MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Marie Vestenická

Labyrint města a ráj parkovacích úloh

Katedra didaktiky matematiky

Vedoucí bakalářské práce: doc. RNDr. Antonín Slavík, Ph.D.

Studijní program: Matematika se zaměřením na vzdělávání

Studijní obor: Matematika se zaměřením na vzdělávání se sdruženým studiem Francouzský jazyk a literatura se z.n.vzd

Praha 2023

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

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

V .... dne ....

Podpis autora

Cesty života jsou nevyzpytatelné, a proto když se ve čtvrtek prvního letního týdne roku 1980 narodilo malé dítko, pravděpodobně nikoho ani nenapadlo, že se jednoho dne stane nadšeným matematikem. Kdyby se tento člověk narodil o 350 let dříve a o kousek dál, s jeho vytrvalostí a odhodláním by se mohl hravě ucházet o místo mušketýra u francouzského krále. Kdybychom v čase popošli ještě o pár let zpět, mohl by se inspirovat Rubensem, Caravaggiem či Rembrandtem a jeho pečlivost, důslednost a smysl pro detail by mu umožnily pronásledovat kariérní dráhu umělce. Kdyby se naopak narodil o pár desítek let později, jeho touha po poznání by ho možná zavedla až na povrch jiných planet. Kdybychom se dostali třeba i do jiného vesmíru, možná by byl bystrým mágem, který jakoby vypadl z příběhů Terryho Pratchetta.

Ať už by tento člověk napříč časem a prostorem tíhnul k jakékoli profesi, jsem ráda, že teď a tady se jmenuje doc. Antonín Slavík, a jeho cesta vedla k matematice. Děkuji tímto jeho mentorům doc. Janě Staré, prof. Ivanu Netukovi, prof. Štefanu Schwabikovi a doc. Jindřichu Bečvářovi, že ho na cestě provázeli. Rovněž jsem vděčná i jeho kolegům na katedře didaktiky, kteří mu pomáhají, a tedy má čas vést práce jako je tato. V neposlední řadě cením rozvážnost jeho rodičů, protože přestože by byl díky lidskému a zároveň preciznímu přístupu opravdu skvělým lékařem, zachraňuje životy jinde a věřím, že matematika ho potřebuje mnohem větší mírou. Nejvíce samozřejmě děkuji doc. Antonínu Slavíkovi, za výborné vedení, neuvěřitelnou trpělivost, otevřenou mysl a především za jeho skvělý smysl pro humor, jenž nechal mimo jiné vzniknout i tomuto poděkování.

Děkuji také mým drahým kamarádům Sarah Böhm a Karlovi Krausovi. Sarah za numerickou podporu ve chvílích, kdy jsem hledala vlastní řešení, která občas nikam nevedla. Kájovi za jeho podnětné otázky, které mě nutily o práci pochybovat a pomohly mi v přípravě mé přednášky.

Nejlepší prací matematika je umění, zdokonalené umění, troufalé jako nejtajnější sny představivosti, jasné a průzračné.
Gösta Mittag-Leffler

Název práce: Labyrint města a ráj parkovacích úloh

Autor: Marie Vestenická

Katedra: Katedra didaktiky matematiky

Vedoucí bakalářské práce: doc. RNDr. Antonín Slavík, Ph.D., Katedra didaktiky matematiky

Abstrakt: V této práci se čtenář seznámí s pojmem parkovací funkce. Další část textu je věnována Rényiho parkovacímu problému. Teorii doplňujeme příklady a programy, které jednotlivé parkovací úlohy lépe ilustrují.

Klíčová slova: parkovací funkce, střední hodnota, generující funkce, pravděpodobnost, Rényiho parkovací problém, diferenciální rovnice se zpožděním

Title: The Labyrinth of the City and the Paradise of Parking

Author: Marie Vestenická

Department: Department of Mathematics Education

Supervisor: doc. RNDr. Antonín Slavík, Ph.D., Department of Mathematics Education

Abstract: The aim of this thesis is to familiarize the reader with the notion of parking functions. The following section is dedicated to Rényi's parking problem. Theory goes hand in hand with examples and programs that better explain individual parking problems.

Keywords: parking functions, mean value, generating functions, probability, Rényi's parking problem, differential equation with delay

Obsah

Úvod

Každý den činíme mnoho rozhodnutí. Ať už zvažujeme, co budeme jíst k snídani, co si oblečeme ven nebo kde budeme pracovat, vše nás do větší či menší míry ovlivňuje. Jedno takové každodenní dilema, kterého se každý z nás účastní pasivně či aktivně, je volba parkování. Ačkoliv se volba parkovacího místa může zdát irelevantní, i pro takovou volbu nám život často nabídne spoustu možností. A jak se říkává: „Když nám život dává mnoho možností, zkusme je matematicky prozkoumat.“

Představme si například manželský pár, který vjíždí do jednosměrné ulice. Když projíždějí okolo stánku se zeleninou, manželka si najednou vzpomene, že je potřeba nakoupit spoustu brokolice na nedělní oběd. Okamžitě požádá manžela, aby zastavil na prvním volném místě. Manžel má samozřejmě brokolici rád, takže jí chce vyhovět. Pokud je místo po jeho pravé straně volné, zaparkuje, jestliže je místo obsazené, hledá další nejbližší volné místo. Možná by nás nepřekvapilo, kdyby právě v tuto chvíli manžel začal zkoumat všechny své možnosti parkování. Jelikož matematikové A. G. Konheim i B. Weiss měli také milé inspirativní manželky, publikovali v roce 1966 tuto úlohu v článku, který na první pohled nemá nic společného s parkováním, ale věnuje se datovým strukturám [8]. Tento typ úlohy položil základy pro pojem parkovací funkce, který v první kapitole zavedeme a objasníme ho na několika příkladech. Ačkoliv parkovací funkce mají kořeny v informatice, jsou využívány také v teorii grup nebo v diskrétní matematice.

Druhá kapitola naváže vzorcem pro počet parkovacích funkcí. Tento vzorec odvodíme a díky H. O. Pollakovi [15] také elegantně dokážeme. Podíváme se i na kombinatorický význam sčítanců ve vzorci a na související pravděpodobnost. V další části kapitoly se zabýváme střední hodnotou počtu šťastných aut a pravděpodobnostními generujícími funkcemi.

V prvních dvou kapitolách máme konečný počet parkovacích míst, auto si vybírá jedno z nich, jde o *diskrétní* problém. V kapitole 3 máme ulici konečné délky, ale parkovací místa nejsou vyznačena, a proto je možností nekonečně mnoho. Jde tedy o *spojitý* problém. Parkovací problém, kterému se poprvé věnoval Alfréd Rényi [10], tedy spočívá v postupném náhodném zaplnění intervalu. Hledáme střední hodnotu počtu aut, která zvládnou zaparkovat. Najdeme ji jako numerické řešení diferenciální rovnice se zpožděním.

Práce je primárně určena čtenářům na úrovni bakalářského studia, zčásti je ale srozumitelná i pro nadané středoškolské studenty.

Vycházeli jsme z výsledků zahraniční literatury. Důležitá je především postupná výstavba definic, důkazů i objevování nových vztahů. Některé důkazy v části 2.2 jsou nové. Během čtení si klademe otázky a čtenář na ně nachází odpovědi pomocí příkladů, úvah a programů, které si může i samostatně vyzkoušet. Jedním z hlavních cílů práce je tedy čtenáře nejen obeznámit s parkovacími problémy, ale také představit různé metody řešení jasně a srozumitelně. Návodné programy a grafy jsou vytvořené v prostředí Wolfram Mathematica, přičemž všechny programy lze nalézt v elektronické příloze práce. Obrázky vznikly za pomoci grafického editoru Krita. Práce je navíc sepsaná formou příběhu, abychom myšlenkový postup mohli doplňovat motivačními otázkami. Jakákoliv podobnost se skutečnými postavami či událostmi je ovšem čistě náhodná.

1. 2984

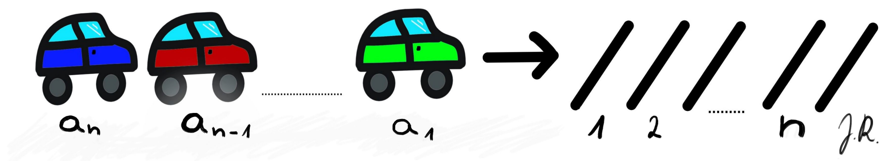
Všechno, co si někdo dokáže představit, může někdo uskutečnit.
Jules Verne

1.1 Město v oblacích

Jak už to tak bývá, civilizace se nepřetržitě vyvíjí. Nárůst populace donutil člověka přemýšlet o rozšíření svého teritoria. Obživu lidí si vzali na starost roboti a důmyslná filtrace vody umožnila využití rozsáhlých ploch oceánu. Od rodinných domků se přešlo ke kompaktním buňkám, zbořily se monstrózní mrakodrapy a zúžily se extravagantní ulice. Postupně vznikala obydlí nejdříve na stromech, poté v oceánu a následně i v oblacích. První vznášející se město Nuage Neuf, přezdívané též jako projekt NN a navržené Buckminsterem Fullerem, se odpoutalo od zemského povrchu roku 2984. Pro uskutečnění tohoto projektu bylo zapotřebí mnoho expertů. Jedním z nich byl matematik Jacob Rossin, který dostal za úkol vytvořit model efektivní infrastruktury pro automobily. Pojďme se podívat na některé zápisky z jeho pracovního deníku.

1.2 Systém jednosměrných ulic

Jednosměrné ulice zabírají méně místa a zároveň jsou více bezpečné. Dostáváme se ovšem k zajímavému problému s parkováním. Představme si, že za sebou v řadě stojí n aut $a_1, a_2, a_3, \ldots, a_n$. Auta postupně vjíždí do jednosměrné ulice. Za prvním autem a_1, vjede do ulice auto a_2, poté třetí a_3 atd.... V jednosměrné ulici se nachází přesně n parkovacích míst očíslovaných 1,2,3,$\ldots, n$.



Obrázek 1.1: Auta vjíždějící do jednosměrné ulice

Každý řidič má své oblíbené parkovací místo. Někdo chce například zaparkovat blíže u vstupu do obchodu, někdo naopak okolo sebe potřebuje více prostoru, a proto zaparkuje dál. Každý řidič tedy nejdříve směřuje ke svému oblíbenému místu a pokud je preferované místo volné, tak řidič ihned zaparkuje. Pokud je místo již obsazené, auta pokračují jednosměrnou ulicí dál, dokud nenarazí na první volné místo, kde následně zaparkují. Auta zůstávají na svém místě do té doby, než zaparkují všechna ostatní vozidla. Přičemž všechna auta parkují podle stejné strategie. *Parkovací funkce* je taková posloupnost preferencí jednotlivých vozidel, která umožní všem autům zaparkovat. Pokud nějaké auto nenašlo volné místo, řekneme, že parkovací proces byl neúspěšný a že se nejedná o *parkovací funkci* [2].

Jak zjistíme, zda mohou všechna auta zaparkovat?

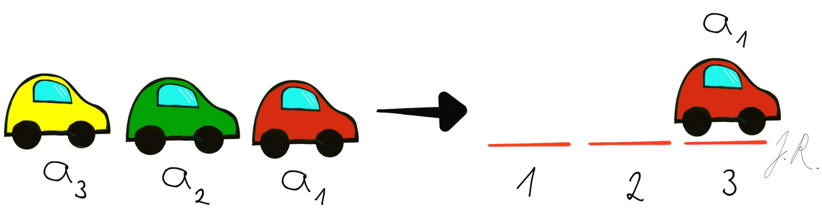
Podívejme se napřed na situace, které mohou nastat a které si Jacob Rossin načrtl do svého deníku.

1.3 Příklady parkovacích funkcí

Známe:

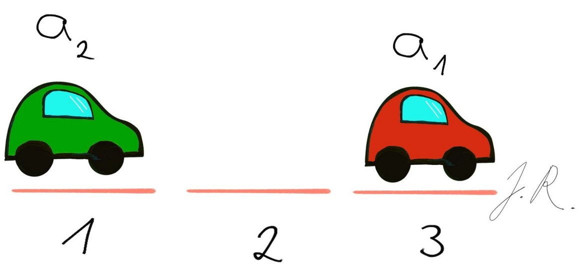
- počet aut přijíždějících do jednosměrné ulice,
- počet parkovacích míst, který odpovídá počtu vozidel,
- parkovací preference řidičů.

Řekněme, že máme tři auta a_1, a_2, a_3, tedy tři parkovací místa očíslovaná 1,2,3. První řidič chce zaparkovat na třetím místě, druhý i třetí řidič plánují zaparkovat na prvním místě. Pokud bychom autům přiřadili jejich preference podle toho, jak auta stojí v řadě za sebou před vjezdem do jednosměrné ulice, dostali bychom posloupnost (3,1,1). Jedná se o parkovací funkci? První řidič vjíždí na prázdné parkoviště, má proto vždy privilegovanou možnost volby, vybere si třetí místo.



Obrázek 1.2: První řidič určitě zaparkuje na oblíbeném místě, má vždy možnost volby.

Druhý řidič obsadí ihned místo 1.

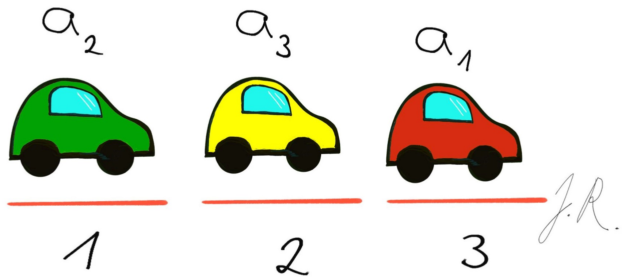


Obrázek 1.3: Druhý řidič zaparkuje na prvním místě.

Třetí řidič by také rád parkoval na začátku ulice, ale první místo je již obsazené autem a_2. Musí proto pokračovat jednosměrnou ulicí dále, dokud nenarazí na první volné místo, což je v tomto případě parkovací místo s číslem 2.

Všechna auta úspěšně zaparkovala, tedy jde skutečně o *parkovací funkci.*

Co kdyby se třetí řidič rozhodl místo prvního místa zaparkovat až na posledním místě? Ačkoliv by se posloupnost preferencí změnila jen nepatrně na (3,1,3),



Obrázek 1.4: Všechna auta zaparkují.

o parkovací funci by nešlo. Proč? Zaparkuje-li první auto a_1 na třetím místě, následuje ho druhé auto a_2, které obsadí místo první. Třetí auto zamíří rovnou k oblíbenému místu 3, které je bohužel už zabrané prvním šťastlivcem. Řidič v autě a_3 se nemůže jednosměrnou ulicí vrátit na první volné místo. Parkoviště nuceně opustí a parkovací proces tím pádem selže.

Ačkoliv si Jacob Rossin vymýšlením příkladů parkovacích funkcí rád krátil dlouhé chvíle na služebních cestách, velmi brzy si uvědomil, že pouhé kreslení obrázků a postupné zaplnění parkoviště není z praktického hlediska časově udržitelné. Co kdyby aut přijelo sto nebo tisíc? Můžeme si vyzkoušet, jak dlouho by trvalo zhodnotit parkovací možnosti pouze 7 aut, jestliže jejich preference jsou (4,4,2,5,6,1,1).

Jacob Rossin si z těchto důvodů položil otázku, jakým způsobem lze snadno a rychle určit, zda libovolný počet aut zaparkuje?

Při příjezdu aut máme dány jejich preference $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$. Je jasné, že pokud se oblíbená místa všech aut navzájem liší, tedy když posloupnost preferencí je permutací množiny $\{1, \dots, n\}$, pak všechna auta úspěšně zaparkují. Takový případ ovšem nemusí vždy nastat. Začněme proto systematickým přerovnáním preferencí $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$ tak, aby tvořily neklesající posloupnost $\beta = (b_1, b_2, b_3, \dots, b_n)$. Potřebujeme, aby všechna auta zaparkovala. Co kdybychom předpokládali, že všechna auta zaparkovat nemohou? Pak by existovalo aspoň jedno auto, které přijelo ke svému oblíbenému místu a našlo by ho už obsazené. Při pokračování jednosměrnou ulicí by další volné místo nenašlo.

Předpokládejme, že existuje i takové, že $b_i > i$. Pak také $b_{i+1}, \dots, b_n > i$. Celkem máme $n - i + 1$ aut, která chtějí zaparkovat na pozici s číslem vyšším než i. Takových pozic je ovšem pouze $n - i$, takže všechna auta určitě zaparkovat nemohou. Kdyby naopak preference řidičů splňovala podmínku $b_i \leq i$, mohla by všechna auta zaparkovat? Předpokládejme, že některá auta by za této podmínky nezaparkovala. Z neobsazených parkovacích míst následně vybereme to s nejvyšším číslem. Nechť má toto místo přiřazené číslo i. Potom aspoň $n - i + 1$ aut preferovalo místa s čísly většími než i. Jsou to ta auta, která zaparkovala na pozicích $i + 1, \dots, n$ a dále auta, která úspěšně nezaparkovala a musela z ulice odjet. To znamená, že platí $b_i, \dots, b_n > i$, což je spor s předpokladem, že $b_i \leq i$. Pojem parkovací funkce tedy můžeme formálně definovat následujícím způsobem [14].

Definice 1.1. *Máme-li posloupnost preferencí* $\pi = (\pi_1, \pi_2, \pi_3, \dots, \pi_n)$, *kterou* *uspořádáme do neklesající posloupnosti $\beta = (b_1, b_2, b_3, \ldots, b_n)$, pak π se nazývá parkovací funkce tehdy a jen tehdy, je-li $b_i \leq i$ pro každé $i \in \{1, \ldots, n\}$.*

Poznámka 1.2. *Z definice vyplývá, že libovolná permutace parkovací funkce je opět parkovací funkce.*

Nové informace se tají jen stěží. Vedení projektu NN některé poznatky Jacoba Rossina zpřístupnilo i pro veřejnost. Navíc začalo aktivně hledat programátory, kteří by postup parkování zautomatizovali. Do výběrového řízení se přihlásila Ada Byronová. Její první výzvou bylo navrhnout program, který na základě preferencí jednotlivých řidičů vyhodnotí, zda všechna auta budou moci zaparkovat neboli jinými slovy, zda se jedná o parkovací funkci.

```
(*Program 1: rozpoznávající parkovací funkce*)
(*Ada Byronová pracovala v prostředí Wolfram Mathematica*)

parkovani[q_] := Apply[And,Thread[Sort[q]<=Range[Length[q]]]]

parkovani[{1,2,2,5,5}]
False
```

Vedení projektu NN software Ady Byronové postupně testovalo. Mezi jejich cvičné příklady patřila třeba posloupnost $(1, 2, 4, 2, 2)$. Jakou hodnotu by měl program správně vrátit?

Ada chtěla rovněž nalézt pro určitý počet vozidel všechny možné parkovací funkce délky n[1]. Tedy hledala ty posloupnosti preferencí, které zaplní všechna místa v ulici. Vytvořila program, který generuje všechny uspořádané n-tice čísel z $\{1, \ldots, n\}$ a pomocí programu 1 z nich vybírá parkovací funkce.

```
(*Program 2: Pro zadané n vrátí seznam všech parkovacích funkcí.*)

seznamPF[x_] := Select[Tuples[Table[Range[x],x]],parkovani]

seznamPF[2]
{{1, 1}, {1, 2}, {2, 1}}
```

Jaké všechny parkovací funkce bychom získali, kdyby do ulice vjížděla právě tři auta?

Máme-li již program, který zjistí, za jakých podmínek všechna auta mohou zaparkovat, bylo by navíc výhodné předpovědět, kde přesně skutečně zaparkují. Vstupem následujícího programu bude parkovací funkce. Na výstupu získáme opět seznam, jehož i-tý prvek odpovídá autu, které zaparkuje na pozici i.

[1] Místo toho abychom psali, že n je počet aut a n je počet parkovacích míst, budeme stručně hovořit o parkovací funkci délky n.

```
(*Program 3:
Pro každé parkovací místo najde auto, které zde zaparkuje.
Výstupní seznam lze vnímat jako parkoviště a hodnoty
v seznamu jako čísla zaparkovaných aut.*)
kdeParkuji[x_] := Module[{list2=Table[0,Length[x]]},
   Do[a = x[[i]]; While[list2[[a]] != 0, a++];
    list2[[a]] = i, {i, 1, Length[x]} ]; list2];

kdeParkuji[{1, 3, 3, 2, 2, 5}]
{1, 4, 2, 3, 5, 6}
```

Kde by dle programu zaparkovala auta s preferencemi $(6, 4, 1, 3, 3, 1, 6, 7, 2)$?

Ada Byronová úspěšně splnila úkoly spjaté s automatizací postupu parkování a vedení požádalo o její spolupráci na projektu. Ačkoliv se Adina práce zdála být hotová, Jacob Rossin dostal ještě jeden nápad na program, jehož výsledky by mu v následujících týdnech ulehčily práci. Zadání od Rossina pro Adu Byronovou znělo následovně: „Milá kolegyně, potřeboval bych prosím naprogramovat funkci, která pro zadanou posloupnost preferencí řidičů najde počet šťastných autíček, tj. těch, která zaparkují na místě shodující se s jejich preferencí.“ Ada zpočátku neviděla pragmatický přínos této myšlenky, i tak prosbě ráda vyhověla.

Ada nejprve navrhla funkci, která pro každé i zjistí, kde zaparkuje i-té auto. Vstupem je parkovací funkce. Jestliže využijeme předchozí program, jde jen o nalezení inverzní permutace, proto na výstupu obdržíme znovu seznam.

```
(*Program 4:
Pro každé auto najde místo, kde zaparkuje.
Výstupní seznam lze vnímat jako čísla aut,
jehož hodnoty jsou pozice, na kterých auta zaparkují.*)


hledameIAuto[y_] := InversePermutation[kdeParkuji[y]]

hledameIAuto[{1, 3, 3, 2, 2, 5}]
{1, 3, 4, 2, 5, 6}
```

Dopátrat se nyní kolik aut zaparkovalo na svých preferovaných místech už je celkem snadné. Stačí srovnat seznam preferencí a seznam, který dostaneme za pomoci funkce z programu 4.

```
(*Program 5: hledání počtu šťastných aut*)

pocetVyvolenych[z_]:=
    Count[MapThread[Equal,{hledameIAuto[z],z}],True]

pocetVyvolenych[{1, 3, 3, 2, 2, 5}]
3
```

Kolik najdeme šťastných autíček, jestliže jejich preference jsou $(6, 1, 5, 2, 2, 1, 5)$?

Jacob Rossin se už nějakou dobu zamýšlel nad průměrnými počty šťastných aut, proto měl z programu velkou radost. Společně s Adou naprogramovali funkci, která pro libovolné číslo n vrátí průměrný počet šťastných aut, přičemž průměr se vezme přes všechny parkovací funkce délky n.

Jestliže vygenerujeme všechny parkovací funkce délky n využitím programu 2, na každou z nich zavoláme funkci z programu 5 a pak vypočítáme průměr, dostaneme průměrný počet šťastných aut.

```
(*Program 6, 1. způsob: průměrný počet šťastných aut*)

prumerSA[n_] := Mean[Map[pocetVyvolenych, seznamPF[n]]]

prumerSA[2]
5/3
```

Naneštěstí tento způsob výpočtu průměrného počtu šťastných aut je použitelný jen pro malé hodnoty n, protože počet parkovacích funkcí velmi rychle roste. Jacob Rossin s tímto výsledkem nebyl spokojený a program se mu zdál pomalý. Plánoval proto najít ještě jiný způsob, jak průměrný počet aut získat. Ovšem aby s Adou problém úspěšně vyřešili, potřeboval se do tématu parkovacích funkcí ponořit hlouběji. Přesný vzorec pro počet parkovacích funkcí a efektivnější způsob, jak počítat průměrné počty šťastných aut, si ukážeme v další kapitole pracovního deníku Jacoba Rossina.

2. Pollakův odkaz

Někdy se ovšem může stát, že některý z řidičů nebude předem znát své oblíbené místo a rozhodne se, až když vjede do ulice. V takové situaci by nebylo snadné předpovědět, kde přesně auto zaparkuje. Pokud jsou preference řidičů zvoleny náhodně, jaká je pravděpodobnost, že všichni zaparkují?

2.1 O krok zpět

Kdo ovládá minulost, ovládá budoucnost.
George Orwell

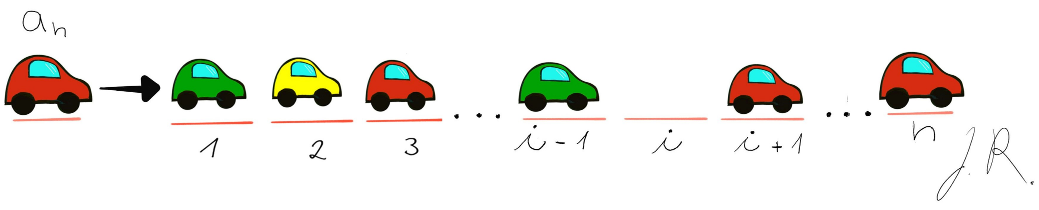
Hledaná pravděpodobnost je dána zlomkem, v jehož jmenovateli je celkový počet posloupností preferencí, čili n^n. Zbývá určit počet příznivých jevů, tj. počet parkovacích funkcí délky n. Jacob Rossin se nejdříve zamyslel nad tím, jak najít počet parkovacích funkcí délky n, umíme-li najít počet parkovacích funkcí délky $n-1$. Rozhodl se proto odvodit rekurentní vztah.

Nechť p_n je počet parkovacích funkcí délky n. Počátečními hodnotami jsou $p_0 = 1$ a $p_1 = 1$. Zbývá určit pravidlo, kterým najdeme počet parkovacích funkcí pro další případy. Takové pravidlo zkonstruujeme [12].

Věta 2.1. *Pro počet parkovacích funkcí délky n platí*

$$p_n = \sum_{i=1}^{n} i \cdot \binom{n-1}{i-1} \cdot p_{i-1} \cdot p_{n-i}.$$

Důkaz. Připomeňme, že n aut vjíždí do ulice s n parkovacími místy. Předpokládejme, že máme parkovací funkci, tudíž všechna auta mohou zaparkovat. Při parkování posledního auta a_n zbývá přesně jedno volné místo. Řekněme, že jde o místo s číslem i. Jinými slovy i představuje první a zároveň jediné volné místo pro přijíždějící auto a_n. Místo i může být na parkovišti zvoleno libovolně. Počítáme tedy všechny posloupnosti preferencí, pro které platí, že poslední auto a_n zaparkuje na i-tém místě.



Obrázek 2.1: Možné očíslování parkovacích míst pro i-té místo.

Množinu aut $a_1, \dots, a_{n-1}$ disjunktně rozdělíme na podmnožiny S a S'. Podmnožina S obsahuje auta parkující na místech $1, \dots, i-1$ a podmnožina S' zahrnuje auta parkující na místech $i+1, \dots, n$. Počet parkovacích funkcí, které odpovídají preferencím řidičů z podmnožiny S, je p_{i-1}. Zároveň víme, že žádné auto z podmnožiny S' nemůže mít preferenci $1, \dots, i-1$, protože pak by místo i bylo již obsazené. Počet parkovacích funkcí na místech $i+1,\dots,n$ je stejný, jako kdybychom parkovací funkce počítali na místech $1,\dots,n-i$, protože počet parkovacích míst po i-tém místě je právě $n-i$. Tudíž počet parkovacích funkcí, které odpovídají preferencím řidičů z podmnožiny S', je p_{n-i}.

Dále existuje $\binom{n-1}{i-1}$ způsobů, jak vybrat $i-1$ aut z možných $n-1$ aut, abychom vytvořili podmnožinu S. Všimněme si, že a_n má i možných preferencí, přičemž preference a_n mohou nabývat jakékoli hodnoty z množiny $\{1,\dots,i\}$.

Za podmínky, že poslední auto a_n zaparkuje na místě i, dostaneme vynásobením dílčích výsledků našeho postupu výraz

$$p_{i-1}\cdot p_{n-i}\cdot i\cdot\binom{n-1}{i-1},$$

který udává počet parkovacích funkcí délky n.

Sčítáme-li přes všechna $i\in\{1,\dots,n\}$, pak počet parkovacích funkcí délky n je:

$$p_n=\sum_{i=1}^{n} i\cdot\binom{n-1}{i-1}\cdot p_{i-1}\cdot p_{n-i}.$$

□

Jacob Rossin našel způsob, jak zjistit počty příznivých případů. Bohužel pracovat opakovaně s předcházejícími členy nemusí být vždy nejefektivnější metoda. Bylo by možné z nalezeného rekurentního vztahu uhodnout obecný vzorec, který by byl vhodnější pro další výpočty? Jacob Rossin s touto myšlenkou zašel za Adou Byronovou, aby mu pomohla zjistit několik členů rekurentní posloupnosti.

```
(*Program 7: výpočet členů rekurentní posloupnosti*)
p[0] = 1;
p[1] = 1;
p[n_] := Sum[Binomial[n-1,i](i+1)p[i]p[n-1-i], {i,0,n-1}];
Table[p[i], {i,0,10}]
```

n	0	1	2	3	4	5	6	7	8	9
p_n	1	1	3	16	125	1 296	16 807	262 144	4 782 969	10^8

Zkusíme nyní uvedená čísla rozložit na prvočinitele.

p_n	1	1	3	16	125	1 296	16 807	262 144	4 782 969	10^8
rozklad	1^1	1^1	3^1	2^4	5^3	$2^4\cdot 3^4$	7^5	2^{18}	3^{14}	$2^8\cdot 5^8$

Obraťme nyní pozornost k opakujícímu se vztahu základu a exponentu v jednotlivých případech rozkladu. Podíváme-li se například na číslo 5^3, lze ho zapsat jako $(4+1)^{4-1}$, číslo 7^5 zase jako $(6+1)^{6-1}$. Jacob Rossin se pokusil z tabulkových hodnot uhodnout obecný vzorec a zjistil, že bychom příznivé jevy mohli počítat pomocí vzorce $(n+1)^{n-1}$.

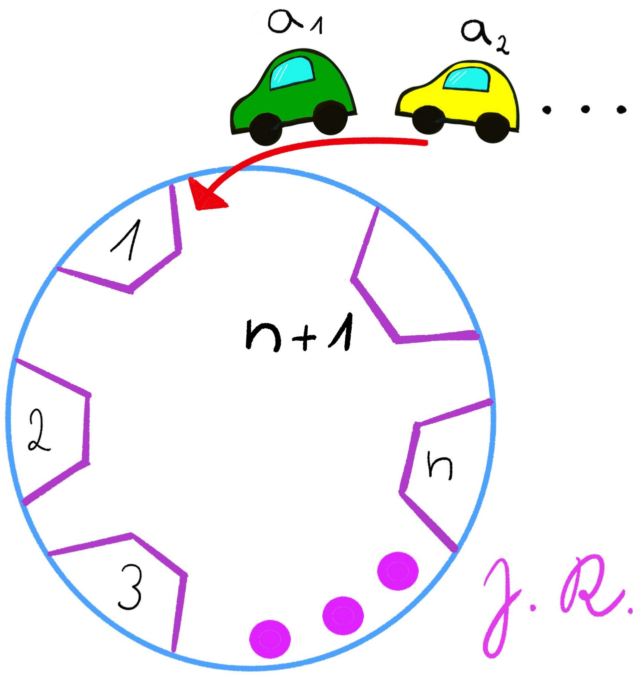
I když Jacob Rossin shledal, že výsledky obecného vzorce pro příznivé jevy souhlasí s tabulkovými hodnotami z rekurentního vztahu, příliš se neradoval. Zatím šlo o pouhé hádání, a to pro něj nebyla ta pravá matematika. Matematika nás učí pochybovat a Jacob Rossin o svém výsledku vskutku pochyboval. Potřeboval důkaz. Vědom si toho, že pokračuje ve šlépějích svých předchůdců, napadlo ho prozkoumat jejich práce. Bohužel vědci, kteří přímo spolupracovali s Buckminsterem Fullerem na prvních plánech měst v oblacích, publikovali před více než 1000 lety. Navíc během kybernetické války v roce 2371 byl virtuální svět téměř zničen, protože se hackerská skupina Trespassers snažila zneužít přísně tajné vládní informace k ovládnutí světa. Jacobu Rossinovi tedy nezbývalo nic jiného, než zapátrat po originálních dokumentech v archivech projektu NN a pod světlem lampičky projít obsah mnoha zaprášených krabic. Jelikož věděl, k jakým výsledkům by rád došel, nakonec našel přesně to, co hledal.

Věta 2.2. ***Konheim a Weiss, 1966.***
Počet parkovacích funkcí délky n je

$$p_n = (n+1)^{n-1}.$$

Elegantní důkaz této věty navrhl Henry O. Pollak o 8 let později.

Důkaz. **Pollak, 1974**, [8], [15], [11], [6]. Přidejme k n parkovacím místům ještě místo $n+1$ a představme si, že toto místo může být také preferencí některých řidičů. Následně uspořádáme tato místa do kruhu. Pokud auto přijede k místu $n+1$ a najde je obsazené, bude pokračovat opět od začátku. Díky přidanému místu $n+1$ máme více parkovacích míst než aut, tudíž všechna auta mohou zaparkovat a zároveň vždy zbyde jedno volné místo.



Obrázek 2.2: Uspořádání parkovacích míst do kruhu.

Z definice víme, že posloupnost preferencí π, jejíž prvky jsou čísla $\{1,\dots,n\}$, představuje parkovací funkci právě tehdy, když při parkování do kruhu zůstane volné místo $n+1$. Kdyby na místě $n+1$ v kruhovém uspořádání zaparkovalo nějaké auto, znamenalo by to, že v jednosměrné lineárně uspořádané ulici s n místy by toto auto místo nenašlo, nejednalo by se tudíž o parkovací funkci.

Definujme zobrazení F zobrazující libovolnou posloupnost preferencí π na novou posloupnost preferencí $F(\pi)$, která vznikne tak, že každému π_j z posloupnosti preferencí $\pi = (\pi_1, \pi_2, \pi_3, \ldots, \pi_n)$ pro každé $j \in \{1, \ldots, n\}$ přiřadíme místo sousedící s π_j při pohybu proti směru hodinových ručiček. Pokud má řidič v původní posloupnosti π preferenci π_j, která odpovídá místu i, pak v nové posloupnosti $F(\pi)$ bude π_j odpovídat místu $i+1$. Výjimkou je místo $n+1$, kdy F zobrazí $n+1$ na místo 1.

Vezmeme-li novou posloupnost preferencí $F(\pi)$, pak se během parkovacího procesu každé auto posune o jednu pozici proti směru hodinových ručiček. Proto právě jedna z posloupností[1] $\pi, F(\pi), F^2(\pi), \ldots, F^n(\pi)$ je parkovací funkcí. Je to jako kdybychom kruh vždy otočili o jedno místo, dokud místo $n+1$ nezůstane volné. Jelikož uvažujeme $n+1$ posloupností preferencí, z nichž právě jedna je parkovací funkce, pak

$$p_n = \frac{(n+1)^n}{n+1} = (n+1)^{n-1}.$$

□

Jacob Rossin našel obecný vzorec pro počet parkovacích funkcí délky n i jeho důkaz. Jestliže na obecný vzorec pro počet parkovacích funkcí délky n aplikujeme binomickou větu

$$(n+1)^{n-1} = \sum_{i=0}^{n-1} \binom{n-1}{i} n^i \tag{2.1}$$

a následně upravíme sčítance v $(n+1)^{n-1}$, získáme:

$$\begin{aligned} \binom{n-1}{i} n^i &= \frac{(n-1)(n-2)\ldots(n-i)}{i!} n^i = \\ &= \frac{n(n-1)(n-2)\ldots(n-i+1)}{i!}(n-i)n^{i-1} = \\ &= \binom{n}{i} n^{i-1}(n-i). \end{aligned}$$

Pokud jsme sčítance takto upravili, můžeme si rozmyslet jejich kombinatorický význam [9].

Věta 2.3. *Počet parkovacích funkcí délky n, kde i řidičů preferuje místa odlišná od 1, je*

$$\binom{n}{i} n^{i-1}(n-i).$$

[1] $F^2 = F \circ F$, F^n značí v našem případě skládání zobrazení.

Důkaz. Z definice parkovací funkce plyne, že některý z řidičů musí mít oblíbené parkovací místo s číslem 1. Označme i počet aut, jejichž preference jsou různé od místa s číslem 1. Dále víme, že i nabývá všech hodnot mezi $0, \ldots, n-1$ a $n-i$ je počet preferencí prvního parkovacího místa. Počet způsobů, jak vybrat čísla aut, jejichž řidiči nepreferovali první místo, je $\binom{n}{i}$. Nyní dokážeme, že pokud i řidičů preferuje místa odlišná od 1, pak počet parkovacích funkcí délky n je:

$$\binom{n}{i} n^{i-1}(n-i).$$

Pozorujme tedy i aut, které mají preference různé od prvního místa, tj. $\{2, \ldots, n\}$. Využijeme Pollakova důkazu a budou nás zajímat ty posloupnosti preferencí z množiny $\{2, \ldots, n+1\}$, kde místo $n+1$ zůstane volné. Z počtu aut a preferencí lze odvodit, že celkový počet funkcí bude n^i. Pomocí výše zavedeného zobrazení F preference *otáčíme* a počítáme, kolikrát bude pozice $n+1$ neobsazená. Máme i aut, pozice $n+1$ bude obsazená i-krát, tedy $n-i$-krát neobsazená. Otočit kruh můžeme n-krát. Proto poměr počtu parkovacích funkcí, kdy je pozice $n-i$ neobsazená, tedy kdy vybraná pozice není preferovaným prvním místem, vzhledem k počtu všech funkcí je:

$$\frac{n-i}{n} n^i = n^{i-1}(n-i).$$

□

Ukázali jsme kombinatorický význam sčítanců ze vztahu (2.1).

Z věty 2.2 vyplývá odpověď na naši úvodní otázku. Pokud jsou preference řidičů zvoleny náhodně, hledaná pravděpodobnost, že všechna auta zaparkují, je:

$$P = \frac{(n+1)^{n-1}}{n^n}.$$

Následující tabulka ukazuje hodnoty pravděpodobností.

n	1	2	3	4	5	6	7	8
P	1	0,75	0,592593	0,488281	0,41472	0,360232	0,318312	0,285087

Limita pravděpodobností pro n jdoucí do nekonečna je:

$$\lim_{n\to\infty} \frac{(n+1)^{n-1}}{n^n} = \lim_{n\to\infty} \frac{(n+1)^n}{n^n(n+1)} = \lim_{n\to\infty} \left(\frac{n+1}{n}\right)^n \frac{1}{(n+1)} =$$

$$= \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n \frac{1}{(n+1)} = 0.$$

Pozorný čtenář si jistě pomyslí, že odpovědí na úvodní otázku kapitola ještě nekončí a že nyní už by Jacob Rossin mohl mít dostatek znalostí k nalezení přesného vzorce pro výpočet průměrného počtu šťastných aut, nad kterým dumal společně s Adou Byronovou v kapitole 1. Je-li čtenář navíc zvídavý, jistě ocení, že má skutečně pravdu.

2.2 Štěstí přeje autíčkům

Štěstí patří těm, kteří patří sami sobě.
Aristoteles

Jacob Rossin nebyl spokojený s programem pro výpočet průměrného počtu šťastných aut a těšil se, až se pustí do hledání nového řešení, které Adě Byronové navrhne. Ovšem než se stihl pustit do práce, dostal od vedení projektu NN spěšnou a nemilou zprávu o naléhavé schůzi. Konstrukce vznášejícího se města vypadala tak, že se struktury trojúhelníkových součástí uspořádaly do tvaru koule. Architekti ovšem neodhadli správné materiály. Bylo tedy třeba naplánovat, kdo vytvoří počítačové modely jemných kovových materiálů především na bázi hliníku a otestuje, že budou vhodné. Ačkoliv Jacob Rossin nesměl na schůzi chybět jako konzultant infrastruktury, neměl podobná zasedání příliš v oblibě. Sedl si proto v konferenční místnosti až úplně dozadu vedle svého kamaráda Viktora Glücka z výzkumu pravděpodobnosti. Výhodou nudných zasedání a zajímavých spolusedících je to, že člověka často něco kloudného napadne. A tak když Jacob Rossin diskutoval s Viktorem své poznatky, napadlo ho zformulovat větu o střední hodnotě počtu šťastných aut.

Věta 2.4. *Střední hodnota počtu šťastných aut pro náhodnou parkovací funkci délky n je*

$$\frac{n}{2}\left(1+\frac{2}{n+1}\right).$$

Důkaz. Pro autíčka uspořádaná v řadě není snadné vypočítat pravděpodobnost, že i-té auto bude šťastné. Co kdybychom proto zkusili parkovací místa uspořádat jiným způsobem? Věta 2.2 a Pollakův důkaz přímo vybízí k uspořádání míst do kruhu. Při takovém uspořádání můžeme místo počítání průměrného počtu šťastných aut pro náhodnou parkovací funkci délky n počítat průměrný počet šťastných aut pro náhodnou posloupnost preferencí $(\pi_1, \ldots, \pi_n)$ s hodnotami z $\{1, \ldots, n+1\}$ pro n aut a $n+1$ míst. Tento záměnný trik lze odůvodnit právě již zmíněným zobrazením F z Pollakova důkazu. Pro každou posloupnost preferencí π platí, že všechny posloupnosti π, $F(\pi)$, $F^2(\pi)$, ..., $F^n(\pi)$, dávají stejný počet šťastných aut. Přitom je mezi nimi právě jedna parkovací funkce.

Pro danou posloupnost preferencí a každé $i \in \{1, \ldots, n\}$ označme $X_i = 1$, pokud i-té auto je šťastné a $X_i = 0$, je-li nešťastné. Jestliže $X_1, \ldots, X_n$ jsou náhodné veličiny, pak průměrný počet šťastných aut zjistíme jako střední hodnotu součtu $X_1 + \cdots + X_n$, což je totéž jako součet středních hodnot.

Mějme tedy náhodnou posloupnost preferencí pro n aut a $n+1$ míst a počítejme podle definice střední hodnotu veličiny X_i. V okamžiku, kdy parkuje i-té auto, je $i-1$ míst z $n+1$ obsazených. Tedy pravděpodobnost, že i-té auto je nešťastné, je

$$\frac{i-1}{n+1}.$$

Chceme-li pravděpodobnost, že i-té auto bude šťastné, musíme vzít doplňkovou pravděpodobnost. Touto úvahou získáváme střední hodnotu.

$$E(X_i) = 0 \cdot P(X_i = 0) + 1 \cdot P(X_i = 1) = P(X_i = 1) = 1 - \frac{i-1}{n+1} \tag{2.2}$$

Dále z předchozí rovnosti vyplývá

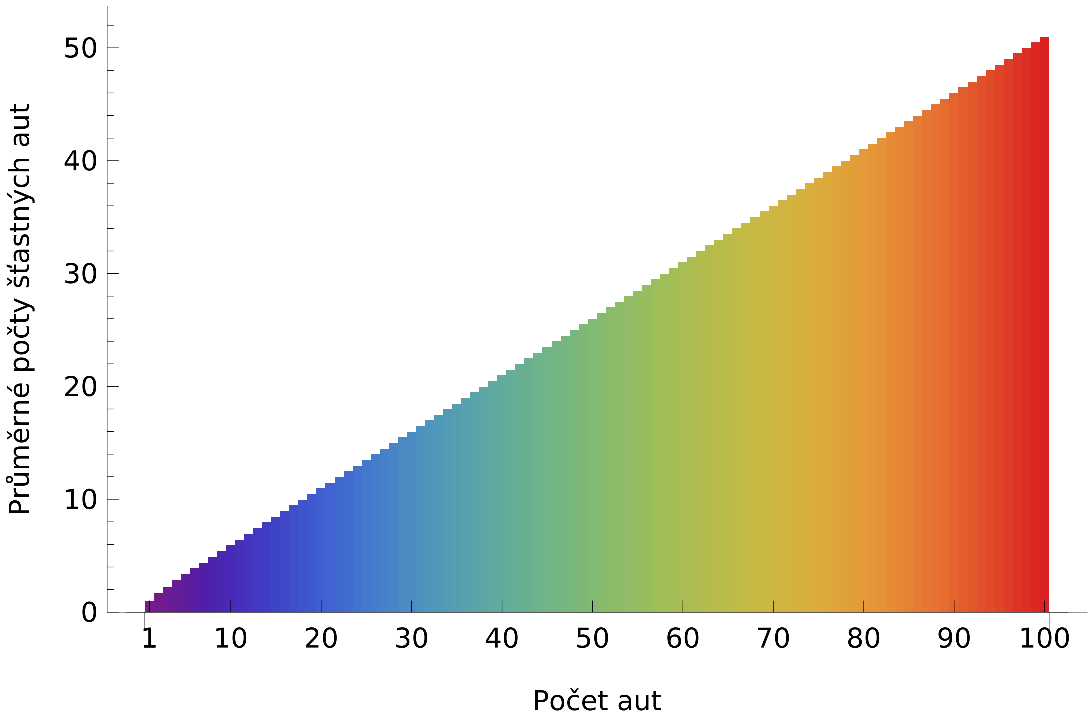
$$E(X_1 + ... + X_n) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} \left(1 - \frac{i-1}{n+1}\right) =$$

$$= n - \frac{(n-1)n}{2(n+1)} = \frac{n}{2}\left(1 + \frac{2}{n+1}\right).$$

□

Lze postřehnout, že v průměru bude více než polovina autíček šťastných. Podívejme se, jak vycházejí průměrné počty šťasných autíček pro nízké hodnoty n. Vzorec pak ještě ilustrujeme pomocí grafu.

n	1	2	3	4	5	6	7	8	9	10
průměr	1	1,667	2,25	2,8	3,333	3,857	4,375	4,889	5,4	5,909



Obrázek 2.3: Výpočet průměrného počtu šťastných aut do $n = 100$.

Jacob Rossin ke svému překvapení shledal schůzi velmi užitečnou, protože stihl odvést spoustu práce. Jeho kolegu Viktora Glücka diskuze o průměrném počtu šťastných aut také bavila. Nicméně zástupci z řad vedení neměli pro nadšení upovídané dvojice zrovna pochopení. Schůze tedy pro Viktora i Jacoba skončila o něco dříve, a protože se už pomalu stmívalo, navrhl Viktor Jacobovi, že ho sveze domů. Začaly se rozsvěcet lampy a jak to tak bývá, většina lidu se hrnula z práce. Doprava postupně houstla a auto našich matematiků zpomalovalo, až se na mostě v moři okolních autíček úplně zastavilo. Jacob Rossin se díval na zástupy aut a pomyslel si, že i v městě v oblacích a ve spleti jednosměrných ulic budou časové intervaly, kdy bude dopravní situace opravdu napjatá. Začali s Viktorem probírat scénáře, jaká je pravděpodobnost, že řidiči v městě v oblacích zaparkují na místech, které preferují. Co kdyby chtěli, aby právě dvě auta zaparkovala na svých oblíbených místech? Nebo tři? Otázku dále zobecnili. Jaká je pravděpodobnost $p_l(n)$, že pro náhodně zvolenou parkovací funkci délky n bude právě l aut šťastných? Neboli jaký je počet parkovacích funkcí délky n s právě l šťastnými auty, označme tento počet $a_l(n)$, který obdržíme, pokud $p_l(n)$ vynásobíme počtem všech parkovacích funkcí délky n?

Abychom na dané otázky našli odpověď, vrátíme se k náhodným veličinám $X_1, \ldots, X_n$. Z Pollakova důkazu věty 2.2 vidíme, že posouvání v kruhu neovlivní, zda je auto šťastné či nikoli. Počet šťastných aut se zachovává. Pravděpodobnostní generující funkce[2] pro každou náhodnou veličinu X_i je:

$$G_i(z) = P(X_i = 0) + P(X_i = 1)z = \frac{i-1}{n+1} + \left(1 - \frac{i-1}{n+1}\right) z,$$

což vyplývá rovněž ze vztahu (2.2). Jelikož je PGF pro součet nezávislých náhodných veličin rovna součinu jejich PGF[3], je patrné, že

$$G(z) = G_1(z) \cdots G_n(z) = \prod_{i=1}^{n} \left[\frac{i-1}{n+1} + \left(1 - \frac{i-1}{n+1}\right) z\right] \tag{2.3}$$

je PGF pro náhodnou veličinu $X_1 + \cdots + X_n$ tj. pro počet šťastných aut. Tedy je-li $p_l(n)$ pravděpodobnost, že pro náhodnou parkovací funkci délky n je právě l aut z n aut šťastných, pak z definice PGF plyne

$$G(z) = \sum_{l=0}^{n} p_l(n) z^l = \prod_{i=1}^{n} \left[\frac{i-1}{n+1} + \left(1 - \frac{i-1}{n+1}\right) z\right] =$$

$$= z \prod_{i=1}^{n-1} \left[\frac{i}{n+1} + \left(1 - \frac{i}{n+1}\right) z\right].$$

Pokud rovnost vynásobíme počtem všech možných jevů neboli počtem všech parkovacích funkcí délky n, kterých je $(n+1)^{n-1}$, dostaneme:

$$\sum_{l=0}^{n} a_l(n) z^l = z \prod_{i=1}^{n-1} \left[i + (n-i+1)z\right]. \tag{2.4}$$

Na obou stranách rovnosti jsou polynomy v proměnné z. Vezmeme-li polynom na pravé straně rovnosti (2.4), $a_l(n)$ je koeficient u z^l a určuje počet parkovacích funkcí s právě l šťastnými autíčky. Jacob Rossin byl rád, že si mohl o matematice s někým mile popovídat. Ovšem kolegiální popovídání může přinést i úskalí. Když Viktor pročítal odvození rovnosti (2.4), z legrace Jacoba pošťouchl, že by mu krok zahrnující součin generujících funkcí ve vztahu (2.5) vůbec nemusel věřit. To se Jacoba samozřejmě malinko dotklo. Proto pro všechny nedůvěřivé matematické skeptiky sepsal následující dodatek.

Věta 2.5. *Nechť $X_1, \ldots, X_n$ jsou nezávislé diskrétní náhodné veličiny nabývající hodnot z $\mathbb{N}_0$. Pak PGF pro jejich součet $X_1 + \cdots + X_n$ odpovídá součinu PGF pro jednotlivé veličiny*

$$G(z) = G_1(z) \cdots G_n(z).$$

[2]Jestliže je X diskrétní náhodná veličina nabývající hodnot $0, 1, 2, \ldots$ s pravděpodobnostmi $P(X = 0), P(X = 1), P(X = 2), \ldots$, pak se funkce $G(z) = \sum_{n=0}^{\infty} P(X = n) z^n$. nazývá pravděpodobnostní generující funkcí této veličiny.

[3]Důkaz tohoto tvrzení ukážeme v dodatku vzápětí.

Důkaz. Rozepíšeme dle definice generující funkci

$$G(z) = \sum_{l=0}^{\infty} P\left(X_1 + \cdots + X_n = l\right) z^l.$$

Mocninné řady se rovnají, pokud se rovnají jejich koeficienty. Podíváme se tedy na všechny případy, kdy součet náhodných veličin bude roven l.

$$P\left(X_1 + \cdots + X_n = l\right) = \sum_{\substack{i_1,\ldots,i_n \in \mathbb{N}_0 \\ i_1+\cdots+i_n = l}} P\left(X_1 = i_1 \wedge X_2 = i_2 \wedge \cdots \wedge X_n = i_n\right).$$

Máme-li nezávislé jevy, pravděpodobnost, že nějaký z nich nastane, můžeme vypočítat jako součin těchto pravděpodobností,

$$\sum_{\substack{i_1,\ldots,i_n \in \mathbb{N}_0 \\ i_1+\cdots+i_n = l}} P\left(X_1 = i_1 \wedge \cdots \wedge X_n = i_n\right) = \sum_{\substack{i_1,\ldots,i_n \in \mathbb{N}_0 \\ i_1+\cdots+i_n = l}} P\left(X_1 = i_1\right) \cdots P\left(X_n = i_n\right). \tag{2.5}$$

Suma z pravé strany (2.5) je koeficient u z^l v následujícím součinu:

$$\prod_{j=1}^{n} \left[\sum_{i=0}^{\infty} P\left(X_j = i\right) z^i \right] = \prod_{j=1}^{n} G_j(z).$$

□

Jacob Rossin dokončil poslední řádky důkazu, ale ještě nebyl hotov. Následující dny opět zapátral v archivech projektu NN, aby si ověřil, zda mu něco ohledně šťastných autíček neuniklo. Zjistil, že nejen vztah (2.4), ale i vzorec ve větě 2.4 souvisí také s myšlenkami, které sepsali jeho předchůdci I. M. Gessel a S. Seo [7], dále pak P. Diaconis ve spolupráci s A. Hicksem [3].

Rossin si uspořádal všechny své poznámky a požádal Adu Byronovou o další programy. První program, který pokračuje v trajektorii šťastných autíček z kapitoly 1 a nepřímo na ně navazuje, nejdříve vypíše členy posloupnosti $\{a_l(n)\}_{l=0}^n$. Pro zadané n vezmeme polynom na pravé straně rovnosti (2.4) a hledáme seznam koeficientů u z^l, které tvoří naši posloupnost $\{a_l(n)\}_{l=0}^n$.

```
(*Program 6, 2. způsob: *)
(*počty parkovacích funkcí délky n s l šťastnými auty,
kde l=0, ..., n*)

koefAl[n_]:=CoefficientList[z(Product[(i+(n-i+1)z),{i,1,n-1}]),z]

koefAl[3]
{0,2,8,6}
```

Nyní už jen vypočítáme pravděpodobnost, že pro náhodně zvolenou parkovací funkci n bude právě l aut šťastných. Při programování využijeme vztah (2.4). Vstupní hodnotou funkce je znovu libovolně zadané n, výstupem je seznam pravděpodobností $\{p_l(n)\}_{l=0}^n$.

```
(*Program 8: *)
(*Pravděpodobnost, že pro náhodně zvolenou pf délky n
bude právě l aut šťastných, kde l=0, ..., n.*)

pravl[n_] := koefAl[n]/((n+1)^(n-1))

pravl[2]
{0,1/3,2/3}
```

Zvolíme-li například $n = 8$, pro rostoucí počet šťastných l aut budou pravděpodobnosti následující:

l	1	2	3	4	5	6	7	8
$p_l(n)$	0,000154	0,0034	0,0274	0,1084	0,2362	0,2972	0,2174	0,0894

Ať už libovolný řidič v městě v oblacích preferuje parkování blízko u vchodu svého domu, pod stromem či na kraji ulice, určitě ho potěší, že Fortuna je jeho oblíbenému místu relativně nakloněna. Ačkoliv paradoxně je někdy obrovské štěstí pokud to, co chceme, vždy nedostaneme.

3. Parkování jako kompromis osudu a svobodné vůle

Svoboda v konkrétním slova smyslu spočívá v možnosti volby.
Simone Weil

Plány na konstrukci jednosměrných ulic již byly v procesu. Pokud řidiči dodrží pár jednoduchých pravidel, parkování bude efektivní a ulice nezaberou mnoho místa. Jacoba Rossina bádání bavilo a vedení projektu NN bylo s jeho výsledky spokojeno. Město v oblacích ovšem stále zahrnovalo i veřejné zájmy, a protože život není spravedlivý, ve stínu veřejných zájmů se vždy skrývá i nešťastná politika. Jedna konkrétní politická figurka přivedla už několik vědců do nesnází. Jednalo se o nabubřelého politika Milo Lairda, který mnoho nových nápadů považoval za zbytečné a nákladné (na rozdíl od jeho vil či zahraničních cest). Doslechl se o plánované infrastruktuře a vydal prohlášení, že všechna pravidla či jednosměrné ulice jsou zbytečné. Lidé přece nesmí být ničím limitováni, mohou si parkovat dle libosti, místa nemusí být nijak ohraničená, parkoviště se vybudují velká, stačí prodloužit ulice a vše bude perfektně fungovat, proč by nemělo? Asi není nutno podrobně rozebírat, jak vedení projektu NN, které se snažilo každý centimetr ve městě v oblacích využít, reagovalo na návrh velkých parkovišť. Veřejnost naneštěstí hrála důležitou roli u investorů, kteří financovali projekt. Vedení pověřilo Jacoba politickým odbojem, který měl výrok Milo Lairda uvést na pravou míru. Jacob si uvědomil, že přesvědčovací proslovy nepatří mezi jeho silné stránky. Na protest proti výroku zvolil jinou strategii. Začal budovat experimentální model založený na připomínkách Mila Lairda. Protože pravdu podepřenou pevnými pilíři matematiky prázdná slova nezboří.

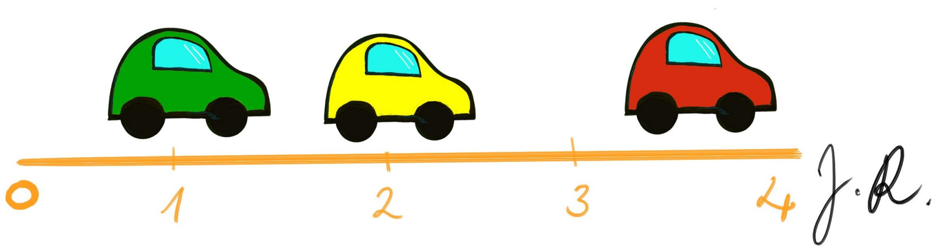
3.1 Experiment

Pro matematický model si Jacob Rossin z projevu nechvalného politika vypsal následující podmínky:

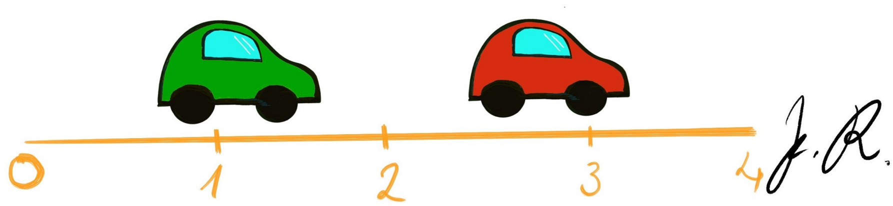
- Vybudují se dlouhé ulice.
- Místa nejsou nijak ohraničená.
- Řidiči mohou parkovat dle libosti.

Dlouhou ulici sloužící k parkování si lze také představit jako interval $[0,x]$. Předpokládejme, že auta mají jednotkovou délku. Auta přijíždějí postupně, každé má nějakou preferovanou pozici. Pozice je buď volná, pak auto zaparkuje, nebo už je obsazená, pak auto odjede. Proces pokračuje tak dlouho, dokud zbývá volné místo pro aspoň jedno auto. Celkový počet aut jednotkové délky, která zaparkují na intervalu $[0,x]$, označíme $N(x)$. Jde o náhodnou veličinu.

Vezměme si například ulici délky 4. Pokud jsou řidiči ohleduplní, auta zaparkují všechna bez problémů. Pokud se ovšem některému řidiči zlíbí zaparkovat bezohledně, zaparkují jen tři autíčka a na poslední nezbyde prostor. Když znásobíme bezohlednost u prvních dvou řidičů, může se dokonce stát, že nikdo další



Obrázek 3.1: Jeden řidič zaparkuje bezohledně.



Obrázek 3.2: Dva řidiči zaparkují bezohledně.

už se nevejde. Jacob Rossin si takové situace opět načrtnul do svého deníku. Pro střední hodnotu počtu aut, která se vejdou na parkoviště, zvolíme značení

$$M(x) = E(N(x)).$$

Rekurentní vztah pro výpočet střední hodnoty počtu aut, která parkoviště zaplní, je následující [13]:

$$M(x+1) = \frac{1}{x}\int_0^x (M(t) + M(x-t))\,\mathrm{d}t + 1. \tag{3.1}$$

Podívejme se na parkování prvního autíčka. První auto zaparkuje jedním okrajem v bodě t a druhým okrajem v bodě $t+1$. Rozdělí ulici délky x na dvě části. Jedna část je délky t a druhá část délky $x-t$. Od prvního auta se tedy v průměru vejde $M(x-t)$ aut vpravo a $M(t)$ aut zaparkuje vlevo. Protože t nabývá hodnot z intervalu $[0,x]$, zprůměrujeme výpočet přes všechna možná t:

$$M(x+1) = \frac{1}{x}\int_0^x (1 + M(t) + M(x-t))\,\mathrm{d}t. \tag{3.2}$$

Zde rovněž předpokládáme, že $M(x) = 0$ pro všechna $x \in [0,1)$. Musíme být ovšem rozvážnější při definování $M(1)$. Pokud položíme $M(1) = 1$, pak pro $t = 1$, nebo $t = x-1$ hodnota výrazu $1+M(t)+M(x-t)$ je vyšší než správný počet aut, protože pravděpodobnost, že další auto zaparkuje přesně na intervalu $[0,1]$, nebo

$[x, x+1]$ je nulová. Nicméně rovnost (3.2) stále platí, jelikož hodnota $M(x) = 1$ nijak neovlivní hodnotu integrálu. Předchozí vztah platí pro všechna $x \geq 0$. Můžeme tedy místo x psát $x-1$ a předpokládat, že $x \geq 1$, což se nám při dalším výpočtu bude hodit. Díky linearitě integrálu navíc můžeme rovnost (3.2) zapsat ve tvaru:

$$M(x) = 1 + \frac{1}{x-1}\int_0^{x-1} M(t)\,\mathrm{d}t + \frac{1}{x-1}\int_0^{x-1} M(x-1-t)\,\mathrm{d}t.$$

Použijeme substituci $u = x - 1 - t$, kde $\mathrm{d}u = -\mathrm{d}t$:

$$\begin{aligned} M(x) &= 1 + \frac{1}{x-1}\int_0^{x-1} M(t)\,\mathrm{d}t - \frac{1}{x-1}\int_{x-1}^{0} M(u)\,\mathrm{d}u, \\ M(x) &= 1 + \frac{1}{x-1}\int_0^{x-1} M(t)\,\mathrm{d}t + \frac{1}{x-1}\int_0^{x-1} M(u)\,\mathrm{d}u. \end{aligned}$$

Integrály mají stejnou hodnotu, tedy platí

$$M(x) = 1 + \frac{2}{x-1}\int_0^{x-1} M(t)\,\mathrm{d}t. \tag{3.3}$$

Pozorujeme, že

$$\begin{aligned} M(x) &= 0, \quad 0 \leq x < 1 \\ M(x) &= 1, \quad 1 \leq x < 2. \end{aligned}$$

Pro interval $2 \leq x < 3$ vyjde

$$M(x) = 1 + \frac{2}{x-1}\int_1^{x-1} 1\,\mathrm{d}t = 1 + \frac{2(x-2)}{x-1}.$$

Dále pro interval $3 \leq x < 4$ už je postup o malinko složitější, protože musíme využít předchozí hodnoty [16]:

$$\begin{aligned} M(x) &= 1 + \frac{2}{x-1}\left(1 + \int_2^{x-1}\left(1 + \frac{2(t-2)}{t-1}\right)\mathrm{d}t\right) \\ &= 1 + \frac{2}{x-1}\left(1 + [3t - 2\ln(t-1)]_{t=2}^{x-1}\right) \\ &= 1 + \frac{2}{x-1}(1 + 3(x-3) - 2\ln(x-2)) \\ &= 1 + \frac{2}{x-1}(3x - 8 - 2\ln(x-2)). \end{aligned} \tag{3.4}$$

Je-li $x = 4$, pak

$$M(4) = 1 + \frac{4}{3}(2 - 2\ln(2)) = \frac{11 - 4\ln 2}{3} \doteq 2{,}7425.$$

Na dalších intervalech délky 1 získáme hodnoty funkce postupným integrováním. Kdybychom ovšem chtěli najít například $M(1000)$, proces by byl opravdu zdlouhavý. Jacob proto začal přemýšlet o výhodnějším způsobu, jak výsledek nalézt. Co kdybychom vytvořili diferenciální rovnici? Do rovnosti (3.3) místo x dosadíme $x+1$, abychom mohli lépe derivovat a vynásobíme tuto rovnost pomocí x:

$$xM(x+1) = x + 2\int_0^{x} M(t)\,\mathrm{d}t.$$

Následně zderivujeme:

$$xM'(x+1) + M(x+1) = 1 + 2M(x), \tag{3.5}$$

dostaneme hezký tvar diferenciální rovnice, ve kterém neznámá funkce a její derivace vystupují při různých hodnotách argumentu. Jedná se proto o diferenciální rovnici se zpožděním. Jelikož lze opět místo x psát $x-1$ a předpokládat, že $x \geq 1$, pak snadno převedeme rovnici do standardního tvaru:

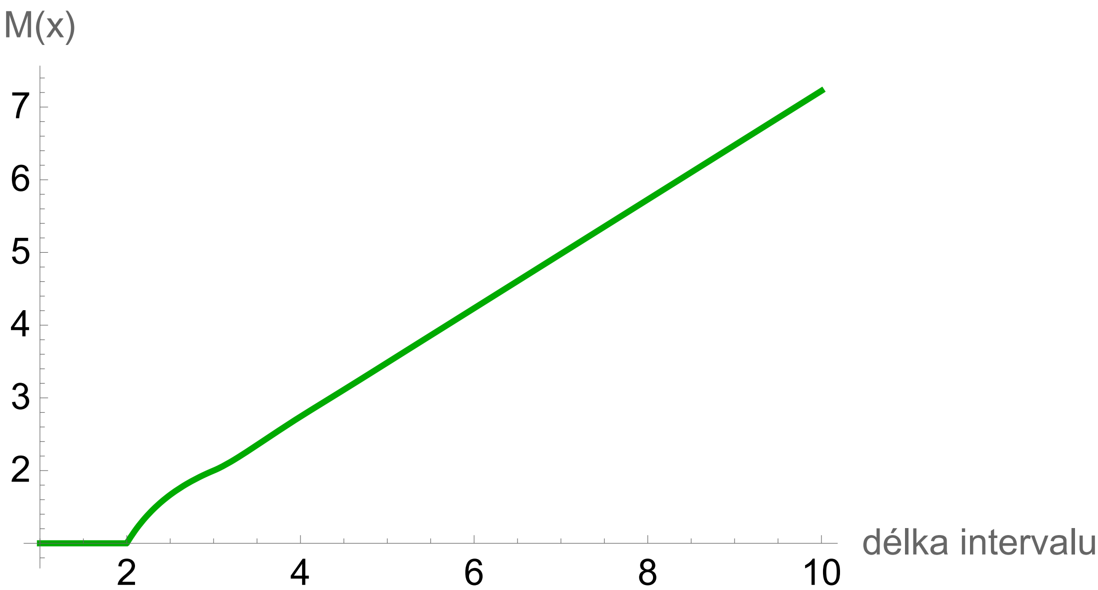
$$M'(x) = \frac{2M(x-1) + 1 - M(x)}{(x-1)}. \tag{3.6}$$

Jacob se těšil, až tuto diferenciální rovnici numericky vyřeší. Ada mu nabídla pomoc, aby řešení šlo rychleji od ruky. Rovnici řešíme na intervalu $[1, 10]$. Protože jde o diferenciální rovnici se zpožděním, počáteční podmínka není hodnota v bodě, ale hodnota na intervalu. Pro výpočet jsme zvolili interval $[1, 2]$.

```
(*Program 9: numerické řešení diferenciální rovnice se zpožděním*)

NDSolveValue[{M'[x] == (2 M[x - 1] + 1 - M[x])/(x - 1),
M[x /; x <= 2] == 1}, M[x], {x, 1, 10}]
```

Program vrátí numerické řešení rovnice. Pro lepší představu vykreslíme graf funkce $M(x)$ na intervalu $[1, 10]$.

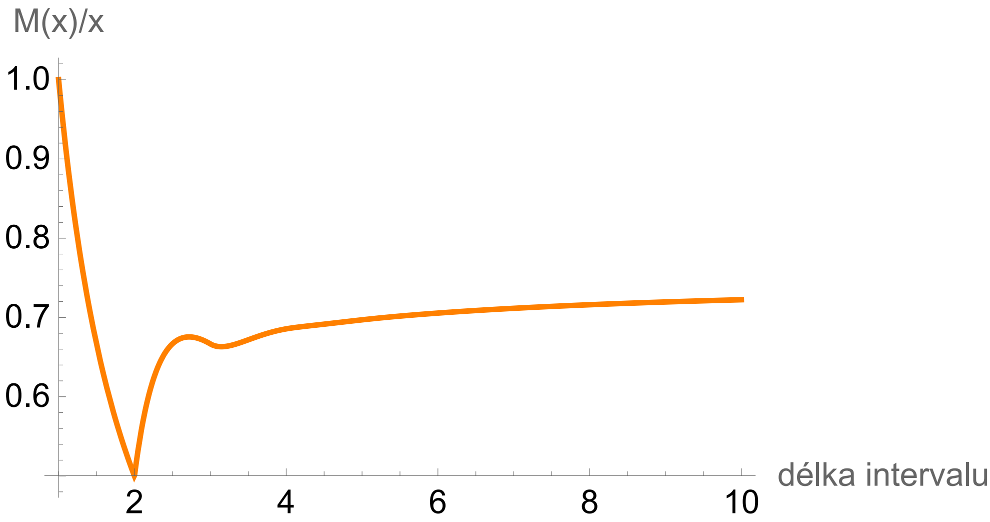


Obrázek 3.3: Znázornění hodnot funkce $M(x)$.

Co kdybychom chtěli zjistit, jaká část parkoviště bude zaplněna vzhledem k intervalu $[0,x]$? Dáme do poměru střední hodnotu počtu aut $M(x)$ a velikost parkoviště x, tedy obsazenou část parkoviště udává podíl:

$$\frac{M(x)}{x}.$$

Pokud tento vztah znázorníme graficky, vidíme, že pro zvyšující se hodnoty x se funkce blíží přibližně k číslu 0,7.



Obrázek 3.4: Zaplnění parkoviště vzhledem k délce intervalu.

Zaznamenáme-li do tabulky několik hodnot pro vyšší x, vidíme postupné přiblížení k číslu 0,74.

x	31	32	33	34	35	36	37
$M(x)/x$	0,7394	0,7397	0,7399	0,7401	0,7403	0,7405	0,7407

Kdyby naše x nabývalo opravdu velkých hodnot, pak část parkoviště C, která bude zaplněná vzhledem k velikosti intervalu x, bude konvergovat k:

$$C = \lim_{x \to \infty} \frac{M(x)}{x} \simeq 0,74759.$$

Jacob Rossin prošel důkladně práci H. Solomona a H. Weinera [13], aby limitní hodnotu C našel přesně. Navíc také zjistil, že C lze vyjádřit za pomoci dvojného integrálu jako:

$$C = \int_0^\infty \exp\left[-2\int_0^t \left(\frac{1-e^{-u}}{u}\right) \mathrm{d}u\right] \mathrm{d}t \simeq 0,74759.$$

Postavíme-li velká parkoviště, bez ohraničených míst, kde lidé parkují dle svobodné vůle, jen přibližně $\frac{3}{4}$ parkoviště se zaplní. Systém by způsobil nejen velké ztráty při stavbě města, ale také by znepříjemnil běžné každodenní fungování všem obyvatelům. Pokud by byl například určitý počet osob přiřazen jednomu obytnému domu, zaparkují-li řidiči přes více míst, kde by pak zaparkovala zbylá autíčka? Jacob Rossin dal dohromady všechny podklady a vyhledal svého blízkého a zároveň výřečného kamaráda Henriho Cappellaia. Henri měl dlouholeté zkušenosti jak s matematikou, tak s nástrahami politické scény. Společně připravili argumentační strategii, která měla veřejnost přesvědčit o absurdním výroku Milo Lairda. Ačkoliv Jacob považoval veřejná setkání za nemilá, přičemž tato emoce byla stupňována faktem, že musí ještě někoho přesvědčovat, na toto setkání se dostavil, protože zoufalá doba si žádá zoufalé činy. Milo Laird nebyl schopen jeho výsledky nijak vyvrátit. Jacob se svým proslovem uspěl a plány na financování města v oblacích i na infrasturkturu vypadaly znovu příznivě, tedy minimálně do té doby než se objeví další vševědoucí politik.

Závěr

Stránky deníku Jacoba Rossina jsou bohužel omezené a podobně je tomu i se stránkami této práce. Pokud si ovšem Jacob někdy koupí další pracovní deník, mohl by do něj například zapsat a načrtnout problematiku couvajících autíček. Dále by autíčka v ulici mohla nejen couvat, ale znovu jet dopředu. Pravděpodobně by se už nejednalo o jednosměrné ulice, takže by byly méně bezpečné. Některé firmy už navíc začaly vyrábět teleportující se auta, takže by při parkování řidiči místo couvání mohli využít rychlejšího přemístění třeba o k míst dozadu. Dále by se Jacob rozhodně mohl zaměřit na auta různých velikostí. Do města dříve či později zamíří autobusy, motocykly či nákladní automobily. Také by bylo vhodné zohlednit potenciální nefunkčnost některých parkovacích míst. Spadne-li na parkoviště strom nebo sluní-li se na nějakém parkovacím místě kočka, auta by se překážce měla vyhnout.

Protože je Jacob Rossin velmi talentovaný matematik, existuje možnost, že bude odvolán ještě na jiné části projektu NN a infrastruktura připadne někomu dalšímu. Modifikace parkovacích problémů si tedy nadšený čtenář může promyslet i samostatně [2]. Třeba přijde na nové poznatky, které by život lidem v městě v oblacích usnadnily. Za tímto účelem doporučujeme zvídavým čtenářům prozkoumat také zdroje [17], [5, část 5.3.1] a [4]. Jsou-li čtenáři navíc milovníky geometrie, ocení i zcela jiný typ parkovacího problému v článku [1].

„Podstata matematiky spočívá v její svobodě.“
Georg Cantor

Literatura

[1] S. R. Blackburn, *The Geometry of Perfect Parking*. Dostupné online: http://personal.rhul.ac.uk/uhah/058/perfect_parking.pdf.

[2] J. Carlson, A. Christensen, P. E. Harris, Z. Jones, A. R. Rodríguez, *Parking Functions: Choose Your Own Adventure*, Coll. Math. J. (2021), 254–264.

[3] P. Diaconis, A. Hicks, *Probabilizing parking functions*, Adv. Appl. Math. 89 (2017), 125–155.

[4] R. Ehrenborg, A. Happ, *Parking cars of different sizes*, Amer. Math. Monthly 123 (2016), No. 10, 1045–1048.

[5] S. R. Finch, *Mathematical constants*, Cambridge University Press, 2003.

[6] J. Françon, *Acyclic and Parking Functions*, J. Comb. Theory (A) 18 (1975), 27–35.

[7] I. M. Gessel, S. Seo, *A refinement of Cayley's formula for trees*, Electron. J. Comb. 11 (2006), No. 2, paper no. R27.

[8] A. G. Konheim, B. Weiss, *An occupancy discipline and applications*, SIAM J. Appl. Math. 14 (1966), 1266–1274.

[9] Ch. Remling, *Parking Functions and the Binomial Theorem*, MathOverflow (version: 2014-06-03). Dostupné online: https://mathoverflow.net/q/168874.

[10] A. Rényi, *On a one-dimensional problem concerning random space filling*, Publ. Math. Inst. Hung. Acad. Sci. 3 (1958), 109–127.

[11] J. Riordan, *Ballots and trees*, J. Comb. Theory 6 (1969), 408–411.

[12] N. Shales, *Recursively counting a parking function*, Mathematics Stack Exchange (2018-04-03). Dostupné online: https://math.stackexchange.com/q/2719580.

[13] H. Solomon, H. Weiner, *A review of the packing problem*, Commun. Stat., Theory Methods 15 (1986), 2571–2607.

[14] R. P. Stanley, *Parking Functions and Noncrossing Partitions*, Electron. J. Comb. 4 (1996), No. 2, paper no. R20.

[15] R. P. Stanley, *Parking functions*, Department of Mathematics (2004), M.I.T., 2–375. Dostupné online: https://math.mit.edu/~rstan/transparencies/parking2.pdf.

[16] B. Tung, *The Efficiency of Random Parking Problem*, Mathematics Stack Exchange (version: 2015-05-15). Dostupné online: https://math.stackexchange.com/q/1284071.

[17] C. H. Yan, *Parking functions*. In M. Bóna (ed.): Handbook of Enumerative Combinatorics, CRC Press, 2015, 835–893.