$. Víme, že úsek (a_{-m}, \dots, a_n) se celý eliminuje, a tedy $R_{(-m,n)} = \sum_{i \in \mathbb{Z}} c_i^t = m + n + 1 - \sum_{j=1}^n a_j \geq 1$. Z předpokladu, že se eliminuje i úsek (a_{-m}, \dots, a_{n+1}) , pak máme

$$1 \leq R_{(-m,n+1)} = m + (n + 1) + 1 - \sum_{j=1}^{n+1} a_j = 1 + R_{(-m,n)} - a_{n+1},$$

po úpravě tedy $a_{n+1} \leq R_{(-m,n)}$.

Takže právě a_{n+1} agentů je spotřebováno eliminací složky a_{n+1} a vznikne právě jeden nový agent směřující doprava. Tedy máme $R_{(-m,n+1)} = R_{(-m,n)} - a_{n+1} + 1$.

Zároveň tím také vznikne jeden nový agent směřující doleva, a tedy $L_{(m,n+1)}^j = L_{(m,n)}^j + 1$. □

Definice 12. Řekneme, že eliminovaný úsek $(a_{-m}, \dots, a_0, \dots, a_n)$ je maximální, pokud $[-m,n] = I^T$.

Z tvrzení 4 okamžitě plyne:

Tvrzení 5. Úsek $(a_{-m}, \dots, a_0, \dots, a_n)$ není maximální právě tehdy, když platí buď

$$m < M \quad a \quad a_{-m-1} \leq L_{(-m,n)} \quad (1.8)$$

nebo

$$n < N \quad a \quad a_{n+1} > R_{(-m,n)}. \quad (1.9)$$

Důsledky.

1. Je-li splněna alespoň jedna z podmínek $a_{n+1} \leq R_{(-m,n)}$ a $a_{-m-1} \leq L_{(-m,n)}$, pak eliminovaný úsek (a_{-m}, \dots, a_n) není maximální.
2. Necht $(a_{-m}, \dots, a_0, \dots, a_n)$, $n < N$ je maximální úsek, který lze eliminovat kliknutím na a_0 . Pak

$$\begin{aligned} e_{n+1} &= a_{n+1} - R_{(-m,n)}, \\ e_{n'} &= a_{n'}, n + 1 < n' \leq N. \end{aligned} \quad (1.10)$$

Analogicky pro $m < M$ je

$$\begin{aligned} e_{-m-1} &= a_{-m-1} - L_{(-m,n)}, \\ e_{-m'} &= a_{-m'}, m + 1 < m' \leq M. \end{aligned} \quad (1.11)$$

Nyní víme, kolik agentů máme k dispozici, eliminuje-li se konkrétní úsek. Zároveň by nás ale také zajímalo, jaký úsek musíme eliminovat, abychom získali dostatek agentů k vynulování nějaké konkrétní složky.

Tvrzení 6. Uvažujme vektor $\mathbf{a} = (a_{-M}, \dots, a_N)$. Předpokládejme, že se úsek (a_{-m}, \dots, a_n) celý eliminuje kliknutím na a_0 . Pak úsek (a_{-m}, \dots, a_n) vypustí alespoň a_{n+1} agentů doprava právě tehdy, když $m \geq \sum_{j=1}^{n+1} (a_j - 1)$.

Zároveň úsek (a_{-m}, \dots, a_n) vypustí alespoň a_{-m+1} agentů doleva právě tehdy, když $n \geq \sum_{i=-1}^{-m+1} (a_i - 1)$.

Důkaz. Z předpokladu, že se úsek (a_{-m}, \dots, a_n) celý eliminuje, a ze vztahu 1.3 platí

$$m + n + 1 - \sum_{j=1}^n a_j = m + 1 - \sum_{j=1}^n (a_j - 1),$$

Alespoň a_{n+1} agentů vypustí právě tehdy, když platí

$$\begin{aligned} m + 1 - \sum_{j=1}^n (a_j - 1) &\geq a_{n+1}, \\ \text{tedy } m &\geq \sum_{j=1}^{n+1} (a_j - 1). \end{aligned}$$

Počet agentů vypuštěných doleva se dokáže analogicky. □

Tvrzení vede k následující definici.

Definice 13. Pro $m, n \geq 0$ definujeme funkce i, j předpisem

$$i(n) = \begin{cases} 0, & \text{pokud } n = 0, \\ \sum_{j=1}^n (a_j - 1), & \text{pokud } n > 0, \end{cases} \quad (1.12)$$

$$j(m) = \begin{cases} 0, & \text{pokud } m = 0, \\ \sum_{i=-1}^{-m} (a_i - 1), & \text{pokud } m > 0. \end{cases} \quad (1.13)$$

Lemma 7. Funkce i a j jsou neklesající.

Důkaz. Zřejmě pro každé $1 \leq n < N$ platí $0 \leq i(n) = \sum_{j=1}^n (a_j - 1) \leq \sum_{j=1}^{n+1} (a_j - 1) = i(n+1)$. Obdobně pak $0 \leq j(m) \leq j(m+1)$. □

Z předchozího plyne:

Tvrzení 8. Mějme pozici \mathbf{a} a předpokládejme, že úsek (a_{-m}, \dots, a_n) se eliminuje kliknutím na a_0 .

Pokud $n < N$, pak se eliminuje také úsek (a_{-m}, \dots, a_{n+1}) právě tehdy, když $m \geq i(n+1)$. Pokud $m < M$, pak se eliminuje také úsek $(a_{-(m+1)}, \dots, a_n)$ právě tehdy, když $n \geq j(m+1)$.

Důsledek 3. Pokud se pozice celá eliminuje kliknutím na a_0 , pak platí

$$i(N) \leq M, \quad (1.14)$$

$$j(M) \leq N. \quad (1.15)$$

Při splnění těchto podmínek pak můžeme i, j chápat jako zobrazení mezi množinami indexů

$$i : [0, \dots, N] \rightarrow [0, \dots, M],$$

$$j : [0, \dots, M] \rightarrow [0, \dots, N],$$

která pak můžeme skládat.

Příklady.

1. Pozice hrací plochy, kde není splněna podmínka 1.14. Prvek a_2 se neeliminuje.

$$\begin{pmatrix} 2 & 1 & 1 & 4 \\ -1 & 0 & 1 & 2 \end{pmatrix}$$

$$i(2) = 3 > 2, \quad j(1) = 1 < 2$$

2. Pozice hrací plochy, kde v obou podmínkách 1.14 a 1.15 platí rovnost, přesto se pozice neeliminuje celá. Podmínky 1.14 a 1.15 tedy nejsou postačující.

$$\begin{array}{c}
 i(2) = 2 \\
 \left(\begin{array}{ccccc}
 3 & 1 & 1 & 1 & 3 \\
 -2 & -1 & 0 & 1 & 2
 \end{array} \right) \\
 j(2) = 2
 \end{array}$$

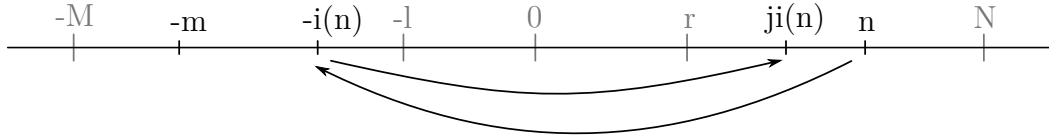
Jinými slovy zobrazení i, j říkájí, že pokud se eliminuje složka $a_{-i(n)}$, pak se eliminuje i složka a_n a pokud se eliminuje $a_{j(m)}$, pak se eliminuje i a_{-m} .

V příkladu 2. vidíme, že takto došlo k zacyklení, kdy k eliminaci a_2 se musí nejprve eliminovat a_{-2} , ale k eliminaci a_{-2} se nejprve musí eliminovat opět a_2 . Proto se neeliminuje ani a_2 , ani a_{-2} .

Pokusíme se popsat, jaké vlastnosti funkce i a j musí mít, aby došlo k eliminaci celé pozice. Využijeme faktu, že zobrazení i, j mezi množinami indexů můžeme skládat. Přirozeně tak můžeme počítat

$$ji(n) = \sum_{i=-1}^{-i(n)} (a_i - 1), \quad (1.16)$$

$$ij(m) = \sum_{j=1}^{j(m)} (a_j - 1). \quad (1.17)$$



Obrázek 1.2: Indexy prvků vektoru \mathbf{a}

Z obrázku 1.2 vidíme, že aby došlo k eliminaci a_n , musí být nejprve eliminováno $a_{-i(n)}$ a k eliminaci $a_{-i(n)}$ musí být nejprve eliminováno $a_{ji(n)}$. Aby to bylo možné, musí platit $ji(n) < n$. Z toho plyne nutná a postačující podmínka pro řešitelnost pozice:

Tvrzení 9. *Mějme pozici (a_{-M}, \dots, a_N) a předpokládejme, že platí podmínky 1.14 a 1.15. Pak se pozice celá eliminuje kliknutím na a_0 právě tehdy, když platí*

$$ji(n) < n, \quad \text{pro všechna } n = 1, \dots, N, \quad (1.18)$$

$$ij(m) < m, \quad \text{pro všechna } m = 1, \dots, M. \quad (1.19)$$

Důkaz. Nejprve předpokládejme, že platí obě podmínky 1.18 i 1.19.

Nechť (a_{-m}, \dots, a_n) je maximální úsek, který se eliminuje kliknutím na a_0 . Zároveň předpokládejme, že $n < N$ a $ji(n+1) < n+1$, tedy $j(i(n+1)) \leq n$.

Pak mohou nastat dvě situace. Je-li $i(n+1) \leq m$, pak z tvrzení 8 víme, že se eliminuje i celý úsek (a_{-m}, \dots, a_{n+1}) . Je-li naopak $m < i(n+1)$, tedy

$m + 1 \leq i(n + 1)$, pak z monotonie zobrazení j platí $j(m + 1) \leq ji(n + 1) \leq n$, a tedy z tvrzení 8 víme, že se eliminuje i úsek $(a_{-(m+1)}, \dots, a_n)$. V obou situacích tak nastává spor s maximalitou úseku (a_{-m}, \dots, a_n) .

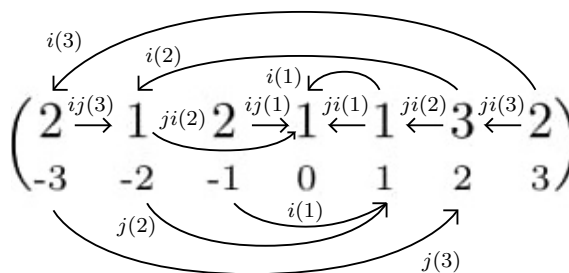
Opačnou implikaci dokážeme sporem. Necht (a_{-m}, \dots, a_n) je opět největší úsek, který se eliminuje kliknutím na a_0 a předpokládejme, že $n < N$.

Tedy se neeliminuje celý úsek $(a_{-(m+1)}, \dots, a_n)$, což nastává právě tehdy, když $m = M$, nebo z tvrzení 8 když $n < j(m + 1)$. Zároveň se neeliminuje ani celý úsek (a_{-m}, \dots, a_{n+1}) , což opět z tvrzení 8 nastává právě tehdy, když $m < i(n + 1)$.

Víme tedy, že $m < i(n + 1)$ platí vždy. Pokud je zároveň $m = M$, pak $M = m < i(n + 1) \leq i(N)$, což je spor s předpokladem, že platí podmínka $i(N) \leq M$. Pokud $n < j(m + 1)$, pak z monotonie j plyne $n < j(m + 1) \leq ji(n + 1)$, tj. $n + 1 \leq ji(n + 1)$, a tedy neplatí ani podmínka 1.18. □

Příklad 3. V obrázku 1.3 je uvedena složitější pozice, která se vynuluje kliknutím na a_0 .

Z šipek můžeme vyčíst, jak probíhá eliminace této pozice. Z vodorovných šipek uprostřed vidíme, že pozice splňuje podmínku z tvrzení 9. Z šipek nad a pod čísly můžeme vyčíst pořadí, v jakém se prvky eliminují. Příslušná zobrazení i , j , ij a ji jsou znázorněna šipkami.



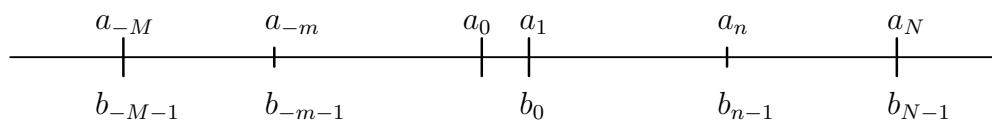
Obrázek 1.3: Pozice splňující podmínku z tvrzení 9

1.1.2 Rozšíření řešitelnosti pozice

Uvažujme pozici $\mathbf{a} = (a_{-M}, \dots, a_0, \dots, a_N)$. Nyní již máme prostředky ke zjištění, jestli kliknutím na a_0 dojde k eliminaci celé pozice. Nyní nás bude zajímat, za jakých podmínek může dojít k eliminaci celé pozice kliknutím na jiný prvek než a_0 hodnoty 1.

Předpokládejme tedy, že klikneme na $a_1 = 1$. Abychom zachovali značení, že klikáme na prvek a_0 , rozlišíme tyto dva případy jako dvě různé pozice hry. Je-li původní pozice $\mathbf{a} = (a_i)$, pak novou pozici hry označíme

$$\mathbf{b} = (b_i), \text{ kde } b_i = a_{i+1} \text{ pro všechna } i.$$



Obrázek 1.4: Posunutí indexů pozice \mathbf{b}

Z předpokladu, že se celá pozice \mathbf{a} eliminuje, víme, že tato pozice splňuje podmínky 1.18 a 1.19. Z nich budeme vycházet i u zjišťování podmínek řešitelnosti pozice \mathbf{b} .

Zobrazení i a j odlišíme podle toho, ke které pozici se vztahují, tj. pro pozici \mathbf{a} máme

$$i_{\mathbf{a}}(n) = \sum_{j=1}^n (a_j - 1), \quad j_{\mathbf{a}}(m) = \sum_{i=-1}^{-m} (a_i - 1), \quad (1.20)$$

pro pozici \mathbf{b} budeme počítat

$$i_{\mathbf{b}}(n) = \sum_{j=1}^n (b_j - 1), \quad j_{\mathbf{b}}(m) = \sum_{i=-1}^{-m} (b_i - 1). \quad (1.21)$$

Lemma 10. *Pro zobrazení $i_{\mathbf{b}}$ a $j_{\mathbf{b}}$ a pro $0 < m < M + 1$ a $0 < n < N - 1$ platí*

$$i_{\mathbf{b}}(n) = i_{\mathbf{a}}(n + 1), \quad j_{\mathbf{b}}(m) = j_{\mathbf{a}}(m - 1).$$

Důkaz. Ze vztahů 1.20 a 1.21 jednoduše plyne

$$\begin{aligned} i_{\mathbf{b}}(n) &= \sum_{j=1}^n (b_j - 1) = \sum_{j=1}^n (a_{j+1} - 1) = \sum_{j=2}^{n+1} (a_j - 1) \stackrel{(*)}{=} \sum_{j=1}^{n+1} (a_j - 1) = i_{\mathbf{a}}(n + 1), \\ j_{\mathbf{b}}(m) &= \sum_{i=-1}^{-m} (b_i - 1) = \sum_{i=-1}^{-m} (a_{i+1} - 1) = \sum_{i=0}^{-m+1} (a_i - 1) \stackrel{(*)}{=} \sum_{i=-1}^{-m+1} (a_i - 1) = \\ &= j_{\mathbf{a}}(m - 1). \end{aligned}$$

Rovnosti (*) plynou z předpokladu celé podkapitoly, tedy že $a_0 = a_1 = 1$. Sčítance $(a_1 - 1)$ a $(a_0 - 1)$ tak celkové součty neovlivní. \square

Abychom mohli zobrazení $i_{\mathbf{b}}$ a $j_{\mathbf{b}}$ skládat, musí platit nutná podmínka pro pozici \mathbf{b} , tedy $i_{\mathbf{b}}(N - 1) \leq M + 1$ a $j_{\mathbf{b}}(M + 1) \leq N - 1$. Zřejmě ale $i_{\mathbf{b}}(N - 1) = i_{\mathbf{a}}(N) \leq M < M + 1$, tedy tato podmínka je splněna vždy. Zároveň $j_{\mathbf{b}}(M + 1) \leq N - 1$, což platí právě tehdy, když $j_{\mathbf{a}}(M) < N$.

Předpokládejme, že je i tato podmínka splněna. Chceme rozhodnout, kdy je pozice \mathbf{b} řešitelná. Podle tvrzení 9 k tomu je nutné a stačí, aby zároveň platilo:

$$j_{\mathbf{b}}i_{\mathbf{b}}(n) < n \quad \text{pro všechna } n = 1, \dots, N - 1, \quad (1.22)$$

$$i_{\mathbf{b}}j_{\mathbf{b}}(m) < m \quad \text{pro všechna } m = 1, \dots, M + 1. \quad (1.23)$$

Tvrzení 11. *Mějme pozice \mathbf{a}, \mathbf{b} , pro které platí $b_i = a_{i+1}$ pro všechna i . Předpokládejme, že pozice \mathbf{a} je řešitelná kliknutím na a_0 a splňuje $j_{\mathbf{a}}(M) < N$. Pak platí podmínka 1.22 právě tehdy, když platí*

$$(n + 1) - j_{\mathbf{a}}i_{\mathbf{a}}(n + 1) + a_{-i_{\mathbf{a}}(n+1)} - 1 > 0 \quad \text{pro všechna } n = 1, \dots, N - 1 \quad (1.24)$$

a podmínka 1.23 platí právě tehdy, když

$$a_{j_{\mathbf{a}}(m-1)+1} > m - 1 - i_{\mathbf{a}}j_{\mathbf{a}}(m - 1) + 2 \quad \text{pro všechna } m = 1, \dots, M + 1. \quad (1.25)$$

Důkaz. Nejprve si pro $n \in \{1, \dots, N-1\}$ vyjádříme $j_{\mathbf{b}}i_{\mathbf{b}}(n)$ v závislosti na \mathbf{a} :

$$\begin{aligned} j_{\mathbf{b}}i_{\mathbf{b}}(n) &= j_{\mathbf{b}}(i_{\mathbf{a}}(n+1)) = j_{\mathbf{a}}(i_{\mathbf{a}}(n+1) - 1) = \sum_{i=-1}^{-(i_{\mathbf{a}}(n+1)-1)} (a_i - 1) = \\ &= \sum_{i=-1}^{-i_{\mathbf{a}}(n+1)} (a_i - 1) - (a_{-i_{\mathbf{a}}(n+1)} - 1) = j_{\mathbf{a}}i_{\mathbf{a}}(n+1) - (a_{-i_{\mathbf{a}}(n+1)} - 1) \end{aligned}$$

Chceme rozhodnout, kdy je splněna podmínka 1.22, vyjádříme si rozdíl

$$\begin{aligned} n - j_{\mathbf{b}}i_{\mathbf{b}}(n) &= n - j_{\mathbf{a}}(i_{\mathbf{a}}(n+1)) + (a_{-i_{\mathbf{a}}(n+1)} - 1) \\ &= (n+1) - j_{\mathbf{a}}(i_{\mathbf{a}}(n+1)) + (a_{-i_{\mathbf{a}}(n+1)} - 1) - 1, \end{aligned}$$

$$\text{a tedy } n - j_{\mathbf{b}}i_{\mathbf{b}}(n) > 0 \Leftrightarrow (n+1) - j_{\mathbf{a}}i_{\mathbf{a}}(n+1) + (a_{-i_{\mathbf{a}}(n+1)} - 1) - 1 > 0.$$

Analogicky pro $m \in \{1, \dots, M+1\}$ máme

$$\begin{aligned} i_{\mathbf{b}}j_{\mathbf{b}}(m) &= i_{\mathbf{b}}(j_{\mathbf{a}}(m-1)) = i_{\mathbf{a}}(j_{\mathbf{a}}(m-1) + 1) = \sum_{j=1}^{j_{\mathbf{a}}(m-1)+1} (a_j - 1) = \\ &= \sum_{j=1}^{j_{\mathbf{a}}(m-1)} (a_j - 1) + (a_{j_{\mathbf{a}}(m-1)+1} - 1) = i_{\mathbf{a}}j_{\mathbf{a}}(m-1) + (a_{j_{\mathbf{a}}(m-1)+1} - 1). \end{aligned}$$

Opět vyjádříme rozdíl

$$\begin{aligned} m - i_{\mathbf{b}}j_{\mathbf{b}}(m) &= m - i_{\mathbf{a}}(j_{\mathbf{a}}(m-1)) + (a_{j_{\mathbf{a}}(m-1)+1} - 1) \\ &= (m-1) - i_{\mathbf{a}}(j_{\mathbf{a}}(m-1)) - a_{j_{\mathbf{a}}(m-1)+1} + 2, \end{aligned}$$

$$\text{a tedy } m - i_{\mathbf{b}}j_{\mathbf{b}}(m) > 0 \Leftrightarrow a_{j_{\mathbf{a}}(m-1)+1} < m - 1 - i_{\mathbf{a}}j_{\mathbf{a}}(m-1) + 2.$$

□

Nyní ukážeme, že je snazší ověřit, zda je pozice \mathbf{b} neřešitelná.

Tvrzení 12. *Pozice \mathbf{b} není řešitelná právě tehdy, když existuje $n \in \{1, \dots, N-1\}$ takové, že*

$$(n+1) - j_{\mathbf{a}}i_{\mathbf{a}}(n+1) + (a_{-i_{\mathbf{a}}(n+1)} - 1) - 1 \leq 0, \quad (1.26)$$

nebo když existuje $m \in \{1, \dots, M+1\}$ takové, že

$$a_{j_{\mathbf{a}}(m-1)+1} \geq m - 1 - i_{\mathbf{a}}j_{\mathbf{a}}(m-1) + 2. \quad (1.27)$$

Podmínka 1.26 nastává právě tehdy, když současně platí

$$(n+1) - j_{\mathbf{a}}i_{\mathbf{a}}(n+1) = 1 \text{ a } a_{-i_{\mathbf{a}}(n+1)} = 1. \quad (1.28)$$

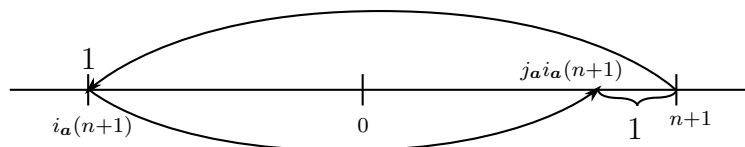
Důkaz. Podmínky 1.26 a 1.27 přímo plynou z tvrzení 11.

Předpokládejme, že pro nějaké $n \in \{1, \dots, N-1\}$ platí 1.26. Zároveň víme, že pozice \mathbf{a} je řešitelná a z tvrzení 9 tak pro každé n platí $(n+1) - j_{\mathbf{a}}i_{\mathbf{a}}(n+1)$

1) > 0 . Zároveň víme, že $a_{-i_a(n+1)} \geq 1$ pro všechna n . Tedy pro všechna n platí $(n+1) - j_a i_a(n+1) + (a_{-i_a(n+1)} - 1) - 1 \geq 0$.

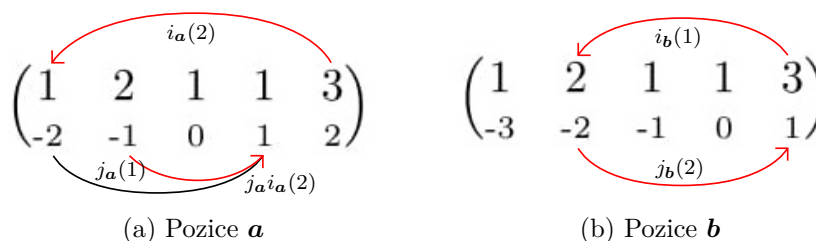
Pro to n , které splňuje zároveň podmínku 1.26 pak máme $0 \geq (n+1) - j_a i_a(n+1) + (a_{-i_a(n+1)} - 1) - 1 \leq 0$, tedy výraz je roven 0, a to nastává právě tehdy, když platí 1.28. □

Ověření podmínky 1.26, neboli 1.28, se tak zužuje na ověření toho, zda nastane situace jako na obrázku 1.5.



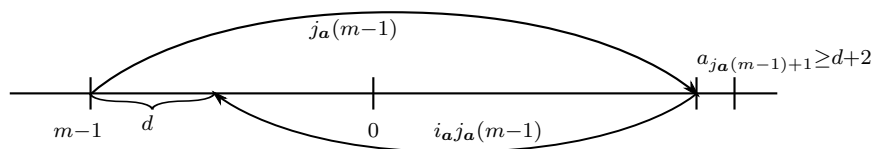
Obrázek 1.5: Podmínka neřešitelnosti pozice **b** I

Obrázek 1.6 ilustruje, jak neřešitelnost pozice **b** souvisí s podmínkou 1.26. Na obrázku 1.6a je zobrazena pozice **a**, kde podmínka platí pro $n+1 = 2$. Ve skutečnosti ale fakt, že $a_{-2} = 1$ znamená, že $j_a(2) = j_a(1)$. Z červených šipek tak vidíme, že pro pozici **b** neplatí podmínka 1.22 pro $n = 1$, neboli zobrazení i_b a j_b „se zacyklí“.



Obrázek 1.6: Příklad podmínky 1.26

Případné ověření podmínky 1.27 již tak jednoduché není. Museli bychom se pro každé $m-1$ podívat, zda hodnota prvku s indexem $j_a(m-1) + 1$ zvětšená o 2 není větší nebo rovna rozdílu $m-1 - i_a j_a(m-1)$, v obrázku 1.7 označeného d .



Obrázek 1.7: Podmínka neřešitelnosti pozice **b** II

Ve skutečnosti ale stačí ověřit pouze podmínku 1.26 z tvrzení 12.

Tvrzení 13. *Pozice **b** není řešitelná právě tehdy, když existuje $n \in \{1, \dots, N-1\}$ takové, že platí 1.26 (nebo ekvivalentně 1.23).*

Důkaz. Předpokládejme, že pozice \mathbf{b} není řešitelná. Pak buď existuje n , pro které platí 1.26 a jsme hotovi, nebo existuje m takové, že platí 1.27. V druhém případě pak platí $i_b j_b(m) \geq m$, a tedy také $j_b i_b j_b(m) \geq j_b(m)$.

Pokud $j_b i_b j_b(m) = j_b(m)$, pak položíme $n = j_b(m)$ a pro toto n platí 1.26. Pokud $j_b i_b j_b(m) > j_b(m)$, pak z monotonie i_b a j_b platí $j_b i_b j_b i_b j_b(m) \geq j_b i_b j_b(m)$. Platí-li rovnost, pak položíme $n = j_b i_b j_b(m)$ a pro toto n opět platí 1.26.

Platí-li ostrá nerovnost $j_b i_b j_b i_b j_b(m) > j_b i_b j_b(m)$, pak znovu použijeme $j_b i_b$ na tuto nerovnost zleva a dostaneme $(j_b i_b)^3 j_b(m) \geq (j_b i_b)^2 j_b(m)$. Opět buď nastane rovnost a položíme $n = (j_b i_b)^2 j_b(m)$ a pro toto n platí 1.26, nebo platí ostrá nerovnost $(j_b i_b)^3 j_b(m) > (j_b i_b)^2 j_b(m)$. Dále můžeme postup opakovat. Všechny prvky posloupnosti $(j_b i_b)^k j_b(m)$ jsou ale omezeny shora číslem $N-1$, tedy existuje k takové, že $(j_b i_b)^{k+1} j_b(m) = (j_b i_b)^k j_b(m)$. Vezmeme nejmenší takové k a opět položíme $n = (j_b i_b)^k j_b(m)$ a pro toto n platí 1.26.

Opacná implikace je zřejmá. □

Důsledkem tohoto tvrzení je tedy existence jednoduchého algoritmu, jak ověřit řešitelnost pozice \mathbf{b} .

V případě že $a_{-1} = 1$, můžeme stejný postup použít k ověření řešitelnosti pozice

$$\mathbf{c} = (c_i), \text{ kde } c_i = a_{i-1} \text{ pro všechna } i.$$

S analogickými důkazy tak pro zobrazení $i_c(n) = i_a(n-1)$ a $j_c(m) = j_a(m+1)$ platí tvrzení

Tvrzení 14. *Mějme pozice \mathbf{a}, \mathbf{c} , pro které platí $c_i = a_{i-1}$ pro všechna i . Předpokládejme, že pozice \mathbf{a} je řešitelná kliknutím na a_0 a splňuje $i_a(N) < M$. Pak pozice \mathbf{c} není řešitelná právě tehdy, když existuje $m = 1, \dots, M-1$ takové, že platí*

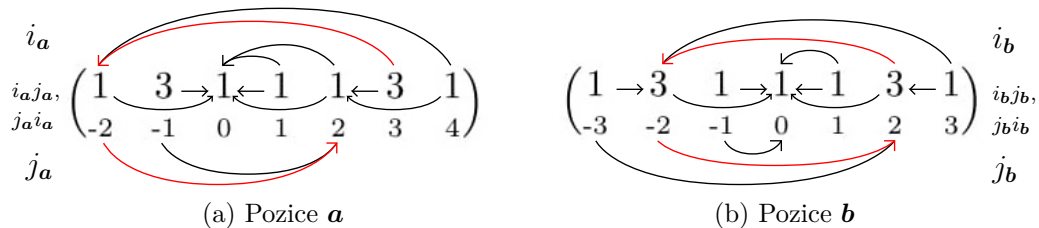
$$(m+1) - j_a i_a(m+1) + a_{-i_a(m+1)} - 1 \leq 0, \quad (1.29)$$

Analogicky pak platí i jednoduchý algoritmus k ověření řešitelnosti pozice \mathbf{c} .

Následující příklad ukazuje, že indexy prvků, na které můžeme kliknout, aby byla pozice řešitelná, nemusí tvořit interval.

Příklad 4. Pro přehlednost jsou vedle obrázků pouze vypsána zobrazení, která znázorňují šipky na té úrovni.

Z červených šipek na obrázku 1.8a u pozice \mathbf{a} a z toho, že $a_{-2} = 1$, víme podle tvrzení 12, že kliknutím na a_1 se pozice neeliminuje celá. Ověříme si tuto skutečnost na obrázku 1.8b, kde uvážíme pozici \mathbf{b} , tedy že $b_i = a_{i+1}$ pro všechna i . Vidíme, že červené šipky nyní tvoří cyklus a tedy nedojde již k eliminaci ani b_{-2} , ani b_2 .



Obrázek 1.8: Řešitelnost hrací plochy

Ze symetrie hrací plochy je zřejmé, že kliknutím na a_2 je pozice \mathbf{a} také řešitelná a z tvrzení 14 bychom opět zjistili, že kliknutím na a_1 není.

1.1.3 Čas eliminace

Nyní již umíme rozhodnout, zda je vektor řešitelný jedním klikem, či nikoliv. Pokud je, v této části ukážeme, že můžeme spočítat, kdy došlo k eliminaci libovolné složky.

Definice 14. Předpokládejme, že je vektor \mathbf{a} řešitelný klikem na a_0 . Pak číslo t_i bude značit čas eliminace složky a_i . Budeme uvažovat $t_0 = 0$.

Zřejmě pro úsek jedniček $(a_{-l}, \dots, a_0, \dots, a_r)$ platí $t_i = |i|$.

Lemma 15. Necht vektor $\mathbf{a} = (a_{-M}, \dots, a_0, \dots, a_N)$ je řešitelný klikem na a_0 . Pak pro $m, n > 0$ platí

$$t_n = t_{-i(n)} + i(n) + n, \quad (1.30)$$

$$t_{-m} = t_{j(m)} + j(m) + m. \quad (1.31)$$

Důkaz. Stačí pro 1.30, 1.31 dokázat analogicky. Zřejmě agent, který vznikl eliminací $a_{-i(n)}$ a míří doprava, je právě ten, který způsobí eliminaci složky a_n . Čas, za který tento agent dojde od $a_{-i(n)}$ k a_n je roven jejich vzdálenosti, tedy právě rozdílu $n - (-i(n)) = n + i(n)$. Úpravou dostaneme vztah 1.30. \square

Lemma 16. Za stejných předpokladů jako lemma 15 platí

$$t_n = t_{ji(n)} + ji(n) + 2i(n) + n, \quad (1.32)$$

$$t_{-m} = t_{-ij(m)} + ij(m) + 2j(m) + m. \quad (1.33)$$

Důkaz. Dosazením $i(n)$ za m do vztahu 1.31 dostaneme $t_{-i(n)} = t_{ji(n)} + ji(n) + i(n)$, dosadíme do 1.30 a po upravení získáme rovnost 1.32. Pro t_{-m} opět analogicky. \square

Za n můžeme dosadit libovolný kladný index, tedy i $ji(n)$. Dostáváme tak vztah

$$t_{ji(n)} = t_{jiji(n)} + jiji(n) + 2iji(n) + ji(n). \quad (1.34)$$

Dosadíme-li do rovnosti 1.30, vyjde

$$t_n = t_{jiji(n)} + jiji(n) + 2iji(n) + 2ji(n) + 2i(n) + n. \quad (1.35)$$

Takto můžeme postupně dosazovat, dokud čas $t_{ji..ji(n)}$ neznáme, nebo není snadný spočítat (např. úsek jedniček). Jsme tedy schopni spočítat čas eliminace jakékoliv složky s kladným indexem. Navíc vztahy 1.34 a 1.35 se získají analogicky i pro záporné indexy a díky vztahům 1.34 a 1.35 umíme spočítat čas vynulování libovolné složky vektoru.

Závěr

V této práci jsme nejprve popsali pravidla hry Bubble Blast 2 a s nimi související pojmy. Zavedli jsme maticový zápis dvourozměrné hrací plochy a popsali její dynamiku. Pro pochopení průběhu a řešitelnosti dvourozměrné verze hry bylo nutné nejprve pochopit jednodušší variantu. Věnovali jsme se tak zkoumání hry na přímce popsané vektorem a předpokládali jsme navíc, že hráč má k dispozici právě jedno kliknutí. Dále jsme dokázali několik tvrzení o chování agentů a vlastnostech stavu hry, které nakonec vedly k formulaci a důkazu nutné a postačující podmínky řešitelnosti zjednodušené jednorozměrné verze hry. Důsledkem pak bylo tvrzení 13, které umožňuje poznat, kdy je pozice řešitelná kliknutím na jeden prvek řešitelná také kliknutím na sousední prvek. Z tvrzení 13 plyne jednoduchý algoritmus, jak to ověřit. Na konci práce jsme také ukázali, že je možné spočítat čas eliminace jednotlivých prvků jednorozměrné hrací plochy.

Využitím informace o přesném času eliminace prvků by se zkoumání řešitelnosti mohlo rozšířit na jednorozměrnou verzi s více možnými kliknutími. Tato varianta by se již dala využít k analýze původní verze, kdy se všechny nenulové prvky nachází v jednom řádku, popřípadě sloupci matice.