

Univerzita Pavla Jozefa Šafárika v Košiciach

Prírodovedecká fakulta

# HALF-LIFE

## DIPLOMOVÁ PRÁCA

Študijný odbor:

Informatika

Školiace pracovisko:

Ústav informatiky

Vedúci záverečnej práce:

prof. RNDr. Stanislav Krajči, PhD.

Košice 2026

Bc. Vasil' Chorev

## **Podakovanie**

Rád by som poďakoval prof. RNDr. Stanislavovi Krajčimu, PhD., za cenné pripomienky a za obetavosť počas tvorby mojej diplomovej práce. V neposlednom rade veľká vďaka patrí mojej rodine a blízkym priateľom, ktorí ma neustále podporovali počas môjho štúdia.



Univerzita P. J. Šafárika v Košiciach  
Prírodovedecká fakulta

---

## ZADANIE ZÁVEREČNEJ PRÁCE

- Meno a priezvisko študenta:** Bc. Vasil' Chorev  
**Študijný program:** informatika (jednoodborové štúdium, magisterský II. st., denná forma)  
**Študijný odbor:** Informatika  
**Typ záverečnej práce:** Diplomová práca  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický
- Názov:** Half-Life  
**Názov EN:** Half-Life  
**Cieľ:**  
1 Navrhnuť fuzziifikáciu Conwayovej hry Life.  
2 Pripraviť softvérové prostredie na editáciu a pozorovanie dynamiky hry.  
3 Nájsť zmysluplné sady parametrov fuzziifikácie.  
4 Hľadať pre takéto sady konfigurácie s predpísaným správaním (špeciálne s osciláciou), prípadne ich aj exaktne charakterizovať.
- Literatúra:**  
1. L. Atanassova and K. Atanassov, Intuitionistic Fuzzy Interpretations of Conway's Game of Life, in: Numerical Methods and Applications, Springer, 2011, pp. 232–239.  
2. Eugene M. Izhikevich et al., Game of Life, Scholarpedia, 10(6):1816, 2015.  
3. Lakhal, Golly, An open source, cross-platform application for exploring Conway's Game of Life and other cellular automata.
- Vedúci:** prof. RNDr. Stanislav Krajčí, PhD.  
**Oponent:** doc. RNDr. Ondrej Krídlo, PhD.  
**Ústav :** ÚINF - Ústav informatiky  
**Riaditeľ ústavu:** doc. RNDr. Ondrej Krídlo, PhD.  
**Dátum schválenia:** 22.04.2026

## Abstrakt

Conwayova Hra života predstavuje základný dvojrozmerný celulárny automat. Predložená práca najskôr poskytuje teoretický prehľad Hry života, zahŕňajúci jej historický vývoj, formálne definície a matematické dôkazy kľúčových vlastností. Hlavným prínosom práce je návrh a analýza nového trojstavového celulárneho automatu *Half-Life*. Tento model možno interpretovať ako diskretnú aproximáciu fuzzy verzie Hry života. Na preskúmanie jeho rozsiahleho priestoru parametrov sme v jazyku Rust implementovali vyhľadávací nástroj na automatickú klasifikáciu vznikajúcich vzorov. Vďaka tomuto nástroju sme zostavili katalóg 514 unikátnych pravidiel podporujúcich vznik pohyblivých štruktúr a 3 381 vzorov. Uskutočnené experimenty naznačili, že nesúvislé intervaly prežitia môžu súvisieť s častejším výskytom pohyblivých alebo oscilujúcich vzorov. Novoobjavené pravidlá a vzory sme následne integrovali do interaktívnej webovej aplikácie vyvinutej v knižnici React, ktorá slúži na simuláciu a vizualizáciu rôznych tried celulárnych automatov v reálnom čase.

**Kľúčové slová:** *celulárne automaty, Game of Life, viacstavové automaty, automat Half-Life, fuzzy logika, simulácia*

## Abstract

Conway's Game of Life represents a fundamental two-dimensional cellular automaton. The presented work first provides a theoretical overview of the Game of Life, covering its history, formal definitions, and mathematical proofs of key properties. The primary contribution of the thesis is the design and analysis of a novel three-state cellular automaton *Half-Life*. This model can be interpreted as a discrete approximation of a fuzzy version of the Game of Life. To explore its vast parameter space, we implemented a parallel search tool in the Rust programming language for automated classification of emerging patterns. Using this tool, we compiled a catalog of 514 unique rules supporting the emergence of moving structures and 3381 patterns. The conducted experiments suggest that discontinuous survival intervals may be associated with a higher occurrence of moving or oscillating patterns. The newly discovered rules and patterns were subsequently integrated into an interactive web application developed using the React library, enabling real-time simulation and visualization of various classes of cellular automata.

**Keywords:** *cellular automata, Game of Life, multistate cellular automata, Half-Life automaton, fuzzy logic, simulation*

# Obsah

Úvod	7
<b>1 Hra života</b>	<b>8</b>
1.1 Formálna definícia a základné typy vzorov . . . . .	10
1.2 História a klasifikácia vzorov . . . . .	11
1.3 Stabilné vzory . . . . .	12
1.4 Prísne stabilné vzory . . . . .	13
1.5 Oscilátory . . . . .	13
1.6 Lode a posuvné vzory . . . . .	16
1.7 Generátory klzákov . . . . .	20
1.8 Rajské záhrady . . . . .	20
1.9 Výpočtová univerzálnosť . . . . .	23
1.9.1 Generátor signálu . . . . .	23
1.9.2 Požierač signálu . . . . .	24
1.9.3 Logika zrážok a pamäť . . . . .	24
<b>2 Prehľad existujúcich prístupov</b>	<b>27</b>
2.1 Generácie Hry života . . . . .	27
2.2 Jednorozmerné celulárne automaty . . . . .	28
2.3 Trojrozmerné celulárne automaty . . . . .	29
2.4 Viacstavové automaty . . . . .	30
2.5 Režim konečnej teploty . . . . .	31
2.6 Moderné priestorové a spojité zovšeobecnenia . . . . .	32
2.6.1 Larger than Life (LtL) . . . . .	32
2.6.2 SmoothLife . . . . .	33
2.6.3 Lenia . . . . .	34
2.7 Fuzzy a intuicionistické celulárne automaty . . . . .	35

2.8	Aplikácie a presahy do iných disciplín . . . . .	36
<b>3</b>	<b>Celulárny automat Half-Life</b>	<b>37</b>
3.1	Formálna definícia . . . . .	37
3.2	Pohyb klzáka . . . . .	40
3.3	Rajské záhrady . . . . .	43
3.4	Parametrizácia a priestor pravidiel . . . . .	44
<b>4</b>	<b>Analytický nástroj Half-Life Explorer</b>	<b>45</b>
4.1	Architektúra a jadro systému . . . . .	45
4.2	Algoritmická klasifikácia správania a vzorov . . . . .	46
4.3	Výsledky prieskumu . . . . .	48
4.4	Galéria objavených vzorov . . . . .	50
4.4.1	Rýchlosť a pohyb . . . . .	50
4.4.2	Periodicita . . . . .	52
4.4.3	Morfologické vlastnosti vybraných vzorov . . . . .	53
4.5	Integrácia výsledkov do aplikácie Fuzzy Life . . . . .	56
4.6	Porovnanie s existujúcimi nástrojmi . . . . .	56
<b>5</b>	<b>Webová aplikácia Fuzzy Life</b>	<b>58</b>
5.1	Architektúra aplikácie . . . . .	59
5.2	Simulačné režimy . . . . .	60
5.2.1	Klasický režim . . . . .	61
5.2.2	Režim Half-Life . . . . .	61
5.2.3	Spojité režim . . . . .	62
5.2.4	Režim konečnej teploty . . . . .	62
5.2.5	Jednorozmerný režim (1D) . . . . .	63
5.2.6	Režim Quartiles . . . . .	64
5.3	Interaktívny vyhľadávač vzorov . . . . .	65
	<b>Záver</b>	<b>67</b>
	<b>Zoznam použitej literatúry</b>	<b>68</b>

# Úvod

Hra ako koncept sprevádza ľudské myslenie od antiky. Už Platón a Aristoteles reflektovali hru ako osobitnú formu činnosti, ktorá nie je viazaná bezprostrednou užitočnosťou, no rozvíja schopnosť porozumieť pravidlám, štruktúram a vzťahom. V modernom chápaní sa hra stáva modelom uzavretého systému s presne definovanými pravidlami a postupnosťou krokov. Hranie tak predstavuje interakciu s formálnym systémom, ktorého vlastnosti možno analyzovať nezávisle od konkrétneho hráča.

Nie všetky hry však vyžadujú aktívne rozhodovanie. Existujú hry, v ktorých hráč po určení počiatočných podmienok už do procesu nezasahuje. Systém sa vyvíja autonómne podľa pevne daných pravidiel. Takéto hry sú v skutočnosti dynamickými modelmi, kde pozorovanie nahrádza stratégiu. Ich význam spočíva nie v súťaži, ale v analýze správania systému samotného.

Jednou z takýchto hier je Conwayova Hra života (angl. Game of Life). Na prvý pohľad ide o mimoriadne jednoduchý systém: dvojrozmerná mriežka buniek, z ktorých každá môže byť v jednom z dvoch stavov: živá alebo mŕtva. Neexistuje hráč v tradičnom zmysle slova, neexistuje cieľ ani víťazstvo. Napriek tomu si však získal mimoriadnu popularitu a natrvalo upútal pozornosť vedeckej komunity aj širokej verejnosti.

Cieľom tejto práce je preskúmať Conwayovu Hru života z teoretického aj praktického hľadiska a navrhnúť jej rozšírenie, v ktorom bunky nadobúdajú viaceré stavy. Hlavným prínosom práce je návrh a definícia nového trojstavového celulárneho automatu, vývoj nástroja na automatizované hľadanie pravidiel a vzorov, ako aj vytvorenie webovej aplikácie na ich vizualizáciu.

Práca je rozdelená do piatich kapitol. Prvá kapitola poskytuje teoretický základ, formálnu definíciu Hry života a klasifikáciu známych vzorov. Druhá kapitola mapuje existujúce modely rozširujúce klasické celulárne automaty. Tretia kapitola formálne definuje model *Half-Life*. Štvrtá kapitola opisuje implementáciu analytického nástroja a zároveň sumarizuje výsledky experimentov. Piata kapitola je venovaná architektúre a funkcionalite vyvinutej webovej aplikácie *Fuzzy Life*.

# Kapitola 1

## Hra života

Hra života, ktorú navrhol John Horton Conway, bola prvýkrát publikovaná v októbri 1970 v stĺpčeku *Mathematical Games* Martina Gardnera v časopise *Scientific American* [1]. Ako bolo uvedené v úvode, nejde o hru v klasickom poňatí, ale skôr o simuláciu života buniek. Priestor hry je rozdelený na nekonečnú štvorcovú mriežku buniek. Každá bunka sa v danom diskretnom čase nachádza v jednom z dvoch stavov: *živá* alebo *mŕtva*. Stav bunky v nasledujúcom časovom kroku závisí výlučne od stavov jej bezprostredných susedov v aktuálnom čase.

Za bezprostredné okolie bunky sa považuje osem susedných buniek.<sup>1</sup> Vývoj prebieha v diskretných krokoch, pričom všetky bunky sa aktualizujú súčasne podľa nasledujúcich pravidiel:

- Ak je bunka živá, prežije do nasledujúcej generácie práve vtedy, ak má 2 alebo 3 živých susedov; inak zanikne.
- Ak je bunka mŕtva, v nasledujúcej generácii sa stane živou práve vtedy, ak má presne 3 živých susedov; inak zostáva mŕtva.

Aplikáciu uvedených pravidiel možno názorne ukázať na jednoduchom príklade. Uvažujme situáciu na Obrázku 1.1. Toto rozloženie buniek je známe ako blikáč (angl. *blinker*). Ide o jednu z najjednoduchších netriviálnych konfigurácií Hry života, ktorá sa pravidelne opakuje s periódou 2. Obrázok demonštruje prechod od počiatočnej konfigurácie k nasledujúcemu stavu a ilustruje, ako počet živých susedov rozhoduje o prežití, zániku alebo vzniku bunky.

---

<sup>1</sup>Toto okolie je známe ako Moorovo susedstvo. Existujú aj varianty Hry života, ktoré využívajú iné typy susedstiev, napríklad von Neumannovo susedstvo, ktoré zahŕňa iba štyroch najbližších susedov (hore, dole, vľavo, vpravo).

0	1	1	1	0
0	2	1	2	0
0	3	2	3	0
0	2	1	2	0
0	1	1	1	0

 $\xrightarrow{\oplus 1}$ 

0	0	0	0	0
1	2	3	2	1
1	1	2	1	1
1	2	3	2	1
0	0	0	0	0

Obr. 1.1: Vývoj blikača. Čísla v bunkách udávajú počet živých susedov. **Zelenou** sú vyznačené bunky, ktoré v ďalšom kroku ožijú alebo prežijú, **červenou** tie, ktoré zaniknú alebo ostanú mŕtve.

Existuje množstvo ďalších pravidiel, ktoré sú predmetom štúdia. Tieto pravidlá sa štandardne zapisujú pomocou tzv. *rulestring* tvaru

$$Bx/Sy,$$

kde  $x$  a  $y$  sú množiny prirodzených čísel udávajúce počty susedov potrebné na zrod, resp. prežitie bunky (B z angl. *birth*, S z angl. *survival*). Pravidlo B3/S23 zodpovedá klasickej Hre života. Každé pravidlo je úplne určené dvoma podmnožinami množiny  $\{0, 1, \dots, 8\}$ :

- množinou počtov susedov, pri ktorých sa mŕtva bunka narodí,
- množinou počtov susedov, pri ktorých živá bunka prežije.

Takto definovaných pravidiel je celkovo  $2^{18}$ , t. j. 262 144. Každé z nich pritom určuje iný tzv. *Life-like* celulárny automat.

Pri skúmaní Hry života sa prirodzene vynára otázka, prečo boli jej prechodové pravidlá zvolené práve v podobe známej schémy B3/S23. Z teoretického hľadiska totiž neexistuje žiadne obmedzenie, ktoré by determinovalo práve túto konfiguráciu pravidiel.

Z tohto pohľadu je Hra života len jedným konkrétnym bodom v širokej triede možných pravidiel. Napriek tomu má práve pravidlo B3/S23 v literatúre a vo výskume osobitné postavenie. Dôvody tohto postavenia možno rozdeliť do niekoľkých kategórií [2].

- **Jednoduchosť pravidiel.** Pravidlo B3/S23 je ľahko zapamätateľné a má prirodzenú interpretáciu: málo susedov znamená izoláciu, vysoký počet susedov vedie k zániku bunky v dôsledku nadmerného počtu susedov a iba stredný počet umožňuje prežitie.
- **Charakter dynamiky.** Hra života sa nachádza medzi režimami vedúcimi k rýchlemu rastu populácie a režimami vedúcimi k zániku všetkých buniek, čo umožňuje vznik stabilných a dlhodobo pretrvávajúcich štruktúr.
- **Historický význam.** Ide o najviac skúmané pravidlo medzi *Life-like* automatmi, s bohatou zbierkou známych konfigurácií a teoretických výsledkov.

## 1.1 Formálna definícia a základné typy vzorov

Hru života možno formálne definovať ako dvojrozmerný celulárny automat. Jeho jednotlivé komponenty a pravidlá sú definované postupne.

**Definícia 1.1** (Mriežka). *Mriežkou* budeme nazývať množinu  $\mathbb{Z}^2$ .

**Definícia 1.2** (Bunka). *Bunkou* budeme nazývať prvok mriežky.

**Definícia 1.3** (Stav bunky). *Stavom bunky* budeme nazývať prvok z množiny  $\{0, 1\}$ , pričom hodnota 0 označuje mŕtvu bunku a hodnota 1 označuje živú bunku.

*Poznámka.* Každdej bunke mriežky prislúcha práve jeden stav. Vo vizualizáciách budeme bunku so stavom 1 znázorňovať čiernou farbou a bunku so stavom 0 bielou farbou.

**Definícia 1.4** (Konfigurácia). *Konfiguráciou* budeme nazývať funkciu  $c: \mathbb{Z}^2 \rightarrow \{0, 1\}$ , ktorá každej bunke mriežky priraduje jej stav. Množinu všetkých konfigurácií budeme označovať Konfigurácie.

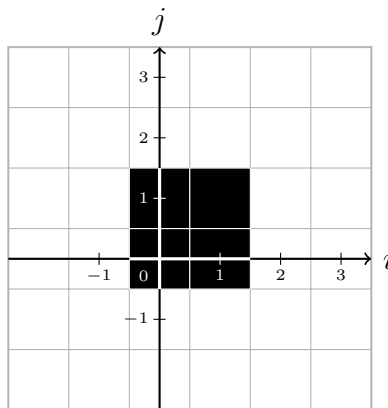
**Definícia 1.5** (Vzor). Konfiguráciu  $c$  budeme nazývať *vzorom*, ak množina

$$\{x \in \mathbb{Z}^2 : c(x) = 1\}$$

je konečná.

*Príklad 1.* Vzor na Obrázku 1.2 môžeme reprezentovať napríklad konfiguráciou  $c$ , v ktorej sú živé práve bunky na pozíciách  $\{\langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle\}$ . Formálne ide o funkciu

$$c(\langle i, j \rangle) = \begin{cases} 1 & \text{ak } \langle i, j \rangle \in \{\langle 0, 0 \rangle, \langle 1, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle\}, \\ 0 & \text{inak.} \end{cases}$$



Obr. 1.2: Konfigurácia  $c$  reprezentujúca vzor so štyrmi živými bunkami.

**Definícia 1.6** (Posuny). *Posunmi* budeme nazývať prvky množiny  $\{-1, 0, 1\}^2 \setminus \{(0, 0)\}$ . Množinu všetkých posunov budeme označovať Posuny.

**Definícia 1.7** (Súčet susedov). Nech  $c$  je konfigurácia a  $x$  je bunka. Definujme funkciu SúčetSusedov: Konfigurácie  $\times \mathbb{Z}^2 \rightarrow \mathbb{N}$  nasledovne

$$\text{SúčetSusedov}(c, x) = \sum_{n \in \text{Posuny}} c(x + n).$$

**Definícia 1.8** (Lokálna prechodová funkcia). Nech  $B, S \subseteq \{0, 1, \dots, 8\}$  a nech  $c$  je konfigurácia a  $x$  je bunka. Funkciu  $f_{B,S}$ : Konfigurácie  $\times \mathbb{Z}^2 \rightarrow \{0, 1\}$  definovanú predpisom

$$f_{B,S}(c, x) = \begin{cases} 1 & \text{ak } c(x) = 0 \text{ a } \text{SúčetSusedov}(c, x) \in B, \\ 1 & \text{ak } c(x) = 1 \text{ a } \text{SúčetSusedov}(c, x) \in S, \\ 0 & \text{inak} \end{cases}$$

budeme nazývať *lokálnou prechodovou funkciou*.

**Definícia 1.9** (Globálna prechodová funkcia). Nech  $B, S \subseteq \{0, 1, \dots, 8\}$  a nech  $c$  je konfigurácia. Funkciu  $\text{GPF}_{B,S}$ : Konfigurácie  $\rightarrow$  Konfigurácie definovanú predpisom

$$(\text{GPF}_{B,S}(c))(x) = f_{B,S}(c, x), \quad \text{pre každé } x \in \mathbb{Z}^2,$$

budeme nazývať *globálnou prechodovou funkciou*.

*Poznámka.* V prípade klasickej Hry života kladieme  $B = \{3\}$  a  $S = \{2, 3\}$  a globálnu prechodovú funkciu  $\text{GPF}_{B,S}$  označujeme skrátene GPF.

**Definícia 1.10** (Evolúcia). *Evolúciou systému* budeme nazývať postupnosť  $(c_t)_{t=0}^{\infty}$  prvkov množiny Konfigurácie, pre ktorú v každom kroku  $t \geq 0$  platí:

$$c_{t+1} = \text{GPF}(c_t).$$

## 1.2 História a klasifikácia vzorov

Krátko po zverejnení hra upútala pozornosť matematikov, informatikov aj amatérskych nadšencov. Prvé skúmania Hry života mali prevažne experimentálny charakter a vychádzali z analýzy vývoja konkrétnych počiatočných konfigurácií.

Po jej popularizácii sa však výskum rýchlo posunul k systematickému skúmaniu evolúcie hry. Ukázalo sa, že aj z jednoduchých deterministických pravidiel vznikajú rôznorodé štruktúry, čo otvorilo otázky periodicity, stability a dlhodobého vývoja systému, a preto

sa jedným z hlavných cieľov ďalšieho výskumu stala klasifikácia vzorov podľa ich správania v čase. Táto klasifikácia, ustálená v literatúre okolo Conwaya a neskôr formalizovaná v monografii *Winning Ways for your Mathematical Plays* [3], rozdeľuje vzory najmä do nasledujúcich tried.

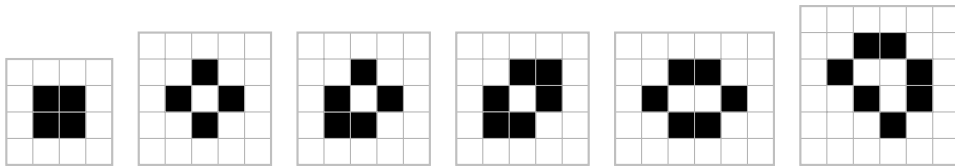
### 1.3 Stabilné vzory

Stabilné vzory sú najjednoduchšie vzory v Hre života, a to také, ktoré sa v priebehu evolúcie vôbec nemenia. Zostávajú identické v každej generácii a označujú sa ako stabilné vzory (angl. *still lifes*).

**Definícia 1.11** (Stabilný vzor). Vzor  $c$  nazývame *stabilným vzorom*, ak platí

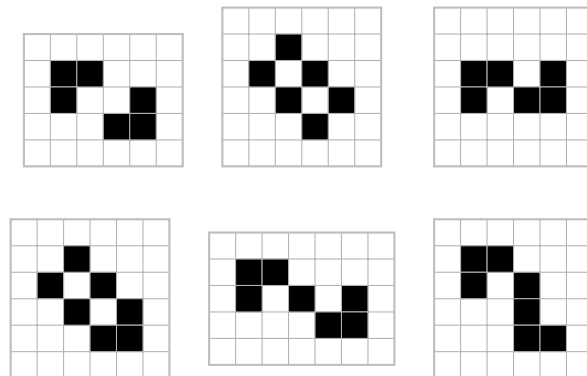
$$\text{GPF}(c) = c.$$

Medzi najznámejšie príklady patria blok (*block*), vaňa (*tub*), loď (*boat*), loďka (*ship*), včelí plást (*beehive*) a bochník (*loaf*).



Obr. 1.3: Prehľad základných stabilných vzorov. Zľava doprava: blok, vaňa, loď, loďka, včelí plást a bochník.

Okrem týchto základných vzorov existujú aj ďalšie malé stabilné vzory zobrazené na Obrázku 1.4, ako napríklad lietadlová loď (*aircraft carrier*), bárka (*barge*), had (*snake*), dlhá loď (*long boat*), dlhý had (*long snake*) a požierač (*Eater 1*).

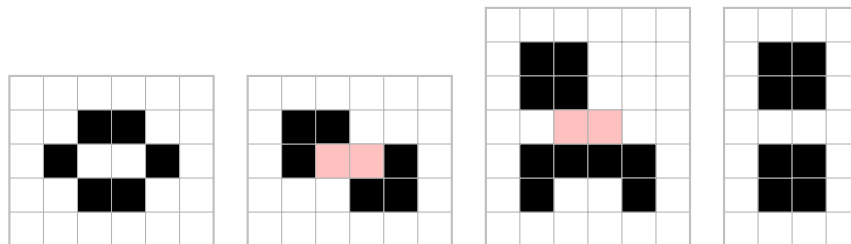


Obr. 1.4: Ďalšie malé stabilné vzory. Hore zľava: lietadlová loď, bárka, had. Dole zľava: dlhá loď, dlhý had a požierač.

## 1.4 Prísne stabilné vzory

Na Obrázku 1.3 a na Obrázku 1.4 sme videli niekoľko známych stabilných vzorov. Pri snahe zostaviť podobný úplný zoznam pre väčšie vzory sa však objavuje zaujímavý problém: na rozličné miesta mriežky možno umiestniť viacero disjunktných kópií už známych stabilných vzorov. Napríklad aj dva navzájom vzdialené bloky vytvárajú nekonečne veľkú množinu triviálnych 8-bunkových stabilných vzorov.

Vzory, ktoré vznikajú iba zoskupením iných stabilných vzorov, sa od svojich častí v podstate nelíšia, a preto sa pri katalogizácii spravidla neuvažujú. Z tohto dôvodu Cook [4] zaviedol pojem prísny stabilný vzor (*strict still life*), ktorým sa označujú stabilné vzory, ktoré sú buď spojené (súvislé), alebo ich jednotlivé komponenty nemožno oddeliť na menšie samostatné stabilné vzory. Dôvodom, prečo sa nezameriavame iba na spojené vzory, je, že niektoré stabilné vzory, napríklad lietadlová loď, sú síce nesúvislé, ale napriek tomu netriviálne – ak ich časti oddelíme, prestanú byť stabilné.



Obr. 1.5: Zľava doprava: včelí plást (*beehive*), lietadlová loď (*aircraft carrier*), *block on table*, bi-blok (*bi-block*).

Včelí plást je prísny stabilný vzor, pretože je spojený. Lietadlová loď je prísny stabilný vzor, pretože jej dve časti samostatne netvorí stabilné vzory, ale spolu spôsobujú zánik buniek zvýraznených ružovou farbou. Vzor *block on table* je prísny stabilný vzor z rovnakého dôvodu. Bi-blok nie je prísny stabilný vzor, pretože sa skladá z dvoch jednoduchších stabilných vzorov.

## 1.5 Oscilátory

Na rozdiel od stabilných vzorov, ktoré nemenia svoj stav, oscilátory predstavujú vzory, ktoré sa síce vyvíjajú, no po konečnom počte generácií  $p$  sa vrátia do svojho pôvodného stavu. Triviálne by sme mohli stabilné vzory považovať za oscilátory s periódou 1, avšak v kontexte Hry života sa tento termín používa iba pre vzory s  $p \geq 2$ .

**Definícia 1.12** (Oscilátor). Vzor  $c$  nazývame *oscilátorom*, ak existuje prirodzené číslo  $p \geq 2$  také, že  $\text{GPF}^p(c) = c$  a zároveň  $\text{GPF}^k(c) \neq c$  pre žiadne  $k \in \mathbb{N}$  také, že  $0 < k < p$ . Číslo  $p$  nazývame *periódou* oscilátora.

Najjednoduchším a najznámejším oscilátorom je blikač (*blinker*). Skladá sa z troch živých buniek v rade. V každej generácii sa strieda medzi horizontálnou a vertikálnou orientáciou, má teda periódu 2.

Aby sme formálne dokázali periodické správanie tohto oscilátora, využijeme princíp matematickej indukcie.

**Tvrdenie 1.1** (Periodita blikača). Nech  $c_H, c_V$  sú konfigurácie definované predpismi

$$c_H(x) = \begin{cases} 1, & \text{ak } x \in \{\langle -1, 0 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle\}, \\ 0, & \text{inak,} \end{cases}$$

$$c_V(x) = \begin{cases} 1, & \text{ak } x \in \{\langle 0, -1 \rangle, \langle 0, 0 \rangle, \langle 0, 1 \rangle\}, \\ 0, & \text{inak.} \end{cases}$$

Ak evolúcia systému začína v konfigurácii  $c_H$ , potom pre každé  $t \geq 0$  platí

$$\text{GPF}^t(c_H) = \begin{cases} c_H, & \text{ak } t \text{ je párne,} \\ c_V, & \text{ak } t \text{ je nepárne.} \end{cases}$$

*Dôkaz.* Dôkaz vykonáme indukciami podľa  $t$ .

1. Pre  $t = 0$  je tvrdenie triviálne, pretože

$$\text{GPF}^0(c_H) = c_H.$$

2. Nech teraz  $k \geq 0$  a predpokladajme, že

$$\text{GPF}^k(c_H) = \begin{cases} c_H, & \text{ak } k \text{ je párne,} \\ c_V, & \text{ak } k \text{ je nepárne.} \end{cases}$$

Ukážeme, že potom platí aj

$$\text{GPF}^{k+1}(c_H) = \begin{cases} c_H, & \text{ak } k + 1 \text{ je párne,} \\ c_V, & \text{ak } k + 1 \text{ je nepárne.} \end{cases}$$

Na tento účel najprv ukážeme, že

$$\text{GPF}(c_H) = c_V \quad \text{a} \quad \text{GPF}(c_V) = c_H.$$

Označme

$$P_H = \{\langle -1, 0 \rangle, \langle 0, 0 \rangle, \langle 1, 0 \rangle\}, \quad P_V = \{\langle 0, -1 \rangle, \langle 0, 0 \rangle, \langle 0, 1 \rangle\},$$

teda množiny živých buniek konfigurácií  $c_H$  a  $c_V$ . Ďalej označme

$$N_M(P_H) = \{\langle x, y \rangle \in \mathbb{Z}^2 : \exists a, b \in \mathbb{Z}, \langle a, b \rangle \in P_H, \max(|x - a|, |y - b|) \leq 1\}.$$

Rozlíšime tri prípady.

(a) Ak  $x \notin N_M(P_H)$ , potom  $c_H(x) = 0$  a  $\text{SúčetSusedov}(c_H, x) = 0$ , takže

$$(\text{GPF}(c_H))(x) = 0.$$

(b) Ak  $x \in P_H$ , potom

$$\text{SúčetSusedov}(c_H, \langle 0, 0 \rangle) = 2,$$

$$\text{SúčetSusedov}(c_H, \langle -1, 0 \rangle) = \text{SúčetSusedov}(c_H, \langle 1, 0 \rangle) = 1.$$

Teda pre  $x \in P_H$  platí

$$(\text{GPF}(c_H))(x) = \begin{cases} 1, & \text{ak } x = \langle 0, 0 \rangle, \\ 0, & \text{ak } x \in \{\langle -1, 0 \rangle, \langle 1, 0 \rangle\}. \end{cases}$$

(c) Nech  $x \in N_M(P_H) \setminus P_H$ . Potom pre bunky  $\langle 0, -1 \rangle$  a  $\langle 0, 1 \rangle$  platí

$$\text{SúčetSusedov}(c_H, \langle 0, -1 \rangle) = \text{SúčetSusedov}(c_H, \langle 0, 1 \rangle) = 3.$$

Keďže ide o mŕtve bunky a hodnota 3 spĺňa podmienku zrodu, dostávame

$$(\text{GPF}(c_H))(\langle x, y \rangle) = \begin{cases} 1, & \text{ak } \langle x, y \rangle \in \{\langle 0, -1 \rangle, \langle 0, 1 \rangle\}, \\ 0, & \text{inak.} \end{cases}$$

Teda jedinými živými bunkami konfigurácie  $\text{GPF}(c_H)$  sú práve bunky z  $P_V$ , a preto

$$\text{GPF}(c_H) = c_V.$$

Analogickým postupom dostávame aj

$$\text{GPF}(c_V) = c_H.$$

Teraz rozlíšime dva prípady.

(a) Ak  $k$  je párne, potom podľa indukčného predpokladu  $\text{GPF}^k(c_H) = c_H$ , a teda

$$\text{GPF}^{k+1}(c_H) = \text{GPF}(\text{GPF}^k(c_H)) = \text{GPF}(c_H) = c_V.$$

Keďže  $k + 1$  je nepárne, tvrdenie v tomto prípade platí.

(b) Ak  $k$  je nepárne, potom podľa indukčného predpokladu  $\text{GPF}^k(c_H) = c_V$ , a teda

$$\text{GPF}^{k+1}(c_H) = \text{GPF}(\text{GPF}^k(c_H)) = \text{GPF}(c_V) = c_H.$$

Keďže  $k + 1$  je párne, tvrdenie v tomto prípade platí.

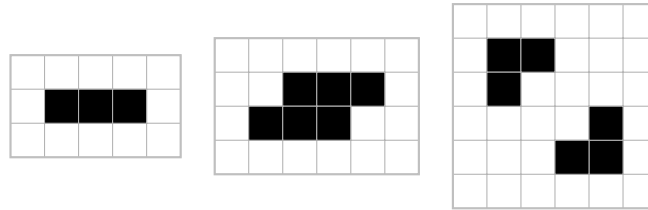
Tým je indukčný krok dokázaný.

Zároveň z dokázaných rovností máme  $\text{GPF}(c_H) = c_V \neq c_H$  a

$$\text{GPF}^2(c_H) = \text{GPF}(\text{GPF}(c_H)) = \text{GPF}(c_V) = c_H.$$

Teda konfigurácia  $c_H$  je oscilátor s periódou 2. □

Ďalšími bežnými oscilátormi s periódou 2 sú ropucha (*toad*) a maják (*beacon*).



Obr. 1.6: Základné oscilátory s periódou 2: blikač, ropucha a maják.

Z teoretického hľadiska je dôležitým výsledkom, že v Hre života existujú oscilátory pre každú celočíselnú periódu  $p \geq 2$ . Tento dlhoročný problém, známy ako *omnipericidita*, bol definitívne vyriešený až v roku 2023 objavením oscilátorov pre posledné chýbajúce periódy 19 a 41 [5]. Zatiaľ čo oscilátory s malými periódami (napr. 2, 3, 4, 5, 6, 8, 14, 15, 30) sa často vyskytujú prirodzene v náhodných poliach, pre iné periódy sú oscilátory zriedkavé a museli byť umelo skonštruované.

## 1.6 Lode a posuvné vzory

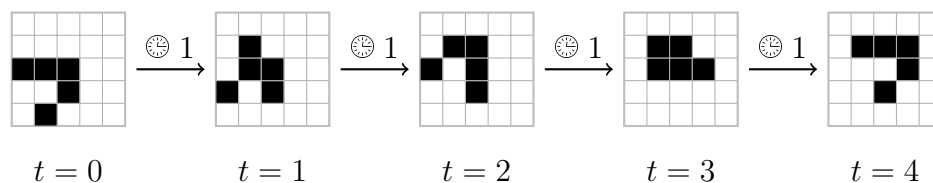
Špeciálnou kategóriou vzorov sú tie, ktoré sa po  $p$  generáciách obnovia do pôvodného tvaru, ale na posunutej pozícii. Tieto vzory sa nazývajú lode (angl. *spaceships*).

**Definícia 1.13** (Lod). Vzor  $c$  nazývame *lodou*, ak existuje prirodzené číslo  $p \geq 1$  (perióda) a nenulový vektor posunu  $\vec{v} \in \mathbb{Z}^2$  taký, že pre každé  $x \in \mathbb{Z}^2$  platí:

$$(\text{GPF}^p(c))(x) = c(x - \vec{v}).$$

Rýchlosť lode sa zvyčajne udáva ako zlomok „rýchlosti svetla“  $c$ . Symbolom  $c$  označujeme maximálnu rýchlosť šírenia informácie v automate: za jednu generáciu môže vplyv bunky zasiahnuť nanajvýš bunky v jej bezprostrednom susedstve, teda najviac o jeden dielik ďalej. Zápis  $c/n$  preto znamená, že vzor sa posunie o jednu bunku za  $n$  generácií, zatiaľ čo zápis  $kc/n$  znamená posun o  $k$  buniek za  $n$  generácií v danom smere.

Najznámejšou a najčastejšie sa vyskytujúcou lodou je klzák (*glider*). Tento malý vzor pozostávajúci len z piatich buniek objavil Richard K. Guy krátko po vzniku Hry života. Klzák sa periodicky opakuje po 4 generáciách, pričom sa posunie o jednu bunku diagonálne. Jeho rýchlosť je teda  $c/4$  diagonálne. Klzák je kľúčovým prvkom v Hre života, pretože umožňuje prenos informácie na veľké vzdialenosti. Jeho objav bol prvým náznakom, že Hra života môže obsahovať štruktúry schopné univerzálneho výpočtu.

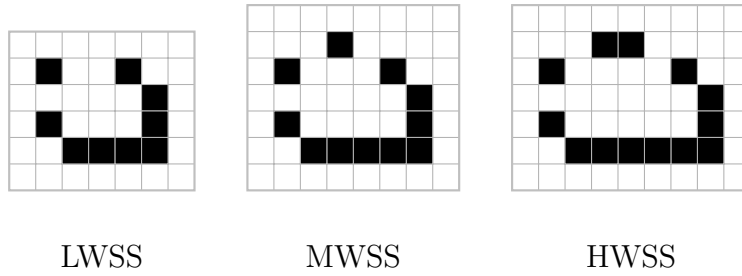


Obr. 1.7: Evolúcia klzáka v piatich po sebe idúcich generáciách. Všimnime si, že v čase  $t = 4$  je vzor totožný so vzorom v  $t = 0$ , ale posunutý o jeden dielik diagonálne doprava a hore.

Okrem diagonálnych vzorov poznáme aj *ortogonálne lode*, ktoré sa po mriežke pohybujú v horizontálnom alebo vertikálnom smere. Medzi prvé objavené objekty tohto druhu patria vzory s rýchlosťou  $c/2$  (posun o jedno políčko každé dve generácie). Túto základnú skupinu tvoria tri štrukturálne príbuzné lode:

- **ľahká loď** (*Lightweight spaceship* – LWSS),
- **stredná loď** (*Middleweight spaceship* – MWSS),
- **ťažká loď** (*Heavyweight spaceship* – HWSS).

Všetky tri spomínané lode majú periódu 4, čo znamená, že po štyroch generáciách nadobudnú svoj pôvodný tvar, pričom sú v danom smere posunuté o dve políčka.



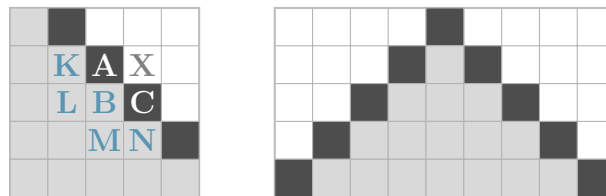
Obr. 1.8: Najznámejšie ortogonálne lode s rýchlosťou  $c/2$ .

Otázka existencie lodí s odlišnými rýchlosťami a smermi zostávala dlhý čas otvorená. Po mnoho rokov boli známe iba lode pohybujúce sa diagonálne rýchlosťou  $c/4$  a ortogonálne rýchlosťou  $c/2$ . Prelom nastal v roku 1989, keď Dean Hickerson objavil ortogonálnu loď  $c/3$  a neskôr aj prvú známu diagonálnu loď odlišnú od klzáka s periódou 4 (tzv. veľký klzák, *big glider*). Ďalšie objavy rozšírili známy súbor lodí o vzory s rýchlosťami ako  $c/5$ ,  $2c/5$  či  $c/12$ .

Táto variabilita prirodzene vyvoláva otázku hornej hranice rýchlosti. Môže sa informácia v Hre života šíriť ľubovoľne rýchlo, alebo je táto rýchlosť obmedzená? Ukazuje sa, že aj v Hre života existujú horné limity rýchlosti šírenia vzorov.

**Tvrdenie 1.2** (Rýchlostné limity lodí). Maximálna diagonálna a ortogonálna rýchlosť, akou sa môže vzor pohybovať prázdny priestorom, je  $c/4$ , resp.  $c/2$ .

*Dôkaz podľa [2].* Začneme dôkazom limitu  $c/4$  pre diagonálne lode. Predpokladajme (za účelom dôkazu sporom), že bunka  $X$  je živá v generácii 2.



Ak je bunka  $X$  živá v generácii 2, potom bunky  $A$ ,  $B$  a  $C$  musia byť živé v generácii 1. Z toho vyplýva, že bunky  $A$  a  $C$  mali po troch živých susedov v generácii 0, teda bunky  $K$ ,  $L$ ,  $B$ ,  $M$ ,  $N$  museli byť v generácii 0 živé.

To však implikuje, že bunka  $B$  mala v generácii 0 aspoň štyroch živých susedov, takže neexistuje spôsob, ako by mohla prežiť do generácie 1. To vedie k sporu.

Ukázali sme, že bunka  $X$  nemôže byť živá v generácii 2. Inými slovami, ak je loď obsiahnutá v určitej oblasti v generácii 0, v generácii 2 sa musí nachádzať na a pod diagonálnou líniou definovanou týmito bunkami. Aplikovaním tohto argumentu znova

vidíme, že loď musí byť na a pod diagonálnou líniou obsahujúcou bunku  $X$  v generácii 4. Z toho vyplýva, že žiadna loď nemôže cestovať diagonálne rýchlejšie ako  $c/4$ .

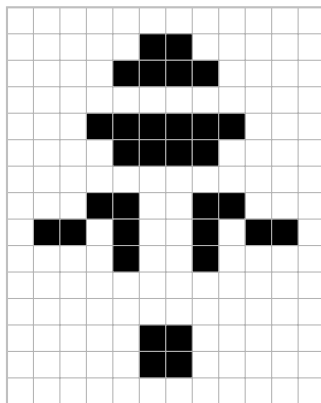
Dôkaz, že rýchlostný limit  $c/2$  platí pre ortogonálne lode, využíva dve takéto diagonálne línie. Ak je loď obsiahnutá v určitej oblasti v generácii 0, potom sme už ukázali, že v generácii 2 musí byť ohraničená diagonálnymi „priamkami“. Z toho vyplýva, že maximálna ortogonálna rýchlosť je  $c/2$ .  $\square$

**Definícia 1.14** (Typ lode). Ak sa loď za  $n$  generácií posunie o  $p$  buniek horizontálne a o  $q$  buniek vertikálne, hovoríme, že je typu  $\langle p, q \rangle/n$ . Jej rýchlosť potom zapisujeme ako  $\langle p, q \rangle c/n$ .

Ďalšou významnou kategóriou sú lode s posunom „jazdca“ alebo „koňa“ (*knightships*). Definujú sa ako lode typu  $\langle 2m, m \rangle/n$ , čo znamená, že sa posunú o dve bunky horizontálne na každú jednu bunku vertikálne (pohyb šachového jazdca). Všeobecnejšie sa tento termín používa pre akékoľvek „šikmé“ lode. Je známe, že takéto vzory musia byť asymetrické a mať periódu aspoň 6.

Prvým elementárnym jazdcom v Hre života sa stal *Sir Robin*, ktorého objavil Adam P. Goucher 6. marca 2018 pomocou vyhľadávacieho programu `ikpx` [6]. Loď nesie systematický názov `282P6H2V1`, skladá sa z 282 buniek a pohybuje sa rýchlosťou  $\langle 2, 1 \rangle c/6$ . Jej konštrukcia bola založená na čiastočnom vzore, ktorý predtým našiel Tomas Rokicki. *Sir Robin* bol dlho jediným známym elementárnym knightshipom, až kým nebol objavený tzv. *sprayer*.

Výskum v tejto oblasti je stále aktívny. V roku 2016 bola objavená loď *Copperhead* s rýchlosťou  $c/10$ , znázornená na Obrázku 1.9, čo ukázalo, že aj medzi pomerne malými a jednoduchými vzormi mohli dlhodobo zostať neidentifikované vzory [7].

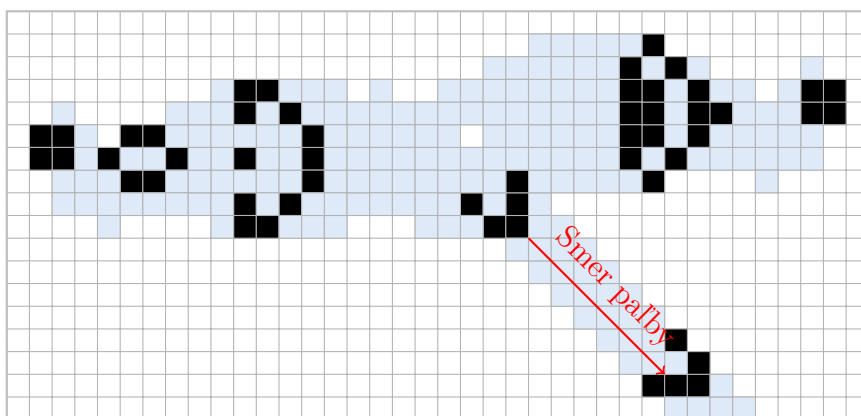


Obr. 1.9: Loď *Copperhead*, ortogonálny vzor v Hre života s rýchlosťou  $c/10$ .

## 1.7 Generátory klzákov

Zásadným míľnikom bol objav tzv. generátora klzákov (*glider gun*) v roku 1970, ktorý uskutočnil Bill Gosper. Išlo o prvú známu konfiguráciu s neobmedzeným rastom, ktorá periodicky produkuje nekonečnú sekvenciu klzákov. Tento objav definitívne ukázal, že Hra života nie je len systémom smerujúcim ku stabilite alebo zániku, ale umožňuje aj otvorenú dynamiku [3].

Najznámejším predstaviteľom tejto triedy je generátor klzákov *Gosper Glider Gun*. Jeho mechanizmus je založený na interakcii dvoch oscilujúcich štruktúr, ktoré sa pohybujú proti sebe. Pri ich vzájomnej zrážke v presne určenom okamihu dôjde k vygenerovaniu nového klzáka, ktorý opustí systém, zatiaľ čo zvyšok konštrukcie sa vráti do pôvodného stavu a cyklus sa opakuje.



Obr. 1.10: Generátor klzákov *Gosper Glider Gun*: prvý objavený vzor, ktorý periodicky vysiela klzáky.

Keďže klzák v Hre života sa pohybuje diagonálne, aj tento generátor môže po vhodnom otočení alebo zrkadlení vysielať klzáky len do štyroch diagonálnych smerov, teda pod uhlami  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$  a  $315^\circ$  vzhľadom na vodorovnú os mriežky.

## 1.8 Rajské záhrady

Zaujímavý pohľad na Hru života získame, ak namiesto sledovania vývoja vzorov v čase uvažujeme opačný smer. Namiesto otázky, na čo sa daný vzor vyvinie, sa pýtame, z akých konfigurácií mohol vzniknúť. Inými slovami, skúmame jeho rodičov [2].

Niektoré vzory, ako napríklad blok, majú množstvo rodičov. To je jeden z dôvodov, prečo sa tak často objavujú v náhodných počiatkových poliach<sup>2</sup>. Iné vzory, ako napríklad

<sup>2</sup>V terminológii Hry života sa takéto náhodné počiatkové konfigurácie často označujú ako *soup* a po ich vývoji zvyčajne zostane stabilný alebo periodický „popol“ (*ash*).

hodiny (*clock*), majú rodičov relatívne málo. Ich vznik z náhodných konfigurácií je preto menej pravdepodobný, čo sa prejavuje aj ich zriedkavejším výskytom.

Prírodnou otázkou je, či existujú aj také vzory, ktoré nemajú žiadneho rodiča. Ide o konfiguráciu, ktorá nemôže vzniknúť evolúciou z iného vzoru, ale môže sa objaviť iba ako počiatočný stav systému. Takýto vzor sa nazýva rajská záhrada (angl. *Garden of Eden*)<sup>3</sup>.

Intuitívny dôvod, prečo musia rajské záhrady existovať, spočíva v tom, že niektoré vzory majú veľa rodičov. Prechodová funkcia Hry života preto nie je surjektívna: viacerô rôznych konfigurácií môže viesť k rovnakému výsledku, a tým sa „vyčerpáva“ priestor možných predchodcov. V dôsledku toho existujú konfigurácie, ku ktorým nevedie žiadny predchodca. Toto tvrdenie môžeme formálne dokázať.

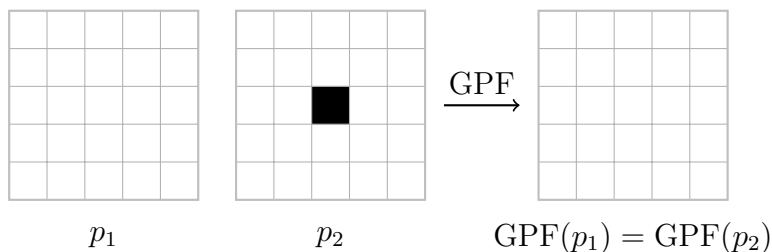
**Definícia 1.15** (Vzájomne vymazateľné lokálne vzory vzhľadom na oblasť  $E$ ). Nech  $E \subseteq \mathbb{Z}^2$  je konečná množina. Lokálne vzory  $p_1, p_2: E \rightarrow \{0, 1\}$  nazývame *vzájomne vymazateľné vzhľadom na oblasť  $E$* , ak  $p_1 \neq p_2$  a pre každé konfigurácie  $c_1, c_2$ , ktoré spĺňajú

- $c_1(x) = p_1(x)$  a  $c_2(x) = p_2(x)$  pre všetky  $x \in E$ ,
- $c_1(x) = c_2(x)$  pre všetky  $x \in \mathbb{Z}^2 \setminus E$ ,

platí, že  $\text{GPF}(c_1) = \text{GPF}(c_2)$ .

**Tvrdenie 1.3** (Existencia rajských záhrad). V Hre života existujú rajské záhrady.

*Dôkaz.* Dôkaz sa opiera o klasický výsledok v teórii celulárnych automatov známy ako *Mooreova–Myhillova veta*, podľa ktorej existuje v automate konfigurácia typu rajská záhrada práve vtedy, keď automat obsahuje lokálne vzájomne vymazateľné vzory vzhľadom na nejakú oblasť  $E$  [8, 9].



Obr. 1.11: Ilustrácia lokálnych vzorov vzájomne vymazateľných vzhľadom na oblasť  $E$ . Konfigurácie  $p_1$  a  $p_2$  sa v jednom kroku vyvinú do totožnej konfigurácie bez ohľadu na vonkajšie prostredie.

<sup>3</sup>Iný možný preklad je aj raj alebo Záhrada Eden. Tento termín zaviedol v kontexte celulárnych automatov John W. Tukey dávno pred vznikom Conwayovej Hry života.

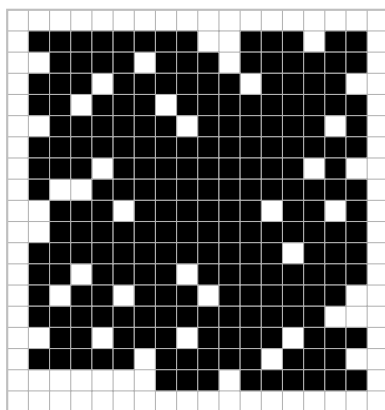
Uvažujme oblasť  $E \subseteq \mathbb{Z}^2$  o veľkosti  $5 \times 5$  buniek. Nech  $p_1$  je lokálny vzor, v ktorom sú všetky bunky v tejto oblasti mŕtve, a  $p_2$  je lokálny vzor, v ktorom je živá iba stredová bunka a zvyšných 24 je mŕtvych, ako je znázornené na Obrázku 1.11.

Rozdiel medzi týmito dvoma vzormi tvorí iba stredová bunka. Tá však vo vzore  $p_2$  nemá vo svojom okolí žiadnu inú živú bunku, preto v ďalšom kroku zomrie. Bunky bezprostredne susediace so stredom (vnútorný štvorec  $3 \times 3$ ) „vidia“ len jedného živého suseda, čo na oživenie nestačí. Zároveň sú izolované od vplyvu akýchkoľvek buniek mimo oblasti  $E$ . Bunky na samom okraji oblasti  $E$  zasa „nevidia“ na stredovú bunku, keďže je od nich vzdialená viac ako jednu bunku.

Stav stredovej bunky teda nedokáže nijako ovplyvniť budúcu konfiguráciu. Obe rôzne lokálne konfigurácie sa v jednom kroku vyvinú do absolútne identického stavu. Tým sme našli lokálne vzory vzájomne vymazateľné vzhľadom na oblasť  $E$  a z Mooreovej–Myhillovej vety bezprostredne vyplýva, že v Conwayovej Hre života musí existovať rajska záhrada.  $\square$

Dôkaz Tvrdenia 1.3 je síce matematicky korektný, no má jeden praktický nedostatok: je nekonštruktívny. Nehovorí nič o tom, ako konkrétnu rajskú záhradu nájsť. Aby sme našli konkrétny príklad v rozumnej veľkosti, musíme zvoliť iný prístup. Osvedčenou metódou je iteratívne budovanie vzoru pomocou heuristického algoritmu [2]:

1. Začneme s jednou bunkou. Analyzujeme, koľko predchodcov (rodičov) by mal vzor, ak by bola táto bunka živá, a koľko, ak by bola mŕtva.
2. Vyberieme ten stav, ktorý vedie k *menšiemu* počtu možných rodičov.
3. Postupne pridávame susedné bunky (zvyčajne v špirále okolo stredu) a v každom kroku zafixujeme stav novej bunky tak, aby sme minimalizovali počet zostávajúcich možných predchodcov celého vzoru.



Obr. 1.12: Príklad rajskej záhrady. Tento vzor nemá žiadneho predchodcu.

Tento *greedy* prístup spočiatku vedie k nárastu počtu rodičov, no po prekročení určitej hranice začne ich počet klesať. V prípade vzoru na Obrázku 1.12 stačilo týmto spôsobom zdefinovať 266 buniek, kým počet možných rodičov klesol na nulu [2]. Výsledkom je konfigurácia, ktorá nemôže vzniknúť zo žiadneho predchádzajúceho stavu – hľadaná rajská záhrada.

## 1.9 Výpočtová univerzálnosť

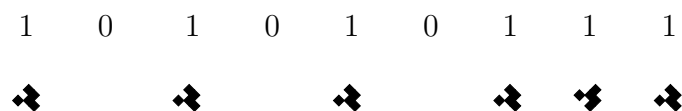
Hra života však presahuje rámec vizuálne zaujímavej simulácie, keďže predstavuje systém s plnou výpočtovou silou. Dôkaz tejto výpočtovej univerzálnosti predložil John Conway so svojím tímom v 70. rokoch. Navrhli konštrukcie logických hradieľ a ďalších komponentov, ktoré teoreticky umožňujú zostrojiť univerzálny Turingov stroj. Tieto koncepty boli publikované v [3, 10]. Dôsledkom Turingovskej úplnosti je algoritmická nerozhodnuteľnosť mnohých otázok týkajúcich sa správania systému.

Keďže realizácia výpočtov v Hre života si vyžaduje mechanizmy na prenos, generovanie a spracovanie informácií, ďalej stručne naznačíme, ako možno z takýchto prvkov poskladať jednoduchú výpočtovú konštrukciu.

### 1.9.1 Generátor signálu

Prvým kľúčovým prvkom je zdroj signálu, ktorý funguje ako taktovací generátor. V Hre života túto funkciu plní vzor *Gosper Glider Gun*, zobrazený na Obrázku 1.10, ktorý sme podrobnejšie opísali v sekcii 1.7. Periodický prúd klzákov, ktorý tento vzor produkuje, slúži ako nosič informácie (konštantná logická 1).

Informácia je v tomto prúde kódovaná binárne: prítomnosť klzáka reprezentuje bit 1, zatiaľ čo jeho absencia bit 0. Prúd klzákov tak plní funkciu analogickú elektrickému signálu v obvodoch, ktorý je následne modulovaný ďalšími prvkami obvodu.

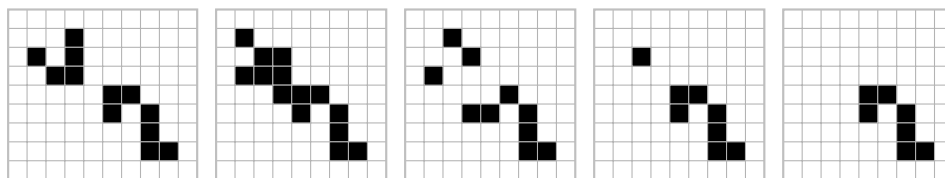


Obr. 1.13: Reprezentácia binárnej informácie v Hre života. Prítomnosť klzáka reprezentuje bit 1, jeho absencia bit 0.

## 1.9.2 Požierač signálu

Mnohé zložité vzory, ktoré sa využívajú pri konštrukcii výpočtových obvodov, sú založené na presúvaní klzákov z jedného miesta na druhé. Preto potrebujeme jednoduché spôsoby, ako klzáky vytvárať, presúvať a odstraňovať. Odstraňovanie klzákov je z týchto úloh najjednoduchšie a realizuje sa pomocou vzorov nazývaných požierače (*eaters*). Sú to stabilné vzory, ktoré pri vhodnej zrážke pohltia klzák bez trvalého poškodenia a následne sa vrátia do pôvodnej podoby.

Najmenší, prvý objavený a najpoužívanější z týchto požieračov je 7-bunkový stabilný vzor *Eater 1*. Jeho hlavnou vlastnosťou je schopnosť pohltiť prilietajúci klzák, ako je znázornené na Obrázku 1.14. Dôvodom jeho širokého využitia nie je len jeho malá veľkosť, ale aj rýchly čas zotavenia: do pôvodného stavu sa vráti už po 4 generáciách od zrážky, čo je najrýchlejší možný čas pre akýkoľvek požierač klzákov.

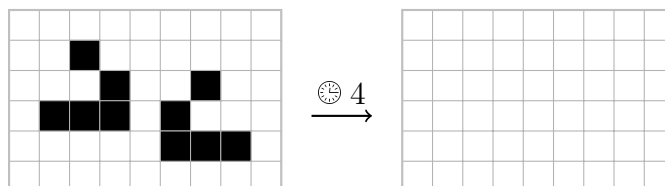


Obr. 1.14: Priebeh kolízie, pri ktorej klzák zaniká a požierač sa následne vráti do pôvodného stabilného stavu.

## 1.9.3 Logika zrážok a pamäť

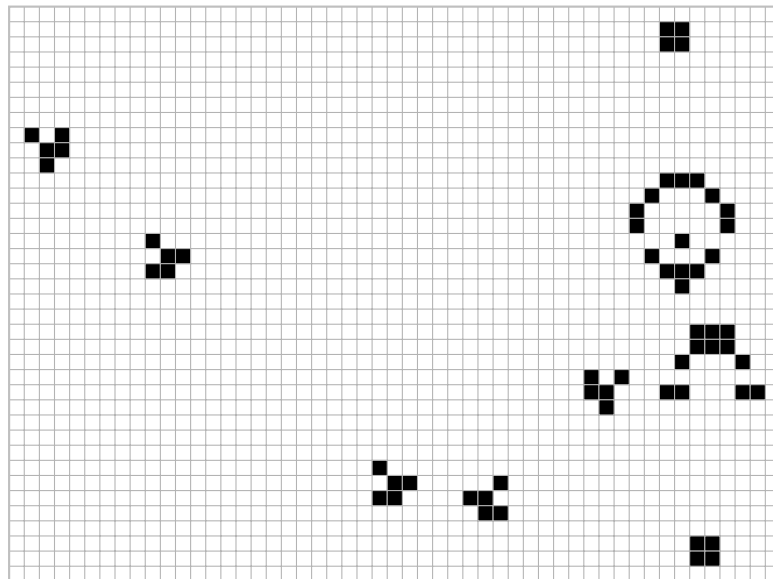
Samotný Turingov stroj vzniká kombináciou vyššie opísaných prvkov. Logické operácie sú realizované pomocou vhodne navrhnutých a časovo zosúladených interakcií medzi klzákami.

Pre realizáciu hradla NOT využívame reakciu vzájomného zničenia (angl. *mutual annihilation*). Ide o kolíziu dvoch klzákov, pri ktorej nedochádza k vzniku žiadneho ďalšieho vzoru a oba klzáky zaniknú.



Obr. 1.15: Detail zanikajúcej reakcie. Dva klzáky (vľavo) sa pri strete navzájom anihilujú (vpravo), čo predstavuje logickú nulu na výstupe hradla.

Samotné hradlo NOT, znázornené na Obrázku 1.16, je skonštruované tak, že využíva nepretržitý prúd klzákov generovaný pomocou generátora klzákov *Gosper Glider Gun*, ktorý slúži ako zdroj logickej 1. Vstupný signál prichádza zľava. Ak vstupný prúd obsahuje klzák (logická 1), tento sa zrazí s prúdom z generátora, čím dôjde k vzájomnému zničeniu a výstup ostane prázdny (logická 0). Ak však vstupný klzák chýba (logická 0), k zrážke nedôjde a klzák z generátora pokračuje bez prekážky vľavo nadol, čím vytvára výstupný signál (logická 1).



Obr. 1.16: Princíp fungovania hradla NOT.

Na rovnakom princípe riadených kolízií možno realizovať aj logické hradlá **AND** a **OR** [3, 11].

Okrem logických operácií je nevyhnutná aj **pamäť**. V Hre života sa realizuje najmä dvoma hlavnými spôsobmi:

- **Dynamická pamäť (*delay-line memory*)**. Informácia je reprezentovaná klzákami, ktoré neustále cirkulujú v uzavretej slučke, zvyčajne vytvorenej pomocou odrážačov. Prítomnosť klzáka v tejto slučke reprezentuje bit 1, zatiaľ čo jeho neprítomnosť zodpovedá bitu 0.
- **Statická pamäť (*auxiliary storage*)**. Systém ukladá hodnoty prostredníctvom polohy stabilného vzoru, najčastejšie bloku, na mriežke. Vzdialenosť tohto vzoru od referenčného bodu kóduje uloženú hodnotu. Obsah registra sa mení (napríklad inkrementáciou alebo dekrementáciou) vysielaním vhodne usporiadaných flotíl klzákov, ktoré daný vzor posunú o presne určenú vzdialenosť.

Kombináciou týchto techník – generátorov, požieračov, logických hradiel a pamäťových registrov – je možné v mriežke Hry života postaviť akýkoľvek digitálny obvod, a teda aj univerzálny Turingov stroj.

Neskôr sa objavili formálnejšie a explicitnejšie dôkazy. Durand a Róka v roku 1999 publikovali zjednodušený a formálne precíznejší dôkaz univerzálnosti [12]. Vo svojej práci dokázali univerzálnosť Hry života vzhľadom na niekoľko výpočtových modelov: booleovské obvody, Turingove stroje a dvojrozmerné celulárne automaty. Tieto rôzne uhly pohľadu zvolili s cieľom objasniť situáciu a zjednodušiť pôvodný dôkaz, pričom presne definovali tieto tri vlastnosti univerzálnosti a vysvetlili vzťahy medzi nimi. Konštruktívny prístup demonštroval Paul Rendell, ktorý postupne v rokoch 2002 až 2015 zostrojil explicitnú implementáciu univerzálneho Turingovho stroja priamo v Hre života [13, 14, 15]. Jeho práca tak poskytla vizuálny a mechanicky konkrétny dôkaz Turingovej univerzálnosti.

Historický vývoj Hry života tak ilustruje, ako jednoduchý model pôvodne zamýšľaný ako matematická kuriozita môže viesť k významným výsledkom na rozhraní dynamických systémov, teórie výpočtov a komplexných systémov.

# Kapitola 2

## Prehľad existujúcich prístupov

Hoci je klasická Conwayova Hra života najznámejším príkladom celulárneho automatu, predstavuje len jeden bod v rozsiahlom priestore možných pravidiel a modelov. Výskum sa už od 70. rokov 20. storočia uberal smermi, ktoré modifikovali jej základné charakteristiky: diskrétny stavový priestor, deterministické pravidlá a fixné okolie. Táto kapitola poskytuje prehľad kľúčových prístupov, ktoré sú relevantné pre problematiku skúmanú v tejto práci.

### 2.1 Generácie Hry života

Rodina pravidiel definovaných na Moorovom okolí s dvoma stavmi je obrovská a označuje sa ako rodina *Life-like* celulárnych automatov. Systematické skúmanie odhalilo široké spektrum dynamických režimov, ktoré sa často výrazne líšia od klasickej Hry života. Zatiaľ čo klasická hra podporuje vznik stabilných aj pohyblivých vzorov, iné pravidlá sa vyznačujú vyššou mierou stability, nepravidelnosti alebo geometrickej regularity.

Tieto pravidlá a vzory, ktoré generujú, sú systematicky klasifikované v rozsiahlych katalógoch [16]. Medzi najvýznamnejšie objavené varianty patria:

- **HighLife (B36/S23)**. Podobné klasickej Hre života, s dodatočnou podmienkou zrodu pri 6 susedoch. Umožňuje existenciu *replikátora*, teda vzoru, ktorý sa periodicky kopíruje a vedie k exponenciálnemu rastu.
- **Day & Night (B3678/S34678)**. Toto pravidlo je charakteristické svojou symetriou. Ak invertujeme stavy všetkých buniek, správanie systému sa nezmení. Podporuje zložité vzory a možno v ňom tiež realizovať logické obvody ako v Hre života.
- **Seeds (B2/S)**. Jednoduché pravidlo, kde každá živá bunka okamžite zomiera (žiadne prežitie), no nová sa narodí, ak má presne 2 susedov. Napriek tomu, že toto pravidlo

neumožňuje existenciu stabilných vzorov ani bežných oscilátorov, vedie k dynamike charakterizovanej rýchlym rastom a vznikom fraktálnych štruktúr.

- **Morley (B368/S245)**. Pravidlo známe vysokým výskytom lodí a tzv. *pufferov*, teda lodí, ktoré za sebou zanechávajú stopu z iných vzorov (dym).
- **Life without Death (B3/S012345678)**. Tiež známe ako *Labyrinth*. V tomto pravidle bunky nikdy neumierajú. Z náhodného počiatočného stavu sa rýchlo vyvinú zložité, bludiskám podobné štruktúry, ktoré sa postupne stabilizujú.

Okrem týchto známych variantov existuje množstvo ďalších pravidiel, ktoré vykazujú vlastnosti, aké v klasickej Hre života nenájdem. Napríklad pravidlo **Replicator (B1357/S1357)** je špecifické tým, že v ňom každý vzor funguje ako replikátor a vytvára svoje kópie. Iné pravidlá modelujú fyzikálne či sociálne javy – napríklad **Vote (B5678/S45678)**, známe aj ako *Majority*, simuluje proces majoritného hlasovania, kde sa oblasti stabilizujú do homogénnych blokov. Zaujímavé je aj pravidlo **Walled Cities (B45678/S2345)**, ktoré vytvára centrá pseudo-náhodnej aktivity oddelené stabilnými stenami, či **Amoeba (B357/S1358)**, ktorého vzory pripomínajú biologické organizmy.

## 2.2 Jednorozmerné celulárne automaty

Okrem dvojrozmernej Conwayovej Hry života sa skúmali aj celulárne automaty na lineárnej mriežke. K prvým významným príspevkom v tejto oblasti patrí článok Jonathana K. Millena *One-Dimensional Life*, ktorý vyšiel v časopise BYTE v roku 1978 [17]. Millen navrhol jednorozmernú verziu Hry života, ktorá sa snaží zachovať ducha Conwayových pravidiel v 1D priestore.

Tento systém využíva okolie zo štyroch buniek na pozíciách  $-2, -1, +1, +2$  (symetria  $YYXYY$ , kde  $X$  je skúmaná bunka). Nech  $N_Y$  je počet živých susedov v tomto okolí. Pravidlá sú definované takto:

- **Zrod.** Mŕtva bunka ožije, ak má 2 alebo 3 živých susedov ( $N_Y \in \{2, 3\}$ ).
- **Prežitie.** Živá bunka prežije, ak má 2 alebo 4 živých susedov ( $N_Y \in \{2, 4\}$ ).

Stephen Wolfram neskôr systematicky klasifikoval jednorozmerné automaty [18]. Millenov model v tejto klasifikácii zodpovedá vonkajšiemu totalistickému (outer totalistic) pravidlu s kódom 20 (parametre  $k = 2, r = 2$ ).

Zaujímavosťou tohto pravidla je absencia stabilných vzorov. Všetky konečné konfigurácie buď zaniknú, alebo sa vyvinú do oscilátorov či posuvných vzorov. Zároveň je známe, že v tomto systéme existujú konfigurácie typu rajská záhrada, pričom najjednoduchším známym príkladom je vzor 101 (živá-mŕtva-živá bunka).

## 2.3 Trojrozmerné celulárne automaty

Prirodzeným rozšírením dvojrozmernej mriežky je prechod do troch dimenzií. V tomto priestore má každá bunka v Moorovom okolí až 26 susedov (kocka  $3 \times 3 \times 3$  bez stredovej bunky). Hľadanie pravidiel, ktoré by v 3D priestore vykazovali podobne bohaté správanie ako Conwayova Hra života (najmä existenciu klzákov), bolo dlho otvorenou výzvou. Keďže počet možných pravidiel s rastúcim počtom susedov exponenciálne narastá, náhodné hľadanie je neefektívne.

Významný pokrok v tejto oblasti dosiahol Carter Bays, ktorý v roku 1987 publikoval článok *Candidates for the Game of Life in Three Dimensions* [19]. Bays navrhol kritériá, ktoré by mal spĺňať vhodný 3D celulárny automat: mal by podporovať vznik stabilných vzorov, oscilátorov a predovšetkým klzákov, pričom by nemal viesť k nekontrolovanému rastu populácie z jednoduchých počiatkových podmienok.

Bays zaviedol notáciu pre pravidlá v tvare štyroch parametrov  $E_b, E_h, F_b, F_h$ , kde bunka prežije, ak počet jej susedov spadá do intervalu  $\langle E_b, E_h \rangle$  (Environment), a nová sa narodí, ak počet susedov spadá do intervalu  $\langle F_b, F_h \rangle$  (Fertility).

Aby zúžil obrovský priestor možných pravidiel, Bays sformuloval a dokázal niekoľko kľúčových tvrdení:

**Tvrdenie 2.1.** Pravidlo  $E_l E_u F_l F_u$  s  $F_l \geq 10$  nemôže podporovať existenciu klzáka.

Je to zrejmé z pozorovania, že mŕtva bunka susediaca s rovinou živých buniek môže mať najviac deväť susedov, čo je zároveň horná hranica pre bunku susediacu s ľubovoľným rovinným zoskupením buniek. Preto každá konfigurácia podľa pravidla  $\langle X Y 10 Z \rangle$  napokon buď zanikne, alebo vytvorí konvexnú konfiguráciu, ktorej vonkajší povrch môže zostať dynamický, no celá konfigurácia sa nikdy nebude pohybovať priestorom.

**Tvrdenie 2.2.** Pravidlo  $E_l E_u F_l F_u$  s  $F_l \leq 4$  vedie k neobmedzenému rastu.

Systematickým skúmaním v tomto zúženom priestore identifikoval dva hlavné kandidáty. Prvým je pravidlo **Life 5766** (prežitie 5–7, zrod 6). Bays ukázal, že toto pravidlo je v istom zmysle 3D analógiou Conwayovej Hry života. Ak vezmeme 2D vzor a „vytiahneme“

ho do tretej dimenzie (vytvoríme hranol hrúbky 2 bunky), za určitých podmienok sa bude v 3D priestore správať identicky ako jeho 2D predloha.

Druhým kandidátom je pravidlo **Life 4555** (prežitie 4–5, zrod 5). Toto pravidlo vykazuje dynamiku, ktorá sa v dôležitých znakoch odlišuje od 2D hry a nemá v nej priamu analógiu. Vyskytujú sa v ňom malé, vysoko symetrické stabilné a oscilujúce formy. Práve v tomto pravidle bol objavený 3D klzák, ktorý nevzniká jednoduchým rozšírením známeho 2D vzoru, ale vzniká spontánne.

Bays sa však neobmedzil iba na túto triedu pravidiel. Diskutoval aj pokusy nájst vhodný variant Life na dvojrozmernej hexagonálnej mriežke, kde má každá bunka šesť susedov. Dospel pritom k záveru, že v tomto prostredí neexistuje pravidlo, ktoré by spĺňalo jeho kritériá pre hru typu Life, pokiaľ neuvažujeme špecifickejšie konštrukcie, ako sú Golayove obaly (*Golay surrounds*). Následne preto rozšíril hexagonálnu mriežku do troch rozmerov, čím získal tzv. hexahedrálnu teseláciu. V tomto prostredí má bunka 12 susedov usporiadaných do štyroch pretínajúcich sa hexagónov.

Aj pre tento model odvodil obmedzujúce tvrdenia: pravidlá s príliš vysokým prahom zrodu nemôžu podporovať klzáky a pravidlá s príliš nízkym prahom zrodu vedú k neobmedzenému rastu. Tým sa hľadanie opäť výrazne zúžilo a pravidlo **Life 3333** zostalo podľa Baysa jediným kandidátom, ktorý spĺňa požiadavky na trojrozmernú hru typu Life v hexahedrálnom priestore.

## 2.4 Viacstavové automaty

Ďalšou možnosťou zovšeobecnenia je rozšírenie počtu diskrétnych stavov buniek. Namiesto dvoch stavov môže bunka nadobúdať tri, štyri alebo ešte viac hodnôt, ktoré môžu reprezentovať rôzne fázy životného cyklu, typy buniek či úrovne hustoty.

Klasickým príkladom trojstavového automatu je *Brian's Brain*, ktorý navrhol Brian Silverman v polovici 80. rokov 20. storočia [20]. Tento model, v modernej terminológii klasifikovaný ako člen rodiny pravidiel *Generations* s notáciou  $B2/S/3^1$ , bol inšpirovaný šírením signálu v neurónových sieťach. Bunky sa nachádzajú v jednom z troch stavov: živá, umierajúca (refraktérna) alebo mŕtva.

Pravidlá sú deterministické a cyklické: mŕtva bunka ožije, ak má presne dvoch živých susedov, živá bunka sa v nasledujúcom kroku automaticky stane umierajúcou a umierajúca bunka sa stane mŕtvou. Táto striktná dynamika spôsobuje, že systém je vizuálne aj

---

<sup>1</sup>Zápis indikuje, že bunka sa narodí pri 2 susedoch (B2), žiadna živá bunka neprežije do ďalšieho kroku (S) a systém má 3 stavy.

experimentálne mimoriadne príťažlivý; vzory často „explodujú“ a šíria diagonálne vlnenie aktivity po celej mriežke.

Na rozdiel od Conwayovej Hry života, kde sú pohybujúce sa vzory relatívne vzácne, v modeli *Brian's Brain* takmer každá generická konfigurácia generuje množstvo komplexných pohyblivých štruktúr, tzv. raketoplány (*spaceships*) a hrable (*rakes*). Napriek tejto prevahe pohybu je možné konštruovať aj statickejšie prvky.

Vďaka tejto bohatej štruktúrálnej rozmanitosti bol Yoel Matveyev v roku 2020 schopný dokázať, že *Brian's Brain* je Turingovsky úplný [21].

## 2.5 Režim konečnej teploty

Režim konečnej teploty (*finite temperature*) je fyzikálne motivovaný model navrhnutý Adachim a kol. [22], ktorý interpretuje Hru života ako termodynamický systém s energetickou funkciou a teplotou. Model využíva spojitý stavový priestor  $S_{ij} \in [0, 1]$ , v ktorom aktuálna hodnota bunky zodpovedá miere jej „živosti“.

Prechodová funkcia je definovaná nasledujúcou sústavou rovníc:

$$x_{ij}(t) = S_{ij}(t) + 2n_{ij}(t), \quad (2.1)$$

$$E_{ij}(t) = E_0 - (x_{ij}(t) - x_0)^2, \quad (2.2)$$

$$S_{ij}(t+1) = \frac{1}{1 + e^{-2E_{ij}/T}}, \quad (2.3)$$

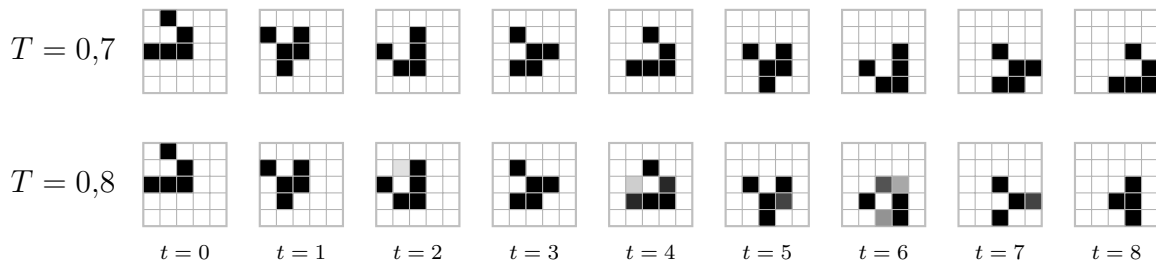
kde  $n_{ij}(t)$  je súčet intenzít ôsmich okolitých susedov,  $E_0$  je energetický posun (určujúci šírku „zóny prežitia“),  $x_0$  je stredová hodnota ideálnych podmienok (posun stavu) a  $T$  reprezentuje teplotu systému.

Rovnica (2.1) mapuje pravidlá Hry života na skalárnu premennú  $x$ , kde stavy vedúce k živej bunke (zrod B3 a prežitie S23) nadobúdajú diskkrétne hodnoty  $x \in \{5, 6, 7\}$ . Energia  $E_{ij}$  podľa vzťahu (2.2) definuje v tomto priestore prevrátenú parabolu so stredom  $x_0 = 6$ . Aby bola energia kladná výhradne pre cieľový interval, čím sa penalizuje zánik bunky v dôsledku izolácie či preľudnenia, autori stanovili optimálny energetický posun na  $E_0 = 2,25$ .

Rovnica (2.3) zabezpečuje, že výstupné stavy zostávajú ohraničené v intervale  $[0, 1]$  a zavádza do systému kľúčový parameter teploty  $T$ , ktorý zásadne mení dynamiku a štatistické vlastnosti systému:

- **Nízka teplota** ( $T \rightarrow 0$ ). Sigmoidálna funkcia konverguje k skokovej funkcii. V tomto limite sa model správa takmer identicky ako klasická Hra života.

- **Mierne teplý režim a teploty rozpadu** ( $T > 0$ ). So zvyšujúcou sa teplotou sa objavujú „tepelné fluktuácie“, ktoré narúšajú diskkrétne stavy buniek. Stabilné a oscilujúce štruktúry dokážu tieto odchýlky určitý čas korigovať, avšak len do dosiahnutia *teploty rozpadu*, ktorá je pre každý vzor individuálna.
- **Termodynamická smrť a chaos** ( $T \geq 1,0$ ). Pri vysokých teplotách fluktuácie prekonajú samoopravnú schopnosť lokálnych pravidiel. Entropia prudko rastie a systém prechádza do chaosu. Všetky lokálne štruktúry zanikajú v šume.



Obr. 2.17: Evolúcia klzáka pri rôznych teplotách v modeli konečnej teploty podľa Adachi a kol. [22]. Pri nižšej teplote ( $T = 0,7$ ) zostáva klzák počas evolúcie rozpoznateľný, zatiaľ čo pri vyššej teplote ( $T = 0,8$ ) sa v dôsledku akumulácie tepelných fluktuácií postupne rozpadá. Svetlosivé a tmavosivé bunky reprezentujú medzistavy excitácie.

## 2.6 Moderné priestorové a spojité zovšeobecnenia

Výskum celulárnych automatov (CA) sa nezastavil len pri modifikácii prechodových pravidiel či zavádzaní viacerých diskrétnych stavov. Zásadným krokom k modelovaniu fyzikálnych a biologických javov bolo uvoľnenie obmedzení týkajúcich sa lokálneho okolia, diskrétného priestoru a synchronného vyhodnocovania.

### 2.6.1 Larger than Life (LtL)

Model *Larger than Life* (LtL), ktorý predstavila Kellie Evans [23], tvorí rodinu dvojrozmerných dvojstavových celulárnych automatov, ktoré zovšeobecňujú niektoré pravidlá na väčšie okolia. Zatiaľ čo klasická Hra života používa okolie s polomerom  $r = 1$ , v modeli LtL je polomer okolia parameter  $r \geq 1$ .

Nech  $x_t(\langle i, j \rangle) \in \{0, 1\}$  označuje stav bunky  $\langle i, j \rangle$  v čase  $t$ . Okolie bunky je dané generalizovaným Moorovým okolím

$$N^r(\langle i, j \rangle) = \{\langle x, y \rangle \in \mathbb{Z}^2 : \max(|x - i|, |y - j|) \leq r\}, \quad (2.4)$$

teda štvorcóm so stranou  $2r + 1$ . V každom kroku sa všetky bunky aktualizujú súčasne podľa počtu živých buniek vo svojom okolí. Ak označíme tento počet ako

$$S_t(\langle i, j \rangle) = \sum_{\langle x, y \rangle \in N^r(\langle i, j \rangle)} x_t(\langle x, y \rangle), \quad (2.5)$$

potom je pravidlo určené intervalom  $[b_1, b_2]$  a intervalom  $[d_1, d_2]$  a má tvar

$$x_{t+1}(\langle i, j \rangle) = \begin{cases} 1 & \text{ak } x_t(\langle i, j \rangle) = 0 \text{ a } S_t(\langle i, j \rangle) \in [b_1, b_2], \\ 1 & \text{ak } x_t(\langle i, j \rangle) = 1 \text{ a } S_t(\langle i, j \rangle) \in [d_1, d_2], \\ 0 & \text{inak.} \end{cases} \quad (2.6)$$

Pre každé pevné  $r$  tak dostávame štvorparametrickú rodinu pravidiel, ktoré Evans označuje päticou  $\langle r, b_1, b_2, d_1, d_2 \rangle$ . V tejto notácii má Conwayova Hra života parametre  $\langle 1, 3, 3, 3, 4 \rangle$ , keďže pri podmienke prežitia sa započítava aj samotná bunka.

Evans vo svojej práci ukazuje, že pravidlá LtL nepodporujú iba stabilné vzory a oscilátory, ale aj pohybujúce sa lokálne konfigurácie nazývané *bugs*. Ide o prirodzené zovšeobecnenie lodí a klzákov na automaty s väčším okolím. Jedným z hlavných záverov práce je, že tieto vzory sú pri vhodnej voľbe parametrov pomerne rozšírené: autorka opisuje lineárne aj kvadratické škálovanie Conwayovho pravidla, nachádza pravidlá podporujúce ortogonálne, diagonálne aj inak orientované vzory.

## 2.6.2 SmoothLife

Ďalším významným zovšeobením Conwayovej Hry života je *SmoothLife*, ktoré navrhol Stephan Rafler [24]. Na rozdiel od klasickej diskkrétnej mriežky sa *SmoothLife* snaží preniesť základné princípy Hry života do spojitého priestoru a k spojitým hodnotám stavovej funkcie. Nech  $f(\mathbf{x}, t) \in [0, 1]$  označuje stav systému v bode  $\mathbf{x}$  a čase  $t$ . Namiesto počítania živých susedov sa v *SmoothLife* používajú dve priemerné hustoty: *vnútorné zaplnenie*  $m$ , ktoré zodpovedá priemeru stavovej funkcie v disku s polomerom  $r_i$ , a *vonkajšie zaplnenie*  $n$ , ktoré predstavuje priemer v prstencovom okolí s vnútorným polomerom  $r_i$  a vonkajším polomerom  $r_a$ . Tieto veličiny sú definované integrálmi

$$m = \frac{1}{M} \int_{|\mathbf{u}| < r_i} f(\mathbf{x} + \mathbf{u}, t) d\mathbf{u}, \quad (2.7)$$

$$n = \frac{1}{N} \int_{r_i < |\mathbf{u}| < r_a} f(\mathbf{x} + \mathbf{u}, t) d\mathbf{u}, \quad (2.8)$$

kde  $M$  a  $N$  sú normalizačné faktory zodpovedajúce obsahu disku a prstenca.

Ďalšou zásadnou zmenou je prechod od diskrétného rozhodovania k hladkej prechodovej funkcii  $s(n, m)$ , ktorá na základe dvojice hodnôt  $m$  a  $n$  určuje nasledujúci stav systému. Namiesto ostrých prahov sa používajú hladké sigmoidálne funkcie, čím sa eliminuje skokové správanie typické pre klasické celulárne automaty.

Hlavným výsledkom je, že aj v tomto spojitom prostredí vznikajú stabilné vzory, oscilujúce štruktúry a predovšetkým pohybujúce sa vzory analogické klzákom a lodiam. Rafler tým ukazuje, že schopnosť vytvárať klzákom podobné štruktúry nie je viazaná na štvorcovú mriežku ani na binárne diskkrétne pravidlá, ale pretrváva aj v hladkom spojitom modeli. Dôležitou vlastnosťou *SmoothLife* je navyše to, že tieto vzory sa môžu pohybovať aj v ľubovoľných smeroch, nielen v smeroch privilegovaných geometriou mriežky.

### 2.6.3 Lenia

Ďalším krokom v zovšeobecňovaní Conwayovej Hry života je *Lenia*, ktorú predstavil Bert Wang-Chak Chan [25]. Ide o dvojrozmerný celulárny automat so spojitým časopriestorovým stavom a zovšeobecným lokálnym pravidlom. *Lenia* nadväzuje na modely *Larger than Life* a *SmoothLife*, no na rozdiel od nich kladie dôraz nielen na samotnú definíciu pravidla, ale aj na systematické štúdium vznikajúcich „životných foriem“.

V diskkrétnej verzii Lenie je stav bunky normalizovaný do intervalu  $[0, 1]$  a okolie je dané diskkrétou guľou s polomerom  $R$ . Namiesto jednoduchého počítania susedov sa nad konfiguráciou vykoná konvolúcia s radiálne symetrickým jadrom  $K$ , čím vznikne potenciál

$$U_t(x) = K * A_t(x). \quad (2.9)$$

Na tento potenciál sa aplikuje rastová funkcia  $G$ :

$$G_t(x) = G(U_t(x)). \quad (2.10)$$

Nový stav sa potom získa inkrementálnou aktualizáciou

$$A_{t+\Delta t}(x) = [A_t(x) + \Delta t G_t(x)]_0^1, \quad (2.11)$$

kde  $[\cdot]_0^1$  označuje orezanie hodnôt do intervalu  $[0, 1]$ .

Dôležitou súčasťou modelu je aj jeho spojitá verzia, ktorá vzniká ako limitný prípad pri zjemňovaní priestorového, časového a stavového rozlíšenia. Chan preto rozlišuje diskrétnu a spojitú *Leniu* a interpretuje prvú z nich ako numerickú aproximáciu druhej. Tým sa *Lenia* zaraďuje medzi tzv. *geometrické celulárne automaty*, ktoré stoja medzi klasickými diskrétnymi celulárnymi automatmi a euklidovskými automatmi.

Hlavným výsledkom práce je, že *Lenia* podporuje mimoriadne bohatú množinu autonómnych lokalizovaných štruktúr. Autor identifikoval viac než 400 druhov rozdelených do 18 rodín. Tieto štruktúry sa podobajú mikroskopickým organizmom nielen vzhľadom, ale aj správaním: môžu byť stabilné, oscilujúce, translujúce, rotujúce alebo sa pohybovať po zložitejších trajektóriách.

Chan zároveň navrhuje biologicky inšpirovaný opis týchto štruktúr. Skúma ich morfológiu, správanie, rozloženie v priestore parametrov a zavádza hierarchickú taxonómiu. Ukazuje, že jednotlivé druhy obsadzujú súvislé oblasti parametrového priestoru „niches“ a že medzi ich symetriou a spôsobom pohybu existujú pravidelné vzťahy. Jednou z navrhnutých interpretácií je tzv. *stability-motility hypothesis*: symetria podporuje stabilitu, zatiaľ čo asymetria umožňuje pohyb.

## 2.7 Fuzzy a intuicionistické celulárne automaty

Ďalším dôležitým prístupom k modelovaniu zložitých systémov sú takzvané fuzzy celulárne automaty (FCA), ktoré vo svojej práci formalizovali Mráz a kol. [26]. Ich motiváciou je syntéza modelov priamo zo správania reálnych systémov, nie iba analýza už vopred zadaných celulárnych automatov. Pri takomto prístupe však často chýbajú presné znalosti o dynamike systému. Autori preto zavádzajú do celulárnych automatov fuzzy logiku, ktorá umožňuje pracovať s neurčitými vstupmi aj s neurčitými pravidlami.

Na rozdiel od klasického konečného automatu tu nevystupujú ostré diskkrétne stavy a prechody, ale vektory fuzzy množín reprezentujúce vstupy a vnútorné stavy spolu so stacionárnou množinou fuzzy pravidiel, ktoré určujú zmenu stavu aj výstupu. Pri prechode od klasického celulárneho automatu k fuzzy celulárnemu automatu sa teda fuzzifikujú predovšetkým stavy buniek a lokálne prechodové pravidlá; podľa aplikácie možno rozšíriť aj definíciu susedstva.

Autori demonštrujú praktické využitie FCA na simulácii šírenia lesného požiaru v homogénnom prírodnom prostredí. Model bol zostavený na základe znalostnej bázy 288 fuzzy pravidiel vytvorených spolu s požiarnymi expertmi. Pravidlá zohľadňovali smer a rýchlosť vetra, nadmorskú výšku, horľavosť bunky, množstvo paliva a poveternostné podmienky. V jednoduchšom homogénnom prípade FCA reprodukoval tvar šírenia požiaru porovnateľne s klasickými semi-eliptickými modelmi, pričom pri zložitejšom reálnom prípade dosiahol približne 75 % zhodu s pozorovaným priebehom požiaru.

Ďalší typ zovšeobecnenia predstavujú *intuicionisticky fuzzy* interpretácie Conwayovej Hry života, ktoré navrhli Liliya Atanassova a Krassimir Atanassov [27]. Každéj bunke

nepriradujú iba binárny stav živá/mŕtva, ale dvojicu hodnôt  $\langle \mu_{i,j}, \nu_{i,j} \rangle$ , kde  $\mu_{i,j}$  vyjadruje stupeň prítomnosti života,  $\nu_{i,j}$  stupeň jeho neprítomnosti a platí  $\mu_{i,j} + \nu_{i,j} \leq 1$ . Zvyšok  $\pi_{i,j} = 1 - \mu_{i,j} - \nu_{i,j}$  predstavuje mieru neurčitosti. Klasická Hra života je v tomto rámci len špeciálny prípad, keď živá bunka zodpovedá dvojici  $\langle 1, 0 \rangle$  a mŕtva bunka dvojici  $\langle 0, 1 \rangle$ .

Autori pritom nenavrhujú jedno jediné fuzzy pravidlo, ale celý súbor možných interpretácií. Zavádzajú šesť kritérií, podľa ktorých možno rozhodnúť, či bunka obsahuje „život“, štyri kritériá zrodu a štyri kritériá zániku. Okrem priameho rozšírenia klasického pravidla uvažujú aj pesimistické, optimistické a priemerové varianty. Samotná zmena stavu bunky sa potom realizuje pomocou intuicionisticky fuzzy negácie; článok uvádza až 34 možných tvarov tejto operácie a následne z nich skladá viacero aktualizáčnych pravidiel, vrátane verzií, ktoré zohľadňujú aj vplyv susedstva.

Hlavným prínosom práce je teda formálny rámec pre reinterpretáciu Hry života v prostredí fuzzy logiky. Článok sa sústreďuje najmä na definície a systematický návrh pravidiel, nie na podrobnú experimentálnu analýzu vznikajúcich vzorov.

## 2.8 Aplikácie a presahy do iných disciplín

Význam celulárnych automatov ďaleko presahuje rámec teoretickej informatiky či diskkrétnej matematiky. Zohráva dôležitú úlohu pri modelovaní dynamických systémov v rôznych vedeckých oblastiach:

- **Počítačová grafika.** Celulárne automaty sa používajú na procedurálne generovanie textúr, terénov alebo animácií pripomínajúcich prírodné procesy (napr. oheň, voda).
- **Simulácia fyzikálnych javov.** Modely ako Lattice Gas Automata, ktoré sú špeciálnym typom celulárnych automatov, sa využívajú na simuláciu prúdenia tekutín.
- **Kryptografia.** Niektoré celulárne automaty vykazujú natolko chaotické a nepredvídateľné správanie, že slúžia ako základ pseudonáhodných generátorov a šifrovacích schém [18].
- **Biológia a ekológia.** Používajú sa na modelovanie šírenia chorôb, rastu populácií alebo interakcií medzi druhmi.

Tieto príklady ukazujú, že princípy Hry života a jej zovšeobecnení majú široký dosah a slúžia ako základný model pre štúdium komplexných systémov.

# Kapitola 3

## Celulárny automat *Half-Life*

Klasická Conwayova Hra života pracuje výlučne s binárnymi stavmi: bunka je buď živá, alebo mŕtva. Tento diskretný prístup síce vedie k emergentným javom, avšak zároveň obmedzuje modelovaciu silu systému. V reálnych biologických, fyzikálnych a sociálnych systémoch sa stavy entít spravidla menia plynulo, nie skokovo. Bunka môže byť „čiasťočne živá“, populácia môže rásť postupne a prechod medzi stavmi nemusí byť ostrý.

Práve toto pozorovanie motivovalo vznik nového modelu, ktorý je hlavným prínosom tejto práce – trojstavového celulárneho automatu s názvom *Half-Life*. Tento model predstavuje most medzi diskretným svetom Conwayovej Hry života a spojitými modelmi, ako je *SmoothLife*. Názov *Half-Life* odkazuje predovšetkým na to, že zmena stavu prebieha po poloviciach. Bunka nezanikne okamžite, ale prechádza cez medzistupeň „polovičného života“.

Pri návrhu modelu sme zvolili ternárny systém, teda systém s tromi stavmi, ako najjednoduchšie možné rozšírenie klasického binárneho modelu. Táto voľba predstavuje ideálny kompromis: zavedenie tretieho, prechodného stavu umožňuje modelovať plynulejšiu dynamiku, no zároveň je systém stále dostatočne jednoduchý na to, aby sme mohli formálne analyzovať jeho vlastnosti. Zložitejšie modely s viacerými stavmi by síce mohli ponúknuť bohatšiu dynamiku, avšak za cenu výrazne vyššej výpočtovej a analytickej zložitosti.

### 3.1 Formálna definícia

Jednotlivé komponenty modelu *Half-Life* budeme definovať v rovnakom poradí ako pri klasickej Hre života.

**Definícia 3.1** (Mriežka automatu *Half-Life*). *Mriežkou automatu *Half-Life** budeme nazývať množinu  $\mathbb{Z}^2$ .

**Definícia 3.2** (Bunka automatu Half-Life). *Bunkou automatu Half-Life* budeme nazývať prvok mriežky automatu Half-Life.

**Definícia 3.3** (Stav bunky v automate Half-Life). Množinu stavov automatu Half-Life definujeme ako

$$\text{Stavy}_{\text{HL}} = \{0, 0,5, 1\}.$$

*Stavom bunky v automate Half-Life* nazývame prvok množiny  $\text{Stavy}_{\text{HL}}$ , pričom stav 0 označuje mŕtvu bunku, stav 0,5 položivú bunku a stav 1 plne živú bunku.

*Poznámka.* Každéj bunke mriežky automatu Half-Life prislúcha práve jeden stav. Vo vizualizáciách budeme bunku so stavom 1 znázorňovať čiernou farbou, bunku so stavom 0,5 sivou farbou a bunku so stavom 0 bielou farbou.

**Definícia 3.4** (Konfigurácia automatu Half-Life). *Konfiguráciou automatu Half-Life* budeme nazývať funkciu  $h: \mathbb{Z}^2 \rightarrow \text{Stavy}_{\text{HL}}$ , ktorá každej bunke priraduje jej stav. Množinu všetkých konfigurácií automatu Half-Life budeme označovať  $\text{Konfigurácie}_{\text{HL}}$ .

**Definícia 3.5** (Vzor automatu Half-Life). *Vzorom automatu Half-Life* budeme nazývať konfiguráciu  $h \in \text{Konfigurácie}_{\text{HL}}$ , ak množina

$$\{x \in \mathbb{Z}^2 : h(x) \neq 0\}$$

je konečná.

**Definícia 3.6** (Posuny v automate Half-Life). *Posunmi v automate Half-Life* budeme nazývať prvky množiny  $\{-1, 0, 1\}^2 \setminus \{(0, 0)\}$ . Keďže susedstvo ostáva rovnaké ako v klasickej Hre života, množinu všetkých posunov budeme aj tu označovať Posuny.

**Definícia 3.7** (Súčet susedov v automate Half-Life). Množinu možných hodnôt súčtu susedov v automate Half-Life budeme označovať nasledovne

$$\text{Súčty}_{\text{HL}} = \left\{ \frac{k}{2} : k \in \{0, 1, \dots, 16\} \right\}.$$

Nech  $h$  je konfigurácia automatu Half-Life a  $x$  je bunka automatu Half-Life. Definujme funkciu  $\text{SúčetSusedov}_{\text{HL}}: \text{Konfigurácie}_{\text{HL}} \times \mathbb{Z}^2 \rightarrow \text{Súčty}_{\text{HL}}$  nasledovne

$$\text{SúčetSusedov}_{\text{HL}}(h, x) = \sum_{n \in \text{Posuny}} h(x + n).$$

Hodnotu  $\text{SúčetSusedov}_{\text{HL}}(h, x)$  budeme nazývať *súčtom susedov bunky  $x$  v konfigurácii  $h$  automatu Half-Life*.

**Definícia 3.8** (Pravidlo automatu Half-Life). *Pravidlom automatu Half-Life*, alebo skrátene *pravidlom*, budeme nazývať dvojicu  $\langle B, S \rangle$ , kde  $B$  je interval hodnôt pre zrod a  $S$  je množina hodnôt pre prežitie.

*Poznámka.* V tejto práci budeme pravidlá automatu Half-Life zapisovať v tvare  $Ba-b/Sc-d, e-f$ , kde  $a-b$ ,  $c-d$  a  $e-f$  označujú intervaly hodnôt. Interval môže byť aj jednoprvkový, potom takýto interval zapisujeme skrátene bez pomlčky. Viacero intervalov zapisujeme oddelené čiarkou. V pravidlových reťazcoch používame pri necelých hodnotách desatinnú bodku, napríklad  $B1/S1.5-2$ ; desatinnú čiarku ponechávame iba v matematickom zápise a v bežnom slovenskom texte. Čiarka v pravidlovom reťazci slúži výlučne na oddelenie intervalov, napríklad v zápise  $S2-3,5-8$ . V matematickom zápise intervalov používame ako oddelovač hraníc stredník, napríklad  $S = [1,5; 2]$ .

**Definícia 3.9** (Cieľová funkcia automatu Half-Life). Nech  $\langle B, S \rangle$  je pravidlo. Funkciu CieľováFunkcia:  $\text{Stavy}_{\text{HL}} \times \text{Súčty}_{\text{HL}} \rightarrow \{0, 1\}$  definovanú predpisom

$$\text{CieľováFunkcia}(q, s) = \begin{cases} 1 & \text{ak } q = 0 \text{ a } s \in B \quad (\text{zrod}), \\ 1 & \text{ak } q \in \{0,5, 1\} \text{ a } s \in S \quad (\text{prežitie}), \\ 0 & \text{inak.} \end{cases}$$

budeme nazývať *cieľovou funkciou automatu Half-Life*. Táto funkcia určuje, či bunka automatu Half-Life smeruje k plne živému stavu 1, alebo k mŕtvemu stavu 0.

**Definícia 3.10** (Lokálna prechodová funkcia automatu Half-Life). Nech  $\langle B, S \rangle$  je pravidlo. Funkciu  $f_{B,S}$ :  $\text{Stavy}_{\text{HL}} \times \text{Súčty}_{\text{HL}} \rightarrow \text{Stavy}_{\text{HL}}$  definovanú predpisom

$$f_{B,S}(q, s) = q + 0,5 \cdot \text{sgn}(\text{CieľováFunkcia}(q, s) - q),$$

budeme nazývať *lokálnou prechodovou funkciou automatu Half-Life*. V automate Half-Life tak zrod prebieha ako  $0 \rightarrow 0,5 \rightarrow 1$  a zánik ako  $1 \rightarrow 0,5 \rightarrow 0$ .

**Definícia 3.11** (Globálna prechodová funkcia automatu Half-Life). Nech  $\langle B, S \rangle$  je pravidlo automatu Half-Life. Funkciu  $\text{GPF}_{B,S}$ :  $\text{Konfigurácie}_{\text{HL}} \rightarrow \text{Konfigurácie}_{\text{HL}}$  definovanú predpisom

$$(\text{GPF}_{B,S}(h))(x) = f_{B,S}(h(x), \text{SúčetSusedov}_{\text{HL}}(h, x)), \quad \text{pre každé } x \in \mathbb{Z}^2,$$

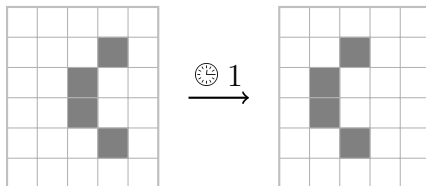
budeme nazývať *globálnou prechodovou funkciou automatu Half-Life*.

**Definícia 3.12** (Evolúcia automatu Half-Life). *Evolúciou automatu Half-Life* pre pravidlo automatu Half-Life  $\langle B, S \rangle$  budeme nazývať postupnosť  $(h_t)_{t=0}^{\infty}$  prvkov množiny  $\text{Konfigurácie}_{\text{HL}}$ , pre ktorú v každom kroku  $t \geq 0$  platí:

$$h_{t+1} = \text{GPF}_{B,S}(h_t).$$

## 3.2 Pohyb klzáka

Na ilustráciu dynamiky a na formálny dôkaz vlastností zvolíme pravidlo určené intervalmi  $B = [1; 1]$  a  $S = [1, 5; 2]$ . V tomto pravidle existuje jednoduchý vzor s periódou 1 a rýchlosťou  $c$ , teda vzor, ktorý sa po mriežke posúva o jednu bunku za generáciu. V ďalšom texte tejto kapitoly budeme pracovať výlučne s pravidlom B1/S1.5-2 a namiesto označenia  $\text{GPF}_{[1;1],[1,5;2]}$  budeme používať skrátenejší zápis GPF.



Obr. 3.18: Evolúcia vzoru  $H_0$  s rýchlosťou  $c$  v pravidle B1/S1.5-2. Vzor sa v každom kroku posunie o jednu bunku doľava.

**Definícia 3.13** (Translácia konfigurácie). Nech  $v = \langle a, b \rangle \in \mathbb{Z}^2$ . Transláciou konfigurácie o vektor  $v$  nazývame zobrazenie  $T_v: \text{Konfigurácie}_{\text{HL}} \rightarrow \text{Konfigurácie}_{\text{HL}}$  definované vzťahom

$$(T_v h)(\langle x, y \rangle) = h(\langle x - a, y - b \rangle)$$

pre každú konfiguráciu  $h \in \text{Konfigurácie}_{\text{HL}}$  a každú bunku  $\langle x, y \rangle \in \mathbb{Z}^2$ .

Pred dôkazom pohybu klzáka si všimnime dôležitú vlastnosť pravidla GPF, a to jeho invariantnosť vzhľadom na translácie.

**Lema 3.1** (Translačná invariancia). Pre každú konfiguráciu  $h \in \text{Konfigurácie}_{\text{HL}}$  a každý vektor  $v = \langle a, b \rangle \in \mathbb{Z}^2$  platí

$$\text{GPF}(T_v h) = T_v \text{GPF}(h).$$

*Dôkaz.* Pre ľubovoľnú bunku  $\langle x, y \rangle$  máme

$$\begin{aligned} \text{SúčetSusedov}_{\text{HL}}(T_v h, \langle x, y \rangle) &= \sum_{n \in \text{Posuny}} (T_v h)(\langle x + n_x, y + n_y \rangle) \\ &= \sum_{n \in \text{Posuny}} h(\langle x + n_x - a, y + n_y - b \rangle) \\ &= \text{SúčetSusedov}_{\text{HL}}(h, \langle x - a, y - b \rangle). \end{aligned}$$

Odtiaľ

$$\begin{aligned}
(\text{GPF}(T_v h))(\langle x, y \rangle) &= f_{B,S}((T_v h)(\langle x, y \rangle), \text{SúčetSusedov}_{\text{HL}}(T_v h, \langle x, y \rangle)) \\
&= f_{B,S}(h(\langle x - a, y - b \rangle), \text{SúčetSusedov}_{\text{HL}}(h, \langle x - a, y - b \rangle)) \\
&= (\text{GPF}(h))(\langle x - a, y - b \rangle) \\
&= (T_v \text{GPF}(h))(\langle x, y \rangle).
\end{aligned}$$

□

Nasledujúce tvrdenie formalizuje skutočnosť, že zvolený vzor je loďou s periódou 1 a rýchlosťou  $c$ .

**Tvrdenie 3.1** (Pohyb klzáka). Nech  $H_0$  je konfigurácia definovaná predpisom

$$H_0(\langle x, y \rangle) = \begin{cases} 0,5, & \text{ak } \langle x, y \rangle \in \{\langle 3, 4 \rangle, \langle 2, 3 \rangle, \langle 2, 2 \rangle, \langle 3, 1 \rangle\}, \\ 0, & \text{inak.} \end{cases}$$

Potom pre každé  $t \geq 0$  platí

$$\text{GPF}^t(H_0) = T_{\langle -t, 0 \rangle} H_0.$$

*Dôkaz.* Dôkaz vykonáme indukciou podľa  $t$ .

1. Pre  $t = 0$  je tvrdenie triviálne, pretože

$$\text{GPF}^0(H_0) = H_0 = T_{\langle 0, 0 \rangle} H_0.$$

2. Nech teraz  $k \geq 0$  a predpokladajme, že

$$\text{GPF}^k(H_0) = T_{\langle -k, 0 \rangle} H_0.$$

Ukážeme, že potom platí aj

$$\text{GPF}^{k+1}(H_0) = T_{\langle -(k+1), 0 \rangle} H_0.$$

Na tento účel najprv ukážeme, že

$$\text{GPF}(H_0) = T_{\langle -1, 0 \rangle} H_0.$$

Označme  $P_0$  množinu položivých buniek konfigurácie  $H_0$  a  $Q_0$  množinu položivých buniek konfigurácie  $T_{\langle -1, 0 \rangle} H_0$ , teda

$$P_0 = \{\langle 3, 4 \rangle, \langle 2, 3 \rangle, \langle 2, 2 \rangle, \langle 3, 1 \rangle\}, \quad Q_0 = \{\langle 2, 4 \rangle, \langle 1, 3 \rangle, \langle 1, 2 \rangle, \langle 2, 1 \rangle\}.$$

Ďalej označme

$$N_M(P_0) = \{\langle x, y \rangle \in \mathbb{Z}^2 : \exists a, b \in \mathbb{Z}, \langle a, b \rangle \in P_0, \max(|x - a|, |y - b|) \leq 1\}.$$

Rozlíšime tri prípady.

(a) Ak  $x \notin N_M(P_0)$ , potom platí  $H_0(x) = 0$  a  $\text{SúčetSusedov}_{\text{HL}}(H_0, x) = 0$ , a teda

$$(\text{GPF}(H_0))(x) = 0.$$

(b) Ak  $x \in P_0$ , potom

$$\text{SúčetSusedov}_{\text{HL}}(H_0, \langle 3, 4 \rangle) = \text{SúčetSusedov}_{\text{HL}}(H_0, \langle 3, 1 \rangle) = 0,5,$$

$$\text{SúčetSusedov}_{\text{HL}}(H_0, \langle 2, 3 \rangle) = \text{SúčetSusedov}_{\text{HL}}(H_0, \langle 2, 2 \rangle) = 1.$$

Keďže ani jedna z týchto hodnôt neleží v intervale prežitia  $S = [1,5; 2]$ , pre každé  $x \in P_0$  dostávame

$$(\text{GPF}(H_0))(x) = 0.$$

(c) Nech  $x \in N_M(P_0) \setminus P_0$ . V tomto prípade rozlíšime dve možnosti.

- Ak  $x \in Q_0$ , potom  $H_0(x) = 0$  a  $\text{SúčetSusedov}_{\text{HL}}(H_0, x) = 1$ , takže

$$(\text{GPF}(H_0))(x) = f_{B,S}(0, 1) = 0,5.$$

- Ak  $x \in N_M(P_0) \setminus (P_0 \cup Q_0)$ , potom  $\text{SúčetSusedov}_{\text{HL}}(H_0, x) \in \{0,5, 1,5\}$ , takže bunka nezaniká ani sa nerodí, a preto

$$(\text{GPF}(H_0))(x) = 0.$$

Teda jedinými nenulovými bunkami konfigurácie  $\text{GPF}(H_0)$  sú práve bunky z  $Q_0$ , všetky v stave 0,5, a preto

$$\text{GPF}(H_0) = T_{\langle -1, 0 \rangle} H_0.$$

Teraz už s využitím Lemy 3.1 dostávame

$$\begin{aligned} \text{GPF}^{k+1}(H_0) &= \text{GPF}(\text{GPF}^k(H_0)) \\ &= \text{GPF}(T_{\langle -k, 0 \rangle} H_0) \\ &= T_{\langle -k, 0 \rangle} \text{GPF}(H_0) \\ &= T_{\langle -k, 0 \rangle} T_{\langle -1, 0 \rangle} H_0 \\ &= T_{\langle -(k+1), 0 \rangle} H_0. \end{aligned}$$

Tým je indukčný krok dokázaný.

Preto pre každé  $t \geq 0$  platí

$$\text{GPF}^t(H_0) = T_{\langle -t, 0 \rangle} H_0.$$

□

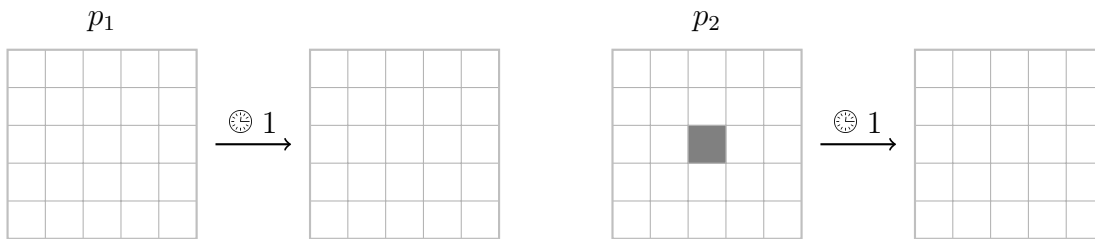
### 3.3 Rajské záhrady

Podobne ako v klasickej Hre života, aj v pravidlách automatu *Half-Life* možno dokázať existenciu vzorov bez predchodcu.

**Tvrdenie 3.2** (*Half-Life* rajské záhrady). V automate *Half-Life* s pravidlom B1/S1.5-2 existujú rajské záhrady.

*Dôkaz.* Keďže *Half-Life* je celulárny automat s konečným stavovým priestorom  $\text{Stavy}_{\text{HL}}$ , môžeme aj tu použiť Mooreovu–Myhillovu vetu.

Uvažujme oblasť  $E$  s rozmerom  $5 \times 5$ . Nech  $p_1$  je lokálna konfigurácia, v ktorej sú všetky bunky v oblasti  $E$  v stave 0. Nech  $p_2$  je lokálna konfigurácia, v ktorej je stredová bunka oblasti  $E$  v stave 0,5 a zvyšných 24 buniek je v stave 0.



Obr. 3.19: Lokálne vzájomne vymazateľné vzory vzhľadom na oblasť  $E$  v pravidle B1/S1.5-2.

Analyzujme evolúciu oboch konfigurácií do ďalšej generácie pri ľubovoľnom, ale v oboch prípadoch zhodnom okolí mimo oblasti  $E$ :

- **Pre konfiguráciu  $p_1$ .** Vo vnútornom podštvorci  $3 \times 3$  majú všetky bunky stav 0 a susedský súčet  $s = 0$ . Keďže  $0 \notin B$ , cieľová funkcia je  $\text{CieľováFunkcia}(0, 0) = 0$ , a teda ich nový stav zostáva 0.
- **Pre konfiguráciu  $p_2$ .** Stredová bunka má stav  $q = 0,5$  a susedský súčet  $s = 0$ , pretože vo svojom Moorovom okolí neobsahuje žiadnu inú nenulovú bunku. Keďže  $0 \notin S$ , jej cieľová funkcia je  $\text{CieľováFunkcia}(0, 5, 0) = 0$ . Jej stav v čase  $t + 1$  je

$$q_{t+1} = 0,5 + 0,5 \cdot \text{sgn}(0 - 0,5) = 0.$$

Osem zvyšných buniek vnútorného podštvorca  $3 \times 3$  má stav 0 a vo svojom okolí jediná položivú bunku, čiže súčet susedov je  $s = 0,5$ . Keďže  $0,5 \notin B$ , ich cieľová funkcia je  $\text{CieľováFunkcia}(0, 0,5) = 0$ , a preto zostávajú v stave 0.

- **Vonkajší prstenec oblasti  $E$ .** Bunky na okraji oblasti  $E$  sú od stredovej bunky vzdialené viac než o jednu bunku, a preto ju vo svojom Moorovom okolí vôbec nevidia. Ich okolia sú teda v konfiguráciách  $p_1$  a  $p_2$  identické, takže sa v oboch prípadoch vyvinú rovnako. To isté platí aj pre všetky bunky mimo oblasti  $E$ .

Vidíme, že hoci sú počiatočné konfigurácie  $p_1$  a  $p_2$  odlišné, po jednom kroku globálnej prechodovej funkcie vedú pri zhodnom vonkajšom okolí k tej istej výslednej konfigurácii. Lokálne vzory  $p_1$  a  $p_2$  sú teda vzájomne vymazateľné vzhľadom na oblasť  $E$ , a preto podľa Mooreovej–Myhillovej vety v automate *Half-Life* s týmto pravidlom existujú konfigurácie bez predchodcu. □

### 3.4 Parametrizácia a priestor pravidiel

V predchádzajúcich sekciách sme pravidlá vyberali najmä intuitívne a skôr experimentálne. Je však prirodzené očakávať, že z hľadiska vzniku zaujímavých vzorov nie sú všetky pravidlá rovnako vhodné: mnohé vedú len k triviálnemu zániku alebo nekontrolovanému rastu, zatiaľ čo iba niektoré podporujú bohatšiu dynamiku. To motivuje potrebu systematického prieskumu priestoru pravidiel, ktorému sa venuje nasledujúca kapitola.

# Kapitola 4

## Analytický nástroj Half-Life Explorer

Manuálne hľadanie pravidiel v takomto rozsiahlom stavovom priestore je neefektívne. Pre predstavu, v režime *Half-Life* môže susedský súčet nadobudnúť 17 rôznych hodnôt (od 0 do 8 s krokom 0,5). To znamená, že pri uvažovaní ľubovoľných podmnožín pre zrod a prežitie existuje až  $2^{17} \times 2^{17} = 2^{34} \approx 1,7 \cdot 10^{10}$  možných pravidiel. Dokonca aj pri obmedzení na jednoduché súvislé intervaly ostáva vyše 23 000 kombinácií. Analytický nástroj sme preto navrhli tak, aby automatizovane identifikoval pravidlá, ktoré podľa zvolených metrik vykazujú požadované vlastnosti, napríklad vznik stabilných alebo periodických štruktúr, a zároveň nevedú prevažne k okamžitému zániku ani k rýchlemu nekontrolovanému rastu.

Nástroj *Half-Life Explorer* realizuje systematické prehľadávanie priestoru pravidiel prostredníctvom masívnej simulácie, čím umožňuje automatickú detekciu a klasifikáciu vznikajúcich vzorov. Pomocou neho sme chceli identifikovať pravidlá s požadovanými vlastnosťami a porovnať ich pomocou kvantitatívnych metrik.

### 4.1 Architektúra a jadro systému

Nástroj sme navrhli s dôrazom na praktickú spracovateľnosť rozsiahleho priestoru pravidiel, pamäťovo úspornú reprezentáciu dát a paralelné spracovanie behov. Vyhľadávací priestor pravidiel *Half-Life* narastá exponenciálne, obzvlášť ak uvažujeme rozdelené intervaly pre prežitie. Na spoľahlivé empirické posúdenie jedného pravidla je preto nutné vyhodnotiť evolúciu tisícok rôznych počiatočných náhodných konfigurácií.

Z architektonického hľadiska je jadro systému modulárne rozdelené takto:

- **Dátový a logický model (`grid.rs`, `rule.rs`).** Zabezpečujú priestorovú reprezentáciu dvojrozmernej mriežky a vyhodnocovanie prechodových funkcií. V záujme jednoduchej a deterministickej internej reprezentácie sú zlomkové stavy buniek

mapované na celočíselné hodnoty, čím sa zároveň predchádza chybám pri výpočtoch s číslami s pohyblivou rádovou čiarkou (floating-point).

- **Moduly vyhľadávania (`explorer_2d.rs`, `explorer_1d.rs`).** Poskytujú infraštruktúru pre prieskum rozsiahleho gridu parametrov pravidiel s využitím generovania tzv. *random soups*. V hlavných 2D experimentoch sa pre každé pravidlo generuje 3 000 počiatočných konfigurácií na mriežke  $48 \times 48$ , pričom do stredu sa vloží štvorcový počiatočný blok s náhodnou veľkosťou rovnomerne zvolenej z množiny  $\{3, 4, 5, 6, 7, 8\}$ . Hustota  $d$  sa náhodne vyberá raz pre celý počiatočný blok z intervalu  $[0,15, 0,45]$  a následne sa každá jeho bunka generuje nezávisle: s pravdepodobnosťou  $d/2$  nadobudne stav 1, s pravdepodobnosťou  $d/2$  stav 0,5 a inak stav 0. Počiatočné stavy teda neobsahujú iba binárne hodnoty 0/1, ale aj medzistav 0,5. Generovanie využíva knižničný generátor `rand::thread_rng()` lokálny pre každé pracovné vlákno a paralelizácia medzi pravidlami je realizovaná pomocou knižnice Rayon.
- **Rozpoznávanie vzorov (`analysis.rs`, `glider.rs`).** Predstavuje analytickú vrstvu nástroja, ktorá je zodpovedná za určenie kategórie nájdeného emergentného správania počas simulovaného úseku. Vnútorne pravidlo automatu používa periodické okrajové podmienky (toroidálne obalenie mriežky), no heuristika klasifikácie beh ukončuje už pri prvom zásahu okraja, aby sa zabránilo interakcii vzoru s jeho vlastnou zabalenou kópiou cez torus.

Aktuálna implementácia explicitne neukladá seed generátora pseudonáhodných čísel a používa viacvláknové `thread_rng()`, preto nie je bitovo reprodukovateľná. Reprodukovať možno všetky konfiguračné parametre experimentu (rozмеры mriežky, interval hustoty, počet behov a limity heuristík) a samotné objavené vzory nástroj priebežne ukladá do výstupných JSONL súborov. Úplná reprodukovateľnosť by si vyžadovala deterministické seedovanie jednotlivých workerov a archiváciu použitých seedov.

## 4.2 Algoritmická klasifikácia správania a vzorov

Metodika klasifikácie je založená na simulácii evolúcie veľkého množstva náhodných vzorov. Každý beh prejde najprv 30-krokovou „warm-up“ fázou a potom ešte najviac 120 krokmi, počas ktorých prebieha vlastná klasifikácia; jeden seed je teda pozorovaný najviac 150 generácií. V každom kroku sa vyhodnocujú nasledujúce kategórie:

- **Zánik (Dead).** Populácia na mriežke po určitom počte krokov klesne na nulu.

- **Explózia (Explode).** Detektor klasifikuje populáciu ako explóziu, ak počet nenulových buniek v ľubovoľnom kroku prekročí prah 400, alebo ak vzor zasiahne okraj mriežky a následná heuristika nepotvrdí, že ide o loď.
- **Oscilátor.** Algoritmus si uchováva históriu posledných 60 orezaných stavov a v každom kroku porovnáva aktuálny tvar s touto históriou. Ak sa niektorý zo skorších súvislých komponentov zopakuje bez zistenej translácie, ide o oscilátor a z časového odstupu medzi zhodnými stavmi možno určiť jeho periódu. Z toho vyplýva, že bežný režim spoľahlivo zachytáva len oscilátory s periódou najviac 60.
- **Posuvný vzor („Spaceship“).** Detektor sleduje opakovanie rovnakého komponentu na inom mieste mriežky. Pomocným signálom je aj tzv. „Edge Hit“, teda zásah hranice simulovanej mriežky nenulovou bunkou, ktorý môže naznačovať pohyb vzoru mimo počiatočnej oblasti. Kandidáti na loď sa následne overujú v samostatnom testovacom priestore  $64 \times 64$  počas najviac 500 krokov. Heuristika ignoruje komponenty väčšie ako 100 buniek a akceptuje iba lode s periódou najviac 60.
- **Chaos.** Ide o reziduálnu pracovnú kategóriu použitého detektora. Spadajú do nej všetky behy, ktoré po najviac 150 generáciách nezaniknú, neprekročia prah explózie a zároveň sa ich v rámci nastavených limitov nepodarí jednoznačne overiť ako oscilátor alebo loď. Označenie „Chaos“ preto v tejto práci neznamená dôkaz matematického chaosu.

Z uvedeného vyplýva, že klasifikácia má charakter heuristického screeningu, nie úplnej klasifikácie dlhodobého správania automatu. Výsledky kategórií „Dead“, „Explode“, „Oscilátor“, „Spaceship“ a „Chaos“ preto treba interpretovať vzhľadom na konkrétne parametre detektora: 150 pozorovaných generácií, históriu 60 stavov pri detekcii periodicity, limit 100 buniek pri overovaní lodí a pomocný verifikačný priestor  $64 \times 64$ .

Pri veľkom počte simulácií vzniká značné množstvo redundantných vzorov. Pre efektívnu analýzu je kľúčová kanonizácia a filtrácia výsledkov. Nástroj do výsledného súboru ukladá iba unikátne vzory, pričom pre každú periódu a tvar ponechá iba jeden reprezentant (eliminujú sa rotácie, zrkadlové obrazy a fázové posuny). Týmto spôsobom sa redukuje objem dát a zvyšuje kvalita výstupu, pričom aplikácia používateľom sprístupňuje len unikátne objavy.

Predvolené nastavenie nástroja používa 12 pracovných vlákien a experimenty sme spúšťali na cloudovej inštancii s 12 vCPU a 32 GB RAM. Keďže sa práca nezameriava na detailné hodnotenie výkonu ani na porovnanie s inými nástrojmi, paralelizácia a pamäťová

architektúra tu vystupujú predovšetkým ako praktické prostriedky umožňujúce realizáciu prieskumu.

### 4.3 Výsledky prieskumu

Jedným z hlavných výstupov programu *Half-Life Explorer* je katalóg pravidiel, pri ktorých sme v simuláciách detegovali vzory sledované zvolenými metrikami.

Ako príklad uvádzame výsledky jedného z najväčších experimentov, ktorý sme zamerali na režim *Half-Life*. Systematickým testovaním všetkých kombinácií jedného *B*-intervalu a jedného *S*-intervalu sme získali  $153 \times 153 = 23\,409$  unikátnych pravidiel. Pre každé z nich nástroj spustil 3 000 simulácií z náhodných počiatočných konfigurácií, čo predstavuje celkovo 70 227 000 behov. V tomto masívnom experimente sme identifikovali 243 pravidiel, ktoré generovali aspoň jeden typ lode.

Tabuľka 4.1 sumarizuje najúspešnejšie pravidlá tohto behu z hľadiska produkcie unikátnych foriem lodí.

Pravidlo	Mŕtve	Explodované	Chaos	Klzáky	Oscilátory
B1/S1.5-2	119	2 340	25	<b>20</b>	3
B1.5-2/S2.5-4.5	1 337	686	167	<b>14</b>	130
B1.5-2/S2.5-4	1 410	645	170	<b>14</b>	100
B1.5/S1	879	0	30	<b>14</b>	29
B2-3/S0.5-1	620	1 728	342	<b>10</b>	33
B1.5-2.5/S2.5-3	868	1 433	204	<b>10</b>	19
B1/S1.5	152	1 879	11	<b>9</b>	2
B1.5/S2-3	1 633	454	402	<b>8</b>	129
B1/S1.5-2.5	115	2 625	95	<b>8</b>	1
B2.5-5.5/S2-2.5	1 981	26	494	<b>7</b>	29

Tabuľka 4.1: Top 10 pravidiel podľa počtu unikátnych lodí. Stĺpce *Mŕtve*, *Explodované* a *Chaos* udávajú počty behov zaradených do príslušných kategórií detektorom, zatiaľ čo stĺpce *Klzáky* a *Oscilátory* udávajú počty unikátnych nájdených vzorov.

Je dôležité poznamenať, že súčet hodnôt v riadku pre dané pravidlo sa nerovná celkovému počtu 3 000 behov. Napríklad pre pravidlo B1/S1.5-2 skončilo  $119 + 2\,340 + 25 = 2\,484$  behov zánikom, explóziou alebo chaosom. Zvyšných 516 behov viedlo k detekcii oscilujúcich alebo pohyblivých štruktúr; po deduplikácii sa do tabuľky započítalo 20 unikátnych typov klzákov a 3 unikátne oscilátory.

Výsledky ukazujú, že použité pravidlá sa líšia nielen v počte nájdených klzákov a oscilátorov, ale aj v tom, ako často simulácie končia zánikom, explóziou alebo chaosom. Napríklad pri pravidle B1/S1.5-2 sme detegovali 20 unikátnych klzákov a 3 unikátne oscilátory, zatiaľ čo pri pravidle B2-4/S0-0.5 sme zaznamenali iba 1 unikátny klzák, no až 533 oscilátorov. Rozdiely sa prejavujú aj v ostatných kategóriách: napríklad pravidlo B1.5/S2-3 pri 8 klzákoch generovalo 129 oscilátorov, pričom 1 633 behov skončilo zánikom a 454 explóziou.

Pre porovnanie sme druhý masívny test orientovali na pravidlá s nesúvislými oblasťami prežitia (tzv. *split-survival*). Inšpiráciou pre tento experiment bol jednorozmerný celulárny automat opísaný v sekcii 2.2, ktorého pravidlo prežitia (2 alebo 4 susedia) taktiež netvorí súvislý interval, ale obsahuje medzeru pre hodnotu 3. Príkladom je pravidlo B3-7/S2-3,5-8, ktoré umožňuje prežitie pri nízkej hustote okolia (hodnoty 2 a 3) a paradoxne aj pri vysokej hustote (hodnoty 5 až 8), avšak pri strednej hodnote 4 bunka umiera. Túto diskontinuitu možno chápať ako možný mechanizmus ovplyvňujúci dynamiku: v pozorovaných prípadoch prispieva k obmedzeniu vzniku veľkých stabilných „kvapiiek“, keďže stredne husté oblasti sa ľahšie rozpadajú. V našich dátach sa tento mechanizmus prejavil cyklickým striedaním rastu a rozkladu, čo viedlo k zvýšenému výskytu oscilátorov. Keďže priestor takýchto pravidiel je veľmi veľký, uvažovali sme iba intervaly s celočíselnými hranicami, čím sa počet možností znížil na 11 475.

Pravidlo	Mŕtve	Explodované	Chaos	Klzáky	Oscilátory
B3-7/S2-3,5-8	1 644	0	8	<b>9</b>	624
B3-8/S2-3,5-7	1 602	0	570	<b>9</b>	141
B2/S2-3,5	1 316	1 385	71	<b>8</b>	7
B2/S2-3,5-6	1 312	1 377	92	<b>8</b>	7
B3-8/S2-3,5-8	1 679	0	8	<b>7</b>	588
B3-7/S2-3,5-7	1 670	0	452	<b>7</b>	225
B2-8/S0-1,5	123	2 134	386	<b>7</b>	58
B2-3/S0-1,7	144	1 594	875	<b>6</b>	56
B2-4/S0-1	123	2 183	326	<b>6</b>	55
B2-4/S0-1,5	116	2 193	326	<b>6</b>	49

Tabuľka 4.2: Najúspešnejšie pravidlá z rozšíreného experimentu so *split-survival* parametrami a výsledky klasifikácie použitým detektorom. Tabuľka môže obsahovať aj pravidlá so súvislým intervalom prežitia, ak sa v tomto behu umiestnili medzi najlepšími.

Výsledky tohto testu sú s touto pracovnou hypotézou konzistentné: hoci počet unikátnych klzákov bol nižší (maximum 9), variabilita oscilátorov bola výrazná. Napríklad spomínané pravidlo B3-7/S2-3,5-8 vygenerovalo 624 unikátnych oscilátorov. Tabuľka 4.2 uvádza pravidlá s najvyšším počtom detegovaných vzorov v tejto časti experimentu.

V druhom experimente sme identifikovali 309 pravidiel produkujúcich klzáky; z nich 266 pravidiel (86 %) využívalo nesúvislé intervaly prežitia a 43 pravidiel (14 %) jeden súvislý interval. Tento výsledok neinterpretujeme ako dôkaz, že pravidlá typu *split-survival* sú všeobecne úspešnejšie než pravidlá so súvislým intervalom prežitia. Naznačuje však, že v skúmanej vzorke a pri použítom nastavení detektora sa nesúvislé intervaly prežitia častejšie objavovali medzi pravidlami, pri ktorých sme identifikovali pohyblivé štruktúry a oscilátory.

Pri interpretácii týchto výsledkov je dôležité odlíšiť parciálne počty v jednotlivých experimentoch od finálneho katalógu. Prvý experiment priniesol 243 pravidiel a druhý 309 pravidiel s aspoň jedným detegovaným klzákom, no tieto množiny sa čiastočne prekrývali, a preto ich nemožno jednoducho sčítať. Po zlúčení a deduplikácii oboch behov vznikol výsledný katalóg obsahujúci 514 pravidiel a 3 381 vzorov.

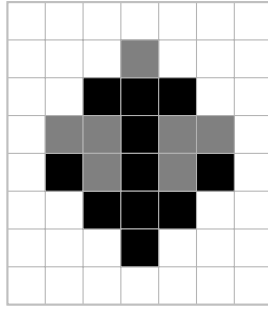
## 4.4 Galéria objavených vzorov

Okrem štatistického vyhodnotenia je dôležité venovať pozornosť aj konkrétnym štruktúram, ktoré jednotlivé pravidlá generujú, vrátane ich morfológických a estetických vlastností. Nasledujúci prehľad sumarizuje kľúčové zistenia a rekordy, ktoré sme identifikovali počas tohto automatizovaného prieskumu, rozdelené podľa jednotlivých kategórií.

### 4.4.1 Rýchlosť a pohyb

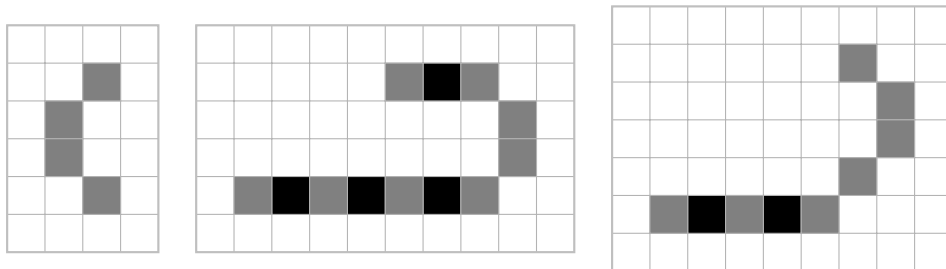
Rýchlosť lodí je jednou z najdôležitejších charakteristík v celulárnych automatoch, keďže priamo súvisí s mechanizmom šírenia informácie v priestore. V klasickej Hre života aj v jej zovšeobecneniach sa rýchlosť typicky vyjadruje ako podiel  $c/k$ , kde  $c$  predstavuje posun o jednu bunku za generáciu.

Jedným z extrémov na tomto spektre je najpomalší objavený klzák s rýchlosťou  $c/26$ , čo znamená, že na posun o jednu bunku potrebuje až 26 generácií. Tento vzor sme identifikovali v pravidlách B3-6/S2-3 a B3-6/S2-3,7-8. Je pozoruhodné, že v oboch prípadoch ide o identický klzák, čo naznačuje, že jeho existencia nie je citlivá na rozšírenie podmienok prežitia o vysoké počty susedov (7 a 8).



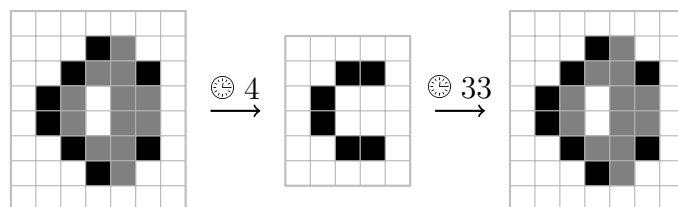
Obr. 4.20: Najpomalší objavený klzák s rýchlosťou  $c/26$ . Tento vzor je loďou v pravidlách B3-6/S2-3 aj B3-6/S2-3,7-8.

Na opačnom konci spektra sa nachádzajú lode s maximálnou možnou rýchlosťou  $c$ , teda s posunom o jednu bunku za generáciu, ktoré sa objavujú v pravidle B1/S1.5-2. Zaujímavosťou tohto pravidla je nielen možnosť konštrukcie ľubovoľne dlhých lodí s nekonečným „chvostom“, ale aj fakt, že vyvracajú predstavu o nutnej symetrii lodí. Zatiaľ čo predchádzajúci vzor bol symetrický, nasledujúca ukážka demonštruje, že táto vlastnosť nie je pravidlom.



Obr. 4.21: Ukážka lodí s rýchlosťou  $c$  v pravidle B1/S1.5-2. Vďaka štruktúre pravidla je možné konštruovať lode s ľubovoľne dlhým chvostom.

Pozoruhodná je aj loď s rýchlosťou  $3c/37$  v pravidle B2.5-8/S1.5-2. Ako vidno na Obrázku 4.22, počas pohybu sa výrazne mení: z masívneho symetrického útvaru prechádza do kompaktnejšej formy.



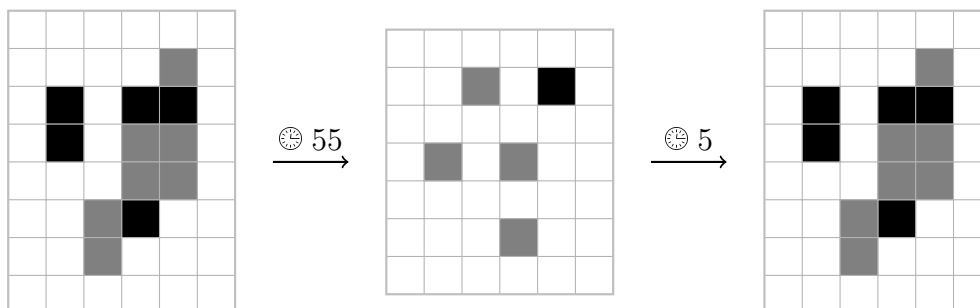
Obr. 4.22: Morfologická premena klzáka  $3c/37$  v pravidle B2.5-8/S1.5-2. Vľavo je zobrazená jeho najväčšia fáza, ktorá po 4 krokoch prechádza do kompaktnej formy (v strede) a po ďalších 33 krokoch sa vracia do pôvodnej fázy.

Okrem týchto extrémov sa podarilo identifikovať bohaté spektrum ďalších rýchlostí. Od maximálnej možnej rýchlosti  $c$  a veľmi rýchlych lodí pohybujúcich sa rýchlosťou  $2c/3$ , cez bežnejšie ortogonálne a stredné rýchlosti ako  $c/3$ ,  $c/2$ ,  $c/4$ ,  $2c/5$ ,  $c/5$  či  $c/6$ . Spektrum pokračuje pomalšími loďami s rýchlosťami  $c/7$ ,  $c/8$  a  $c/9$ , až po veľmi pomalé lode dosahujúce  $c/12$ ,  $c/13$ ,  $c/15$ ,  $c/16$ ,  $c/18$  a  $c/20$ . Zoznam dopĺňajú pomalé lode s rýchlosťami  $c/24$  a  $c/26$ , ako aj exotické rýchlosti  $2c/31$  a  $3c/37$ .

#### 4.4.2 Periodicita

Periodicita vzorov je ďalším kľúčovým parametrom, ktorý odhaľuje zložitosť vnútorných cyklov celulárneho automatu. Zatiaľ čo v klasickej Hre života sú najbežnejšie oscilátory s periódou 2 (blikač, ropucha, maják), vo viacstavových automatoch typu *Half-Life* sme zaznamenali výrazne širšie spektrum správania.

V hlavnom katalógu sme identifikovali **44 unikátnych periód** v rozsahu od 1 do 60, pričom hornú hranicu limitovala nastavená dĺžka histórie detektora. Najvyššiu periódu medzi oscilátormi zachytenými v tomto katalógu dosiahol vzor *Oscillator P60 1*. Tento vzor kontrastuje s inými zachytenými oscilátormi: je pomerne malý vzhľadom na svoju dlhú periódu.



Obr. 4.23: Oscilátor s periódou 60 v pravidle B1.5/S1. Počas svojho cyklu nadobúda aj formu piatich izolovaných buniek.

Medzi loďami drží rekord už spomínaný klzák s rýchlosťou  $3c/37$ , ktorý má periódu **37**. To znamená, že tento vzor potrebuje 37 generácií na to, aby prešiel celým svojim vnútorným cyklom a vrátil sa do pôvodnej fázy (posunutý v priestore).

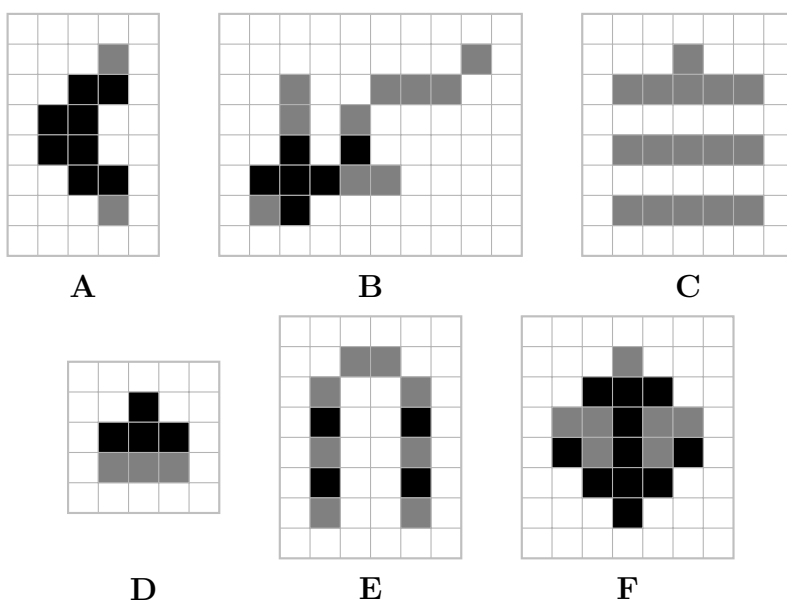
Zaujímavým fenoménom je dominancia periódy 4. Vo väčšine pravidiel tvoria oscilátory s periódou 4 často dominantnú časť všetkých objavených oscilujúcich vzorov, čo pravdepodobne súvisí s interakciou medzi dĺžkou polčasu rozpadu buniek (prechod  $1 \rightarrow 0,5 \rightarrow 0$  trvá 2 kroky) a medzerou v pravidle prežitia.

Osobitný experiment sme venovali pravidlu B4-6/S1-3,4-6, pri ktorom sme sa zamerali na hĺbkové hľadanie vzorov. Hoci sme v tomto pravidle neidentifikovali žiadnu loď, zaznamenali sme výrazný výskyt oscilátorov. V rozsahu periód 16 až 200 nástroj našiel oscilujúce vzory pre 180 z 185 hodnôt; chýbali iba periódy 163, 179, 191, 193 a 197. Táto skutočnosť naznačuje, že pravidlo B4-6/S1-3,4-6 by mohlo byť omniperiodické, čo je vlastnosť, ktorá sa v celulárnych automatoch dokazuje len veľmi ťažko.

### 4.4.3 Morfológické vlastnosti vybraných vzorov

V závere tejto časti sa stručne zameriame aj na morfológické vlastnosti vybraných vzorov. Zaujímavým fenoménom, ktorý môžeme pozorovať pri vizuálnej analýze lodí v automatoch typu *Half-Life*, je vzťah medzi ich tvarom a smerom pohybu. Hoci sa pohybujeme v diskretnom svete celulárnych automatov, kde neplatia zákony aerodynamiky tekutín, evolučné pravidlá často formujú lode do tvarov, ktoré nám intuitívne pripomínajú letiace objekty. V mnohých prípadoch možno rozlíšiť akési „čelo“ objektu, ktoré razí cestu, a „chvost“, tvorený bunkami postupne prechádzajúcimi do nižších stavov (z hodnoty 1 do 0,5 a následne 0).

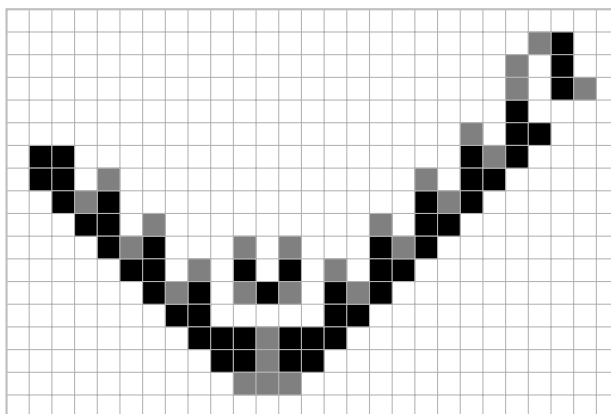
Túto heuristiku ilustruje séria vzorov uvedených na Obrázku 4.24. Pri každom z nich možno skúsiť odhadnúť smer pohybu iba na základe geometrie konfigurácie. V elektronickej verzii práce sú jednotlivé vzory opatrené hypertextovými odkazmi na aplikáciu *Fuzzy Life*, v ktorej je možné zobrazíť ich skutočný vývoj.



Obr. 4.24: Vybrané lode s odlišnou geometriou. Kliknutím na jednotlivé vzory možno overiť ich skutočný smer pohybu v simulácii.

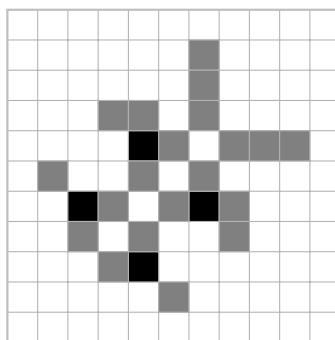
Treba však zdôrazniť, že táto vizuálna heuristika neplatí absolútne. Pri niektorých vzoroch sa smer pohybu zhoduje s intuitívnym očakávaním (napríklad pri vzore **B**), avšak existujú aj prípady, ktoré pôsobia výrazne kontraintuitívne. Ukázkovým príkladom je vzor **F**. Jeho geometria nápadne pripomína lietadlo s jasne definovaným „nosom“ a „krídlami“, čo zvädza k predpokladu pohybu smerom nadol. V skutočnosti sa však tento objekt pohybuje smerom nahor.

Ďalším príkladom vzoru, ktorého dynamika a tvar evokujú javy z reálneho sveta, je klzák na Obrázku 4.25. Ten pri svojom pohybe vytvára a zachováva charakteristický tvar pripomínajúci let krdla vtákov v tvare klina (V-formácie).



Obr. 4.25: Klzák s rýchlosťou  $c/2$  (perióda 16, posun 8) v pravidle B2.5-3/S3-4.5 s obrysom v tvare písmena V.

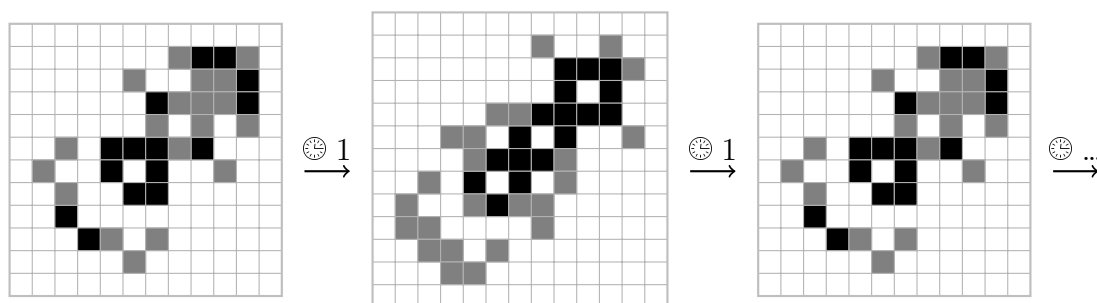
Takáto vizuálna podobnosť s prirodzenými javmi je v celulárnych automatoch pomerne častá a často vedie k intuitívnemu pomenovaniu vzorov podľa objektov z reálneho sveta. Výraznú vizuálnu dynamiku vykazuje aj ďalšia objavená loď, ktorej tvar pripomína drobného chrobáka s vystretými tykadlami. Táto analógia spolu so schopnosťou samostatného pohybu prirodzene evokuje predstavu jednoduchého organizmu.<sup>1</sup>



Obr. 4.26: „Chrobák“ v pravidle B1.5-2/S2.

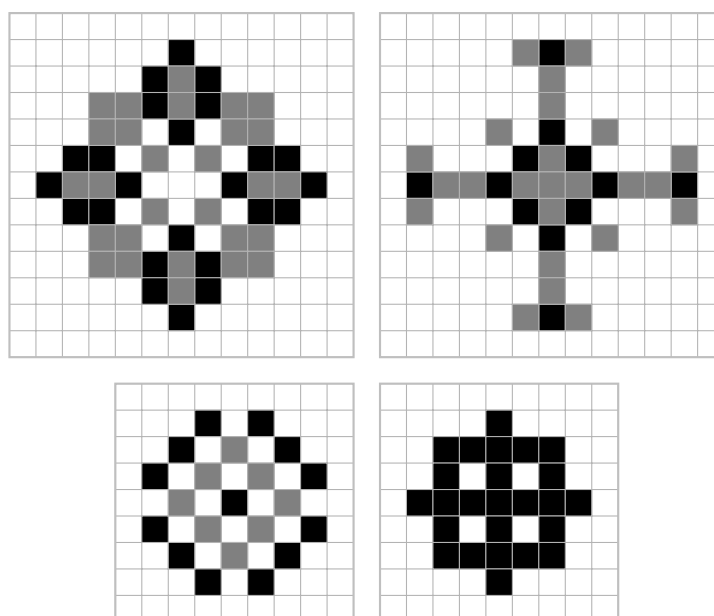
<sup>1</sup>Pozoruhodné však je, že vzor sa priestorom pohybuje opačným smerom.

Ďalším príkladom je vzor na Obrázku 4.27. Hoci ide o jeden koherentne sa pohybujúci objekt, jeho konfigurácia sa skladá z viacerých priestorovo oddelených častí, ktoré zostávajú počas cyklu koordinované.



Obr. 4.27: Viackomponentový klzák s rýchlosťou  $c/2$  v pravidle B1.5-2/S2.5-4.5. Má periódu 4 a posun o 2 bunky.

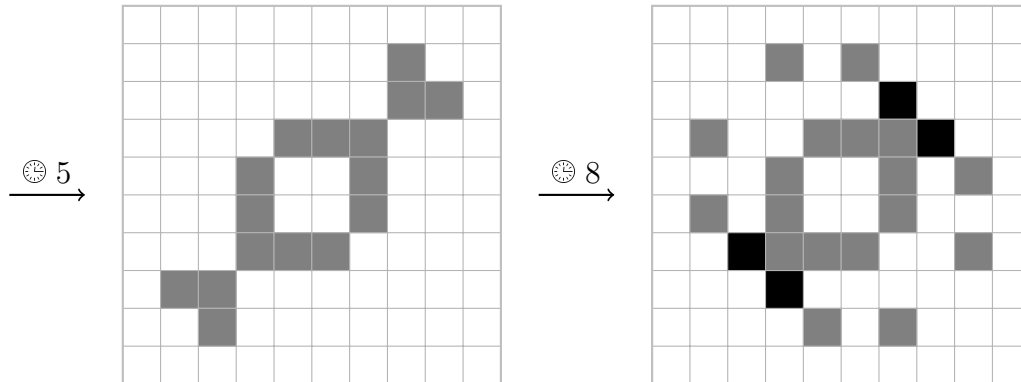
Aj medzi stabilnými, na mieste oscilujúcimi vzormi možno nájsť vzory s výraznou geometrickou regularitou. Oscilátor na Obrázku 4.28 je zaujímavý kombináciou radiálnej symetrie a relatívne dlhej periódy vzhľadom na svoje rozmery. Počas cyklu strieda viacero fáz so zachovanou stredovou súmernosťou. V určitých okamihoch pripomína geometrickú rozetu či zložitý ornament, inokedy nadobúda tvar hviezdy alebo kríža.



Obr. 4.28: Symetrický oscilátor v pravidle B3-8/S2-3,5-6 zachytený v časoch  $t = 6, 8, 13$  a 28. Vzhľadom na periódu 32 tu neuvádzame všetky jeho fázy.

Do kategórie nezvyčajných a esteticky pútavých vzorov radíme aj kompaktný oscilátor s periódou 16. Jeho vývoj je charakteristický výraznou premenou tvarov – kým v jednej

fáze pripomína zabalený cukrík, v inej jeho dynamika priamo evokuje bunkové delenie (mitózu).



Obr. 4.29: Oscilátor s periódou 16 v pravidle B2-7.5/S0.5-1 zachytený v dvoch morfolo- gicky odlišných fázach.

## 4.5 Integrácia výsledkov do aplikácie Fuzzy Life

Výstupy, ktoré sme vygenerovali analytickým nástrojom, neslúžia iba ako štatistický prehľad, ale aj ako dátový vstup pre webovú aplikáciu *Fuzzy Life*. Do aplikácie sme integrovali pravidlá a vzory, ktoré sme identifikovali počas experimentov, čím vznikla knižnica vzorov získaných offline prehľadávaním priestoru pravidiel. Aplikácia tieto vzory sprístupňuje prostredníctvom preddefinovanej knižnice vzorov „Pattern Presets“. *Half-Life Explorer* tak zabezpečuje prepojenie medzi výpočtovo náročným prehľadávaním stavového priestoru a následným zobrazením výsledkov v interaktívnej aplikácii.

## 4.6 Porovnanie s existujúcimi nástrojmi

Komunita okolo Hry života vytvorila v priebehu desaťročí množstvo nástrojov určených na simuláciu a analýzu celulárnych automatov. Portál ConwayLife Wiki uvádza rozsiahly zoznam softvéru [28], ktorý pokrýva rôzne potreby od vizualizácie po automatizované vyhľadávanie vzorov.

Pre naše porovnanie sú relevantné najmä nástroje Golly a apgsearch. Golly je orientovaný predovšetkým na simuláciu veľkých mriežok, pričom využíva algoritmus HashLife a podporuje skriptovanie v jazykoch Lua a Python. Nástroj apgsearch je zameraný na automatizované prehľadávanie pravidiel a katalogizáciu vzorov v prepojení s online databázou Catagolue.

Z hľadiska nášho modelu však tieto nástroje nepokrývajú všetky požadované vlastnosti priamo. Väčšina existujúcich vyhľadávacích programov je optimalizovaná pre štandardné rodiny pravidiel (*Life-like*, *Generations*) a binárne reprezentácie stavov. Náš trojstavový model so špecifickým mechanizmom „vyhladzovania“ a so zlomkovými hodnotami preto vyžadoval odlišnú reprezentáciu stavov aj iný spôsob vyhodnocovania pravidiel. Adapácia existujúcich riešení by si v takomto prípade vyžadovala zásah do nízkoúrovňových optimalizácií, často založených na bitových operáciách a špecifických inštrukčných sadách.

Vlastná implementácia v jazyku Rust bola potrebná najmä na vytvorenie nástroja, ktorý by zodpovedal špecifickým požiadavkám skúmaného modelu, a to predovšetkým v týchto oblastiach:

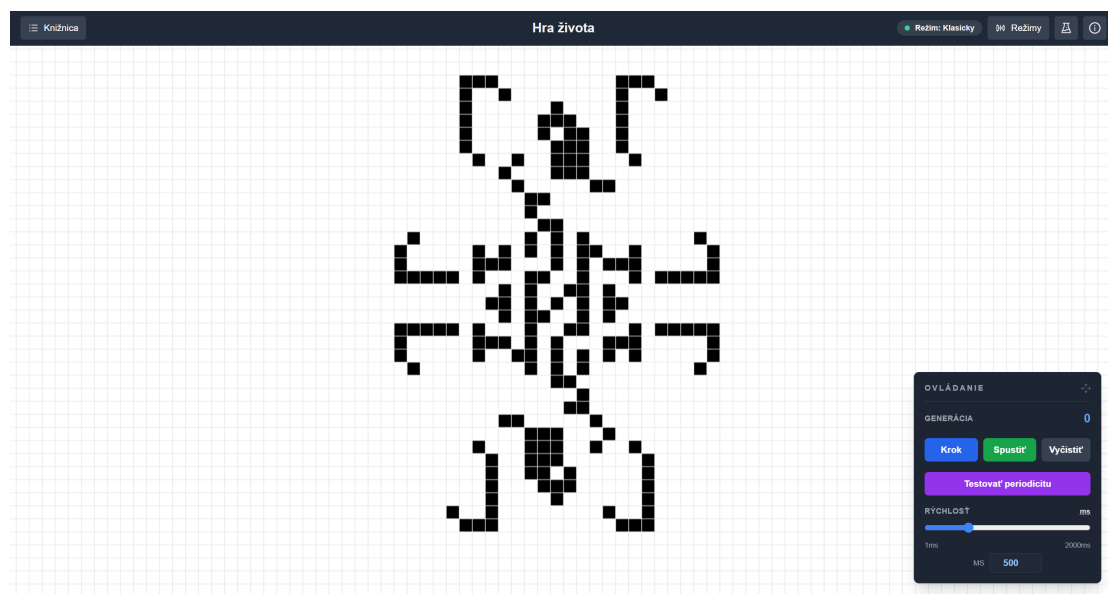
- **Pamäťová reprezentácia.** Umožnila plnú kontrolu nad reprezentáciou stavov v pamäti aj nad operáciami nad nimi.
- **Heuristiky.** Umožnila navrhnúť heuristiky prispôsobené skúmaným pravidlám a parametrom.
- **Paralelné spracovanie.** Umožnila spracúvať výpočty bez nutnosti prispôbovať sa obmedzeniam všeobecných simulátorov.

# Kapitola 5

## Webová aplikácia Fuzzy Life

Na účely praktického skúmania a vizualizácie nielen automatu *Half-Life*, ale aj ďalších zovšeobecnení Hry života, sme vyvinuli interaktívnu webovú aplikáciu *Fuzzy Life*.<sup>1</sup> Cieľom projektu je poskytnúť interaktívne prostredie na pozorovanie správania rôznych tried celulárnych automatov, od klasického binárneho modelu až po jeho spojité a viacstavové rozšírenia.

Na Obrázku 5.30 je zobrazené hlavné používateľské rozhranie webovej aplikácie.



Obr. 5.30: Hlavné používateľské rozhranie webovej aplikácie *Fuzzy Life*. Dominantným prvkom je interaktívna simulačná mriežka, ktorú dopĺňajú ovládacie prvky na riadenie behu simulácie a konfiguračný panel pre voľbu režimov.

<sup>1</sup>Dostupná online na adrese: <https://fuzzylife.netlify.app/>.

## 5.1 Architektúra aplikácie

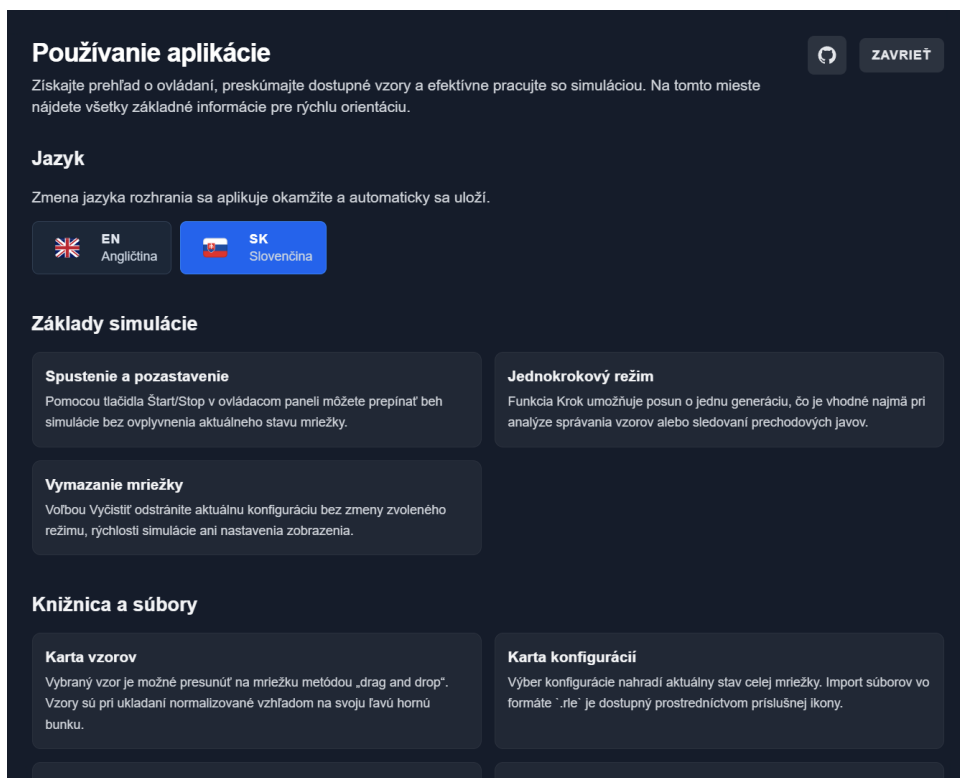
Aplikácia *Fuzzy Life* je realizovaná ako jednostránková webová aplikácia (SPA) postavená na knižnici React. Vizualizácia simulácie prebieha prostredníctvom HTML5 Canvas API, ktoré je vhodné na priebežné vykresľovanie mriežkových simulácií v prehliadači. Hlavné komponenty architektúry sú:

- **Simulačné jadro.** Centrálny komponent `App.js` riadi hlavnú simulačnú slučku. Funkcia `nextGeneration` prechádza všetky bunky mriežky a aplikuje na ne prechodovú funkciu `computeNextState` aktuálne zvoleného režimu. Aktualizácia buniek prebieha v každom kroku simulácie synchronne.
- **Správa stavu.** Na správu globálneho stavu aplikácie sa používa knižnica Zustand. Lokálny stav komponentov je spravovaný prostredníctvom React Hooks (`useState`, `useEffect`, `useRef`, `useCallback`). Pre správu asynchrónnych udalostí (napríklad používateľského vstupu pri kreslení na mriežku) sa využíva knižnica RxJS.
- **Dátová vrstva a integrácia vzorov.** Aplikácia načítava objavené lode a oscilátory zo statického súboru `patterns.js`, ktorý je automatizovaným výstupom offline analytického nástroja *Half-Life Explorer*. Súbor obsahuje RLE definície prepojené s konkrétnymi pravidlami.
- **Vizualizácia.** Mriežka je vykreslená na plátno pomocou HTML5 Canvas API, ktoré podporuje posúvanie (`pan`) a priblíženie/oddialenie (`zoom`). Každý simulačný režim implementuje vlastnú metódu `renderCell`, ktorá definuje spôsob vykresľovania buniek – napríklad odtiene sivej pre spojité stavy alebo farebne kódované úrovne pre diskkrétne viacstavové režimy.
- **Internacionalizácia.** Aplikácia je plne lokalizovaná do angličtiny a slovenčiny pomocou knižníc `i18next` a `react-i18next`. Preložené sú všetky texty používateľského rozhrania vrátane opisov režimov, názvov vzorov a nápovedy.

Z hľadiska softvérovej architektúry je kľúčovým návrhovým prvkom abstraktná trieda `LifeMode`, z ktorej dedia všetky simulačné režimy. Táto trieda definuje spoločné rozhranie:

- `computeNextState(grid, x, y)` – výpočet ďalšieho stavu bunky,
- `renderCell(ctx, x, y, value)` – vykreslenie bunky,
- serializáciu a deserializáciu vzorov vo formáte RLE.

Tento polymorfný prístup umožňuje jednoduché pridávanie nových režimov bez nutnosti zásahov do hlavnej simulačnej slučky.

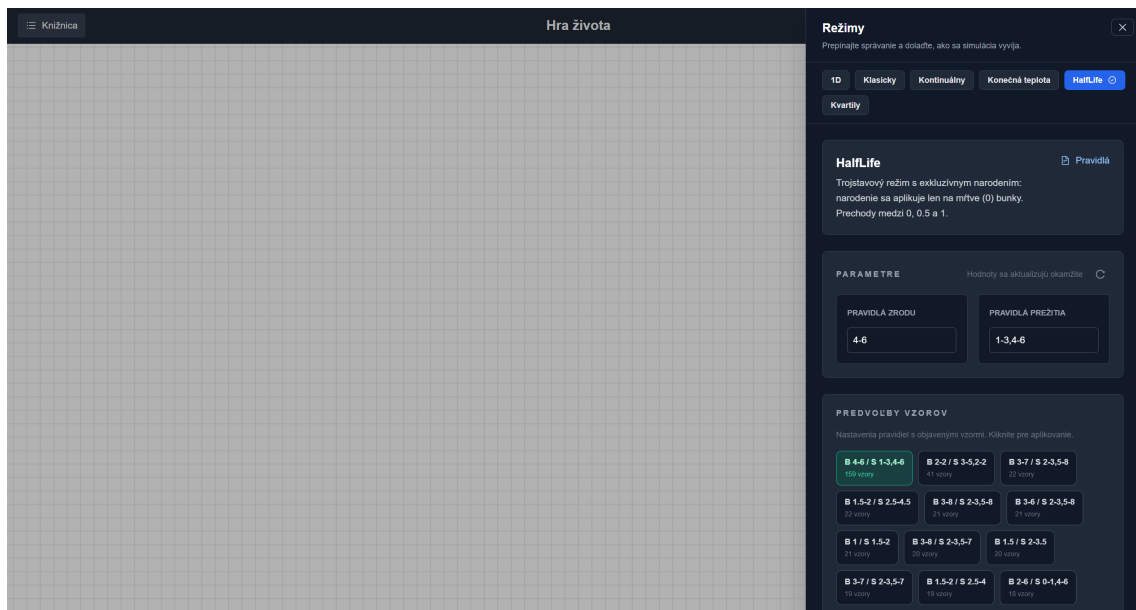


Obr. 5.31: Používateľská príručka integrovaná priamo v prostredí aplikácie. Okno prehľadne sumarizuje kľúčové funkcionality vrátane jednokrokového režimu pre analýzu prechodov, manipulácie so vzormi pomocou techniky *drag-and-drop* a možností lokálneho úložiska.

## 5.2 Simulačné režimy

Aplikácia implementuje celkovo 6 simulačných režimov, z ktorých každý definuje odlišný stavový priestor, okolie a prechodovú funkciu. Všetky dvojrozmerné režimy využívajú Moorovo okolie s 8 susedmi.

Používateľské rozhranie pre výber a detailnú konfiguráciu týchto režimov, vrátane nastavenia pravidiel zrodu, prežitia a predvolieb vzorov pre režim Half-Life, je znázornené na Obrázku 5.32.



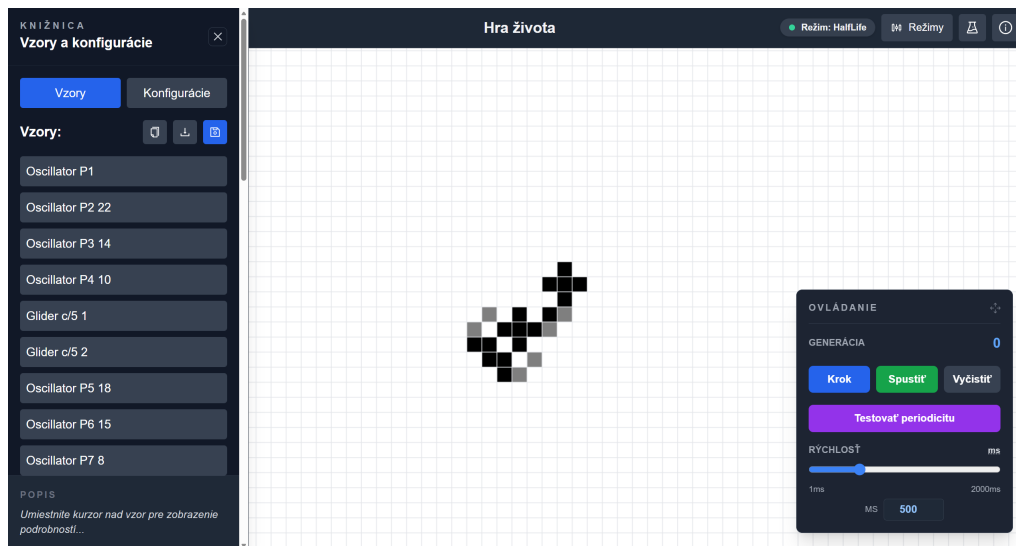
Obr. 5.32: Bočný panel aplikácie *Fuzzy Life* pre správu simulačných režimov. Obrázok dokumentuje rozhranie pre prepínanie medzi šiestimi dostupnými modelmi a interaktívne nastavovanie parametrov pravidiel pre aktuálne zvolený režim Half-Life.

### 5.2.1 Klasický režim

Klasický režim implementuje pôvodnú Conwayovu Hru života s pravidlom B3/S23. Prechodová funkcia presne zodpovedá deterministickým binárnym pravidlám. Z hľadiska aplikácie slúži tento režim predovšetkým ako referenčná implementácia najjednoduchšieho podporovaného modelu. Stav mriežky je možné udržiavať v jednoduchých binárnych, respektíve celočíselných štruktúrach a pri vykresľovaní prostredníctvom Canvas API sa stavy priamo mapujú na čiernu a bielu farbu (alebo definované témy) bez potreby výpočtov farebných prechodov či priesvitnosti.

### 5.2.2 Režim Half-Life

Tento režim implementuje jadro našej práce – trojstavový model s pravidlami pre zrod, prežitie a postupným prechodom. Systém poskytuje plnú parametrizáciu B/S intervalov a podporuje aj rozšírené pravidlá s nesúvislými intervalmi prežitia (tzv. *split-survival*), napríklad voliteľný druhý interval  $S_2$ . Významnou súčasťou tohto režimu je integrovaná knižnica predvolieb, ktorá sprístupňuje všetky objavené pravidlá, v ktorých sme našli aspoň jeden klzák.



Obr. 5.33: Knižnica predvolených pravidiel a vzorov v režime *Half-Life*. Rozhranie umožňuje vyberať pravidlá, pri ktorých boli automatizovaným prieskumom nájdené netriviálne vzory. Získané vzory je možné interaktívne vkladať priamo do simulačného priestoru.

### 5.2.3 Spojitý režim

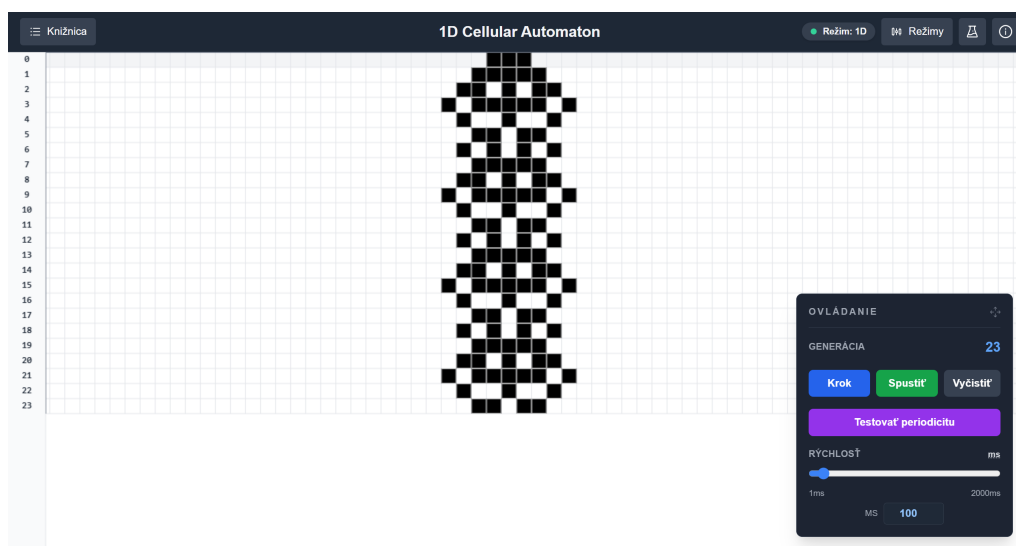
Spojité režim (*Continuous*) vizualizuje myšlienku fuzzy logiky, kde bunky nenadobúdajú len diskkrétne stavy, ale plynulé hodnoty intenzity v intervale od 0 do 1. Namiesto striktného okamžitého prepisovania stavov systém aplikuje parametre útlmu (*decay*) a príťažlivosti, čo vedie k plynulému rozsvecovaniu a zhasínaniu buniek. Z implementačného hľadiska si to vyžaduje udržiavanie stavu v dátových poliach s pohyblivou rádovou čiarkou (napríklad `Float32Array`). V rámci renderovacieho cyklu sa tieto spojité hodnoty prepočítavajú na alfa kanál (priesvitnosť) vrstvy alebo na odtiene sivej farby, čo na obrazovke vytvára vizuálne difúzne stopy a doznievajúce „chvosty“ za pohybujúcimi sa objektmi.

### 5.2.4 Režim konečnej teploty

Režim konečnej teploty implementuje logistický energetický model opísaný v sekcii 2.5. V aplikácii slúži ako parametrizovaná verzia tohto modelu, v ktorej môže používateľ priamo meniť teplotu  $T$ , energetický posun  $E_0$  a posun stavu  $x_0$ . Keďže bunky nadobúdajú spojité hodnoty v intervale  $[0, 1]$ , režim využíva rovnaký typ dátovej reprezentácie a vizualizácie odtieňmi ako ostatné spojité modely.

## 5.2.5 Jednorozmerný režim (1D)

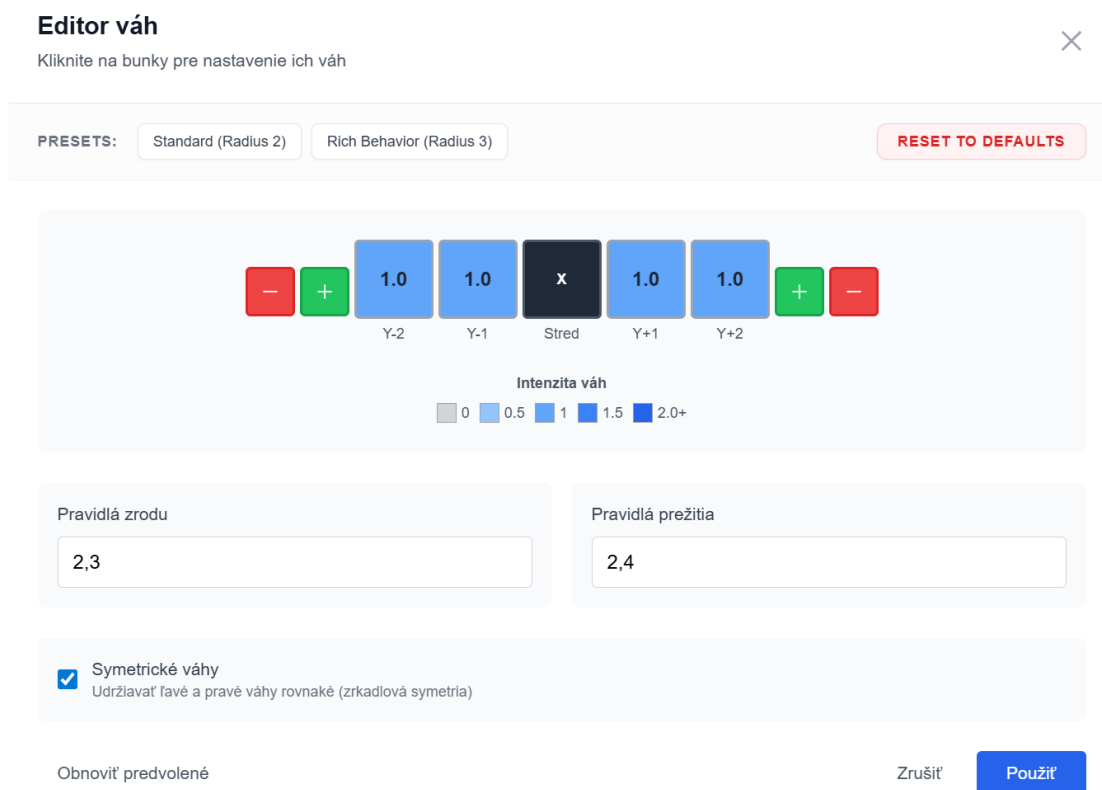
Tento režim implementuje jednorozmerný celulárny automat inšpirovaný Millenovými pravidlami, ktorého teoretický základ bol opísaný v sekcii 2.2. Z implementačného a vizualizačného hľadiska ide o osobitný prípad. Výpočet prebieha iba nad aktuálnym riadkom buniek, takže aplikácia namiesto plnohodnotnej dvojrozmernej mriežky v každom kroku vytvára nový riadok z lokálneho susedstva každej bunky. Vizualizácia pritom predchádzajúce stavy neprepisuje, ale ukladá ich na plátno po riadkoch smerom nadol. Vzniká tak dvojrozmerný časopriestorový diagram, kde horizontálna os reprezentuje priestor a vertikálna os čas. Tento spôsob vykresľovania umožňuje na prvý pohľad sledovať periodicitu oscilátorov aj trajektórie posuvných vzorov v podobe diagonálnych línií. Obrázok 5.34 zobrazuje jednorozmerný režim a jeho časopriestorovú vizualizáciu.



Obr. 5.34: Jednorozmerný režim v aplikácii *Fuzzy Life*. Vývoj systému je zobrazený ako časopriestorový diagram, v ktorom horizontálna os reprezentuje priestor a vertikálna os čas.

Na rozdiel od pevne daného teoretického modelu je implementácia v aplikácii zámerne parametrizovaná. Predvolená konfigurácia zodpovedá Millenovmu pravidlu so susedstvom typu  $YYXYY$  a pravidlami zrodu  $B = \{2, 3\}$  a prežitia  $S = \{2, 4\}$ , no používateľ môže meniť veľkosť susedstva na oboch stranách stredovej bunky, a teda prechádzať od tvaru  $YYXYY$  k širším jadrom typu  $YYYXYYY$  a podobne. Okrem jednoduchého počítania živých susedov aplikácia podporuje aj vážený režim, v ktorom má každý ľavý a pravý sused samostatne nastaviteľnú váhu. Pravidlá zrodu a prežitia preto možno zadávať nielen ako diskkrétne počty susedov, ale aj ako intervaly prípustného váženého súčtu, vrátane

neceločíselných hodnôt. Možnosti parametrizácie jednorozmerného režimu, vrátane editora váh a predvolieb pre rôzne polomery susedstva, sú zobrazené na Obrázku 5.35.



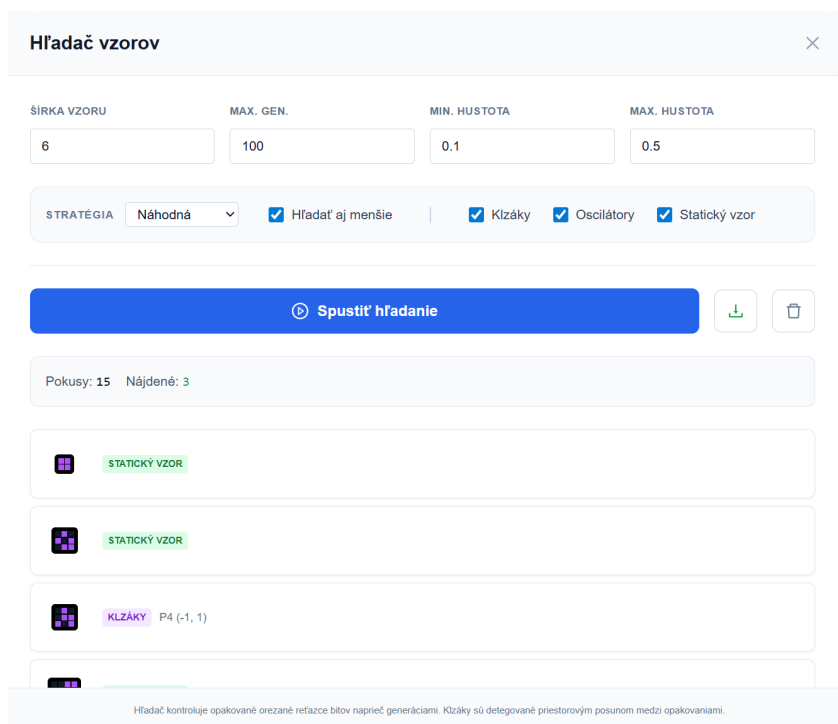
Obr. 5.35: Editor váh jednorozmerného režimu v aplikácii *Fuzzy Life*. Rozhranie umožňuje nastavovať váhy ľavých a pravých susedov, pravidlá zrodu a prežitia, ako aj zapnúť alebo vypnúť zrkadlovú symetriu medzi oboma stranami. Súčasťou sú aj predvolby pre štandardné a rozšírené susedstvo.

### 5.2.6 Režim Quartiles

Režim *Quartiles* je priame zovšeobecnenie nášho modelu *Half-Life*. *Quartiles* rozdeluje stavový priestor presne na päť pevných úrovní (0; 0,25; 0,5; 0,75; 1). Tento režim bol pôvodne navrhnutý aj ako implementačný kompromis medzi úplne spojitým modelom a striktnými diskretnými modelmi. Z hľadiska vykresľovania prostredníctvom Canvas API to znamená jednoduchšiu reprezentáciu farieb: aplikácia si predpočíta paletu piatich konkrétnych odtieňov (farieb) a bunkám len priraduje príslušný index z tejto palety. Eliminuje sa tým nutnosť dynamického prepočítavania RGBA reťazcov v každom snímku pre každú bunku. Keďže tento päťstavový model otvára ešte širší priestor pravidiel a parametrov než *Half-Life*, je sľubným smerom pre budúci výskum.

## 5.3 Interaktívny vyhľadávač vzorov

Hoci masívne paralelné hľadanie spadá pod kompetenciu natívneho analytického nástroja v jazyku Rust, webová aplikácia obsahuje aj klientsky *Vyhľadávač vzorov* („Pattern Finder“) zobrazený na Obrázku 5.36. Ide o lokálny simulátor bežiaci priamo v prehliadači, ktorý umožňuje automatizovane objavovať zaujímavé štruktúry v aktuálne nastavenom celulárnom automate.



Obr. 5.36: Rozhranie nástroja *Vyhľadávač vzorov* v aplikácii *Fuzzy Life*.

Používateľ môže nastaviť počiatočné podmienky vyhľadávania:

- **Priestor a čas.** Šírka počiatočného náhodného zhľuku (napr.  $6 \times 6$  buniek s možnosťou hľadania aj menších vzorov) a maximálny počet generácií (napr. 100 krokov), počas ktorých sa má vzor vyhodnocovať.
- **Hustota.** Minimálna a maximálna počiatočná hustota živých buniek (napr. interval od 0,1 do 0,5). Toto nastavenie pomáha predísť generovaniu vzorov, ktoré by okamžite zanikli v dôsledku izolácie alebo nadmerného počtu susedov.
- **Stratégia a ciele.** Výber konkrétnych kategórií vzorov, ktoré chce používateľ hľadať (klzáky, oscilátory, statické vzory) s využitím generatívnej stratégie, napríklad náhodnej („Random“).

Po spustení algoritmus na pozadí neustále iteruje cez náhodné konfigurácie a v reálnom čase zobrazuje štatistiku (počet uskutočnených pokusov a počet úspešných nálezov). Keď systém nájde štruktúru zodpovedajúcu zadaným kritériám, zaradí ju do výsledkového zoznamu. Aplikácia následne umožňuje jej zobrazenie na hlavnom plátne simulátora a ďalšie skúmanie jej správania.

# Záver

V tejto práci sme preskúmali Conwayovu Hru života z teoretického aj experimentálneho hľadiska a zároveň rozšírili jej klasický model o nové myšlienky vychádzajúce z konceptu fuzzifikácie. V úvodnej časti práce sme predstavili historický kontext vzniku Hry života, jej formálne definície a vybrané matematické vlastnosti, ktoré stoja za jej prekvapivo bohatou dynamikou. Napriek veľmi jednoduchým lokálnym pravidlám dokáže tento celulárny automat generovať široké spektrum komplexného správania, čo z neho robí zaujímavý objekt matematického aj experimentálneho skúmania.

Hlavným prínosom práce je návrh a analýza nového trojstavového celulárneho automatu s názvom *Half-Life*. Tento model rozširuje klasickú dvojstavovú Hru života o medzistav. Zavedenie tretieho stavu vedie k odlišnej dynamike systému a umožňuje pozorovať nové typy vzorov a prechodových javov, ktoré sa v klasickej verzii nevyskytujú alebo sú výrazne zriedkavejšie. Automatizovaný prieskum viedol k vytvoreniu finálneho deduplikovaného katalógu 514 unikátnych pravidiel a 3 381 vzorov. Tieto výsledky empiricky ukazujú, že aj pri relatívne jednoduchých pravidlách môže model *Half-Life* generovať zaujímavé stabilné aj pohyblivé štruktúry.

V práci sme vytvorili aj webové prostredie určené na experimentovanie s celulárnymi automatmi. Implementovaný systém umožňuje interaktívne skúmanie rôznych pravidiel, vizualizáciu vývoja konfigurácií a jednoduché testovanie nových modelov. Takéto prostredie sa ukázalo ako užitočný nástroj nielen pri vývoji automatu *Half-Life*, ale aj pri všeobecnom skúmaní správania celulárnych automatov.

Možným smerom ďalšieho výskumu je podrobnejšia matematická analýza vlastností navrhnutého automatu, napríklad štúdium jeho stabilných vzorov, pohybujúcich sa vzorov alebo otázok súvisiacich s existenciou špecifických tried vzorov. Presnejšie posúdenie vplyvu nesúvislých intervalov prežitia by si vyžadovalo normalizované porovnanie úspešnosti jednotlivých tried pravidiel vzhľadom na veľkosť prehladávaného priestoru. Zaujímavé by bolo aj systematické porovnanie dynamiky modelu *Half-Life* s inými rozšíreniami Hry života alebo jeho ďalšie zovšeobecnenie.

# Zoznam použitej literatúry

- [1] GARDNER, Martin. The Fantastic Combinations of John Conway’s New Solitaire Game “Life”. *Scientific American*. 1970, roč. 223, č. 4, s. 120–123. ISSN 0036-8733. Dostupné z DOI: [10.1038/scientificamerican1070-120](https://doi.org/10.1038/scientificamerican1070-120).
- [2] JOHNSTON, Nathaniel; GREENE, Dave. *Conway’s Game of Life: Mathematics and Construction*. Self-published, 2022. ISBN 978-1-794-81696-1. Dostupné z DOI: [10.5281/zenodo.6097284](https://doi.org/10.5281/zenodo.6097284).
- [3] BERLEKAMP, Elwyn R.; CONWAY, John H.; GUY, Richard K. *Winning Ways for Your Mathematical Plays, Vol. 2: Games in Particular*. London: Academic Press, 1982. ISBN 0-12-091102-7.
- [4] COOK, Matthew. Still Life Theory. In: *New Constructions in Cellular Automata*. Oxford University Press, 2003. ISBN 9780195137170. Dostupné z DOI: [10.1093/oso/9780195137170.003.0008](https://doi.org/10.1093/oso/9780195137170.003.0008).
- [5] CEPELEWICZ, Jordana. *Math’s ‘Game of Life’ Reveals Long-Sought Repeating Patterns* [online]. Quanta Magazine, 2024-01-18. [cit. 2026-03-01]. Dostupné z: <https://www.quantamagazine.org/maths-game-of-life-reveals-long-sought-repeating-patterns-20240118/>.
- [6] APG. *Elementary knightship* [online]. Fórum ConwayLife.com, 2018-03-06. [cit. 2026-03-17]. Dostupné z: <https://conwaylife.com/forums/viewtopic.php?t=3303>. Diskusné vlákno na fóre.
- [7] ZDR. *is this c/10 spaceship known?* [online]. Fórum ConwayLife.com, 2016-03-05. [cit. 2026-02-25]. Dostupné z: <https://conwaylife.com/forums/viewtopic.php?f=2&t=2057&p=27988#p27988>. Príspevok na fóre.
- [8] MOORE, Edward F. Machine models of self-reproduction. In: *Proceedings of Symposia in Applied Mathematics*. American Mathematical Society, 1962, zv. 14, s. 17–33.

- [9] MYHILL, John. The converse of Moore's Garden-of-Eden theorem. *Proceedings of the American Mathematical Society*. 1963, roč. 14, č. 4, s. 685–686. Dostupné z DOI: 10.1090/S0002-9939-1963-0155764-9.
- [10] WAINWRIGHT, Robert T. Life is universal! In: *Proceedings of the 7th Conference on Winter Simulation – Volume 2*. Washington, DC: Winter Simulation Conference, 1974, s. 449–459. WSC '74. Dostupné z DOI: 10.1145/800290.811303.
- [11] CARLINI, Nicholas. *Building Digital Logic in the Game of Life* [online]. 2020-04-01. [cit. 2026-02-26]. Dostupné z: <https://nicholas.carlini.com/writing/2020/digital-logic-game-of-life.html>.
- [12] DURAND, Bruno; RÓKA, Zsuzsanna. The Game of Life: Universality Revisited. In: DELORME, M.; MAZOYER, J. (ed.). *Cellular Automata: A Parallel Model*. Dordrecht: Springer Netherlands, 1999, s. 51–74. ISBN 978-94-015-9153-9. Dostupné z DOI: 10.1007/978-94-015-9153-9\_2.
- [13] RENDELL, Paul. Turing Universality of the Game of Life. In: ADAMATZKY, Andrew (ed.). *Collision-Based Computing*. London: Springer London, 2002, s. 513–539. ISBN 978-1-4471-0129-1. Dostupné z DOI: 10.1007/978-1-4471-0129-1\_18.
- [14] RENDELL, Paul. A Universal Turing Machine in Conway's Game of Life. In: *2011 International Conference on High Performance Computing & Simulation*. 2011, s. 764–772. Dostupné z DOI: 10.1109/HPCSim.2011.5999906.
- [15] RENDELL, Paul. *Turing Machine Universality of the Game of Life*. Zv. 18. Switzerland: Springer Cham, 2015. Emergence, Complexity and Computation. ISBN 978-3-319-19842-2. Dostupné z DOI: 10.1007/978-3-319-19842-2. 1. vydanie.
- [16] GOUCHER, Adam P. *Catagolue: Life-like cellular automata* [online]. Catagolue. [cit. 2026-03-17]. Dostupné z: <https://catagolue.hatsya.com/rules/lifelike>. Webová stránka venovaná Life-like pravidlám.
- [17] MILLEN, Jonathan K. One-Dimensional Life. *BYTE Magazine*. 1978, roč. 3, č. 12, s. 68–74. ISSN 0360-5280.
- [18] WOLFRAM, Stephen. *A New Kind of Science*. Wolfram Media, 2002. ISBN 1-57955-008-8. Dostupné z: <https://www.wolframscience.com/nks/>.
- [19] BAYS, Carter. Candidates for the Game of Life in Three Dimensions. *Complex Systems*. 1987, roč. 1, č. 3, s. 373–400. ISSN 0891-2513. Dostupné z: [https://www.complex-systems.com/abstracts/v01\\_i03\\_a01/](https://www.complex-systems.com/abstracts/v01_i03_a01/).

- [20] TOFFOLI, Tommaso; MARGOLUS, Norman. *Cellular Automata Machines: A New Environment for Modeling*. 1987. ISBN 978-0-262-20060-8. Dostupné z DOI: 10.7551/mitpress/1763.001.0001.
- [21] MATVEYEV, Yoel. *Re: Brian's Brain* [online]. Fórum ConwayLife.com, 2020-08-25. [cit. 2026-03-01]. Dostupné z: <https://conwaylife.com/forums/viewtopic.php?&p=110762#p110762>. Príspevok na fóre.
- [22] ADACHI, Susumu; PEPPER, Ferdinand; LEE, Jia. The Game of Life at finite temperature. *Physica D: Nonlinear Phenomena*. 2004, roč. 198, č. 3, s. 182–196. ISSN 0167-2789. Dostupné z DOI: 10.1016/j.physd.2004.04.010.
- [23] EVANS, Kellie M. Larger than Life: Digital Creatures in a Family of Two-Dimensional Cellular Automata. *Discrete Mathematics & Theoretical Computer Science*. 2001, roč. DMTCS Proceedings vol. AA, Discrete Models: Combinatorics, Computation, and Geometry (DM-CCG 2001). ISSN 1365-8050. Dostupné z DOI: 10.46298/dmtcs.2288.
- [24] RAFLER, Stephan. *Generalization of Conway's "Game of Life" to a continuous domain – SmoothLife*. 2011. Dostupné z arXiv: 1111.1567 [nlin.CG].
- [25] CHAN, Bert Wang-Chak. Lenia: Biology of Artificial Life. *Complex Systems*. 2019, roč. 28, č. 3, s. 251–286. ISSN 0891-2513. Dostupné z DOI: 10.25088/complexsystems.28.3.251.
- [26] MRAZ, M.; ZIMIC, N.; LAPANJA, I.; BAJEC, I. Fuzzy cellular automata: from theory to applications. In: *Proceedings 12th IEEE International Conference on Tools with Artificial Intelligence*. 2000, s. 320–323. Dostupné z DOI: 10.1109/TAI.2000.889889.
- [27] ATANASSOVA, Liliya; ATANASSOV, Krassimir. Intuitionistic Fuzzy Interpretations of Conway's Game of Life. In: DIMOV, Ivan; DIMOVA, Stefka; KOLKOVSKA, Natalia (ed.). *Numerical Methods and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, s. 232–239. ISBN 978-3-642-18466-6.
- [28] PRISPIEVATELIA LIFEWIKI. *Tutorials/Software* [online]. ConwayLife.com, 2023 [cit. 2026-03-01]. Dostupné z: <https://conwaylife.com/wiki/Tutorials/Software>. Wiki stránka.