Základy matematiky a statistiky pro humanitní obory

studijní text k předmětům PLIN004 a PLIN006

Vojtěch Kovář

Předměty jsou určeny studentům humanitního zaměření (především oboru Český jazyk se specializací počítačová lingvistika), kteří pro své studium potřebují nejnutnější základy matematiky a statistiky. Cílem předmětů je seznámit studenty humanitních oborů se základy vysokoškolské matematiky a statistiky. Hlavními probíranými oblastmi jsou výroková a predikátová logika, úvod do teorie množin, relace a funkce, základy formální lingvistiky, teorie grafů, popisná statistika a teorie pravděpodobnosti.

Obsah

1 Proč potřebují lingvisté matematiku

Matematika, jakou ji známe ze střední školy, se v mnohém liší od matematiky, s níž se budeme seznamovat v tomto předmětu. Zatímco hlavním cílem středoškolské matematiky je naučit se počítat s čísly, abychom byli schopni spočíst daně, spotřebu auta, úroky z hypotéky, případně věci, o kterých často moc nemáme představu, k čemu slouží (soustavy rovnic, matice a integrály), cílem vysokoškolské matematiky je zejména naučit studenty uvažovat abstraktně a v obecnostech. V tomto kurzu nejsou důležité konkrétní vědomosti a úkoly, které budeme procházet. Jde o způsob myšlení, který si budete muset osvojit, abyste byli schopni tyto úkoly řešit.

Tento způsob uvažování je velmi důležitý zejména v dnešní době, kdy prožíváme rychlou technologickou expanzi ve výpočetní technice a setkáváme se na každém kroku s nejrůznějšími automatickými nástroji, včetně nástrojů pro automatické zpracování přirozeného jazyka. Jako studenti lingvistiky zaměření na počítačové technologie se s takovými nástroji a lidmi kolem nich budete setkávat ještě mnohem častěji.

Matematika je základem všech technických oborů: slouží jako zásobárna abstraktních pojmů, znalostí o nich, přesných definic a poskytuje prostředky pro přesné dokazování tvrzení o těchto abstraktních pojmech. Znalosti o abstraktních pojmech pak můžeme snadno využít při řešení konkrétních problémů. Z tohoto důvodu mají všichni studenti technických oborů (včetně informatiky) základní vzdělání ve vysokoškolské matematice. Tím, že takovýto základ dostanete také, se vám otevřou dveře ke snažší komunikaci s technicky zaměřenými lidmi, budete schopni lépe pochopit způsob jejich uvažování a diskutovat s nimi o problémech na jiné úrovni než lingvisté bez matematického vzdělání. V konkrétním případě počítačové lingvistiky si usnadníte spolupráci s vývojáři nástrojů automatického zpracování jazyka a budete schopni podílet se na vývoji těchto nástrojů.

Kromě tohoto cíle předpokládáme, že vám kurz usnadní studium předmětů na Fakultě informatiky.

2 Definice - věta - důkaz

Jak jsme již naznačili, vysokoškolská matematika se nezabývá návody, jak něco spočítat (jako tomu bylo na střední škole), ale slouží zejména jako soubor poznatků o abstraktních pojmech. Z tohoto důvodu se ve většině učebnic vysokoškolské matematiky setkáváme se stylem výkladu **definice-věta-důkaz**.

Definicí se rozumí vymezení pojmu. Definice musí obsahovat pouze pojmy, které byly dříve definovány. Obecně v matematice pracujeme pouze s pojmy, které byly přesně definovány. Často je pojem vymezen jako libovolný objekt, který splňuje nějakou množinu logických výroků (tzv. axiomů) – viz dále.

Příkladem jednoduché definice může být např.: Prvočíslo je jakékoli přirozené číslo, které je dělitelné pouze jedničkou a samo sebou. Aby tato definice byla plnohodnotná, musíme však již mít definovány pojmy přirozené číslo, dělitelný a jednička. Nelze spoléhat na to, že tyto pojmy jsou intuitivně jasné (např. patří 0 mezi přirozená čísla?). V dalším výkladu osvětlíme, jak vyrobíme (zadefinujeme) všechny matematické objekty pouze na základě dvou pilířů matematiky – matematické logiky a teorie množin.

Věta je formulací poznatku o definovaných pojmech. Příkladem matematické věty může být např. Existuje právě jedno sudé prvočíslo. Opět je potřeba mít předem definován pojem sudý. (Jak?) Jednodušší věta bývá též označována jako lemma.

Důkaz slouží k ověření pravdivosti tvrzení, a to po jednotlivých krocích tak, aby nikdo nemohl pochybovat o tom, že věta je pravdivá. S využitím matematické logiky můžeme pojem důkazu formalizovat (formálně definovat důkaz). Více v dalším výkladu.

V tomto textu se výše popsaného stylu výkladu budeme držet spíše zřídka. Důvodem je snaha překlenout propast mezi matematikou a humanitními obory a zjednodušit uchopení matematických pojmů.

2.1 Typy důkazů

V matematických a informatických textech se setkáváme s různými typy důkazů: nejčastější jsou následující:

Přímý důkaz je strukturálně nejjednodušší – použitím definic a již dokázaných tvrzení odvodíme přímo znění věty. Příklad: Mějme definována přirozená čísla a základní operace na nich. Dále mějme definován pojem sudé číslo jako násobek 2 (x je sudé, pokud existuje takové přirozené k, že $x = 2*k$). Dokážeme tvrzení *10 je sudé číslo*. Kroky důkazu:

1. $10 = 5 * 2$ (ze základní aritmetiky přirozených čísel)

2. $5 * 2$ je sudé číslo (z definice – $k = 5$)

3. tedy 10 je sudé číslo

Nepřímý důkaz využívá logickou ekvivalenci (viz též dále) výroků $A \Rightarrow B$ a $\neg B \Rightarrow \neg A$. Jinými slovy, větu tvaru $A \Rightarrow B$ můžeme dokázat tím, že dokážeme tzv. obměnu $\neg B \Rightarrow \neg A$.

Důkaz sporem je velice podobný. Zde předpokládáme, že dokazovaná věta neplatí a použitím známých faktů odvodíme spor – nesmyslné tvrzení, např. $1 = 0$, nebo neplatnost některého z předpokladů. Příklad: Mějme definována přirozená čísla, základní početní operace, dělitele (x je dělitelem y, pokud existuje přirozené z tak, že $x * z = y$), kladná racionální čísla (r/s taková, že r a s jsou přirozená a nemají jiného společného dělitele než 1), a druhou odmocninu ($a^2 = b$, pokud $b * b = a$). Dokážeme sporem, že $\sqrt{2}$ není racionální číslo.

1. Předpokládejme, že $\sqrt{2}$ je racionální číslo.

2. tedy $\sqrt{2} = r/s$, kde r a s jsou přirozená a nemají společného dělitele (kromě 1)

3. úpravou dostáváme $\sqrt{2} * s = r$

4. $2 * s * s = r * r$ (umocněním na druhou)

5. Protože levá strana je sudá, pravá je také sudá, tedy r je sudé (dokažte si podrobněji)

6. Tedy $r = 2 * c$ pro nějaké přirozené c

7. nahrazením dostaneme $2 * s * s = 2 * c * 2 * c$ a po vydělení dvěma $s * s = 2 * c * c$

8. podle úvahy z kroku 5 je s také sudé

9. r i s jsou sudá, tedy mají společného dělitele 2, což je spor s předpokladem.

Matematická indukce je důkazová technika, kterou využíváme, pokud potřebujeme něco dokázat pro velmi dlouhou nebo nekonečnou posloupnost objektů. Více o indukci v následující kapitole.

3 Matematická logika

Matematická logika je univerzálním jazykem matematiky, všechny matematické definice, věty a důkazy je možno zapsat jazykem matematické logiky. Jeho hlavními výhodami jsou univerzálnost (stejný zápis všude na světě znamená totéž) a jednoznačnost (na rozdíl od přirozených jazyků, které jsou víceznačné – např. v mnohých oficiálních textech můžeme nalézt věty typu „k jednání potřebujete pas a řidičský průkaz nebo občanský průkaz", kdy není jasné, co si s sebou vlastně máte vzít).

V případě důkazů pak logika zavádí pojem elementárního kroku a formalizuje tak dříve zmíněnou vágní formulaci krok za krokem.

Existuje více druhů matematických logik. Pravděpodobně jste již slyšeli o výrokové a predikátové logice, existují ovšem i složitější typy logik, jako např. modální, temporální, intenzionální... Naším cílem v této kapitole je probrat základy výrokové a predikátové logiky a naučit se je prakticky používat pro čtení a zápis matematiky. Pro podrobnější vhled do temných zákoutí matematické logiky doporučujeme předmět Matematická logika na Fakultě informatiky.

3.1 Výroková logika

Základní jednotkou výrokové logiky je **výrok**. Výrokem chápeme tvrzení, o němž lze říci, zda je pravdivé nebo nepravdivé, jinými slovy, **lze mu přiřadit pravdivostní hodnotu**. (Nezáleží přitom na tom, jestli tuto pravdivostní hodnotu známe, tj. nemusíme vědět, zda je výrok pravdivý). Příkladem výroku mohou být např. věty „12345678915264591 je prvočíslo", „všechny krávy jsou modré" a podobně.

K zachycení pravdivosti výroku používáme přiřazení hodnoty 0 (nepravda) nebo 1 (pravda), někdy zapisujeme jako $v(A) = 1$ (výrok A platí).

Ke konstrukci složitějších výroků (také **výrokových formulí** nebo **formulí výrokové logiky**) z výroků jednodušších používáme *logické funkce*. Mějme výroky A a B. Definujeme výroky

- $\neg A$ **(negace)** s pravdivostními hodnotami
  - $v(\neg A) = 1$, pokud $v(A) = 0$,
  - $v(\neg A) = 0$, pokud $v(A) = 1$;
- $A \Rightarrow B$ **(implikace)** s pravdivostními hodnotami
  - $v(A => B) = 0$, pokud $v(A) = 1$ a $v(B) = 0$,
  - $v(A => B) = 1$ v ostatních případech.

Tyto logické spojky se nazývají **základní**, neboť jejich kombinací lze vyjádřit všechny ostatní logické funkce.[1] Všechny ostatní logické funkce nazýváme **odvozené**. Mezi nejpoužívanější patří:

- $A \wedge B$ **(konjunkce, též logické „a")** s pravdivostními hodnotami
  - $v(A \wedge B) = 1$, pokud $v(A) = 1$ a $v(B) = 1$,
  - $v(A \wedge B) = 0$ v ostatních případech;
- $A \vee B$ **(disjunkce, též logické „nebo")** s pravdivostními hodnotami
  - $v(A \vee B) = 0$, pokud $v(A) = 0$ a $v(B) = 0$,
  - $v(A \vee B) = 1$ v ostatních případech;
- $A \Leftrightarrow B$ **(ekvivalence)** ve významu $(A \Rightarrow B) \wedge (B \Rightarrow A)$

Implikaci čteme jako „jestliže A, pak B", ekvivalenci čteme jako „A platí právě tehdy, když platí B". Pozor na přesnou sémantiku implikace: pokud je na levé straně nepravdivý výrok, pak je celá implikace pravdivá, tedy výrok „jestliže jsou všechny krávy modré, pak je uhlí bílé" by byl pravdivý.

Pravdivost různých složených výroků zkoumáme nejčastěji pomocí pravdivostních tabulek, které znázorňují pravdivostní hodnoty složených výroků v závislosti na pravdivostních hodnotách jednodušších výroků a které jistě znáte ze střední školy. Pravdivostní tabulka pro výrok $(A \vee B) \Rightarrow C$ vypadá např. takto:

A	B	C	$(A \vee B)$	$(A \vee B) \Rightarrow C$
1	1	1	1	1
1	1	0	1	0
1	0	1	1	1
1	0	0	1	0
0	1	1	1	1
0	1	0	1	0
0	0	1	0	1
0	0	0	0	1

(Dejte vždy pozor na to, že je nutné vypsat všechny kombinace pravdivostních hodnot základních výroků.)

Výrok, který je za všech okolností platný (v pravdivostní tabulce má všude jedničky), se nazývá **tautologie**. Výrok, který neplatí nikdy (v pravdivostní tabulce má všude nuly) se nazývá **kontradikce**.

[1] Nicméně, existují i jiné dvojice a dokonce i samostatné spojky, které by mohly být podle tohoto kritéria označeny za „základní"; volba negace a implikace je dána konvencí.

Výroková logika nám též poskytuje základ pro formalizaci pojmu *důkaz*. Využíváme k tomu tzv. **axiomy** a **odvozovací pravidla**. Standardní podoba výrokové logiky nám poskytuje tři schémata axiomů (kde dosazením konkrétních výroků vzniknou axiomy) a jedno odvozovací pravidlo. Schémata axiomů jsou:

1. $A \Rightarrow (B \Rightarrow A)$
2. $(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))$
3. $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$

(Jako cvičení si pravdivostní tabulkou můžete ověřit, že všechny uvedené axiomy jsou vždy pravdivé, nezávisle na pravdivosti základních výroků.)

Odvozovací pravidlo máme jediné, a to **modus ponens**:

- pokud platí A a platí $A \Rightarrow B$, odvodíme, že platí B

Důkazem potom rozumíme posloupnost výroků, z nichž každý je buď axiomem nebo aplikací odvozovacího pravidla na předchozí výroky. Příkladem formálního důkazu je následující posloupnost výroků, která dokazuje, že pro libovolný výrok X platí $X \Rightarrow X$:[2]

1. $(X \Rightarrow ((X \Rightarrow X) \Rightarrow X)) \Rightarrow ((X \Rightarrow (X \Rightarrow X)) \Rightarrow (X \Rightarrow X))$ / axiom 2
2. $X \Rightarrow ((X \Rightarrow X) \Rightarrow X)$ / axiom 1
3. $(X \Rightarrow (X \Rightarrow X)) \Rightarrow (X \Rightarrow X)$ / aplikace modus ponens na 2. a 1.
4. $X \Rightarrow (X \Rightarrow X)$ / axiom 1
5. $X \Rightarrow X$ / aplikace modus ponens na 4. a 3.

Samozřejmě všechny důkazy nebudeme provádět takto formálně – budeme se vždy spoléhat na to, že už něco víme. Výše uvedený příklad ale ukazuje, jaké jsou základní stavební kameny dokazování v matematice – každý důkaz může být rozepsán až na základní axiomy a aplikace odvozovacího pravidla, jako je tomu výše, a může být plně mechanicky ověřena jeho správnost.

V následující kapitole popíšeme některé aspekty predikátové logiky, se kterými se můžete často setkat.

[2]Ponechme nyní stranou fakt, že dokazovaný výrok je tautologie, což můžeme snadno ověřit pravdivostní tabulkou.

3.2 Predikátová logika

Predikátová logika je rozšířením logiky výrokové. Zavádí zejména **proměnné**, **kvantifikátory** a **predikáty**.

Proměnné, jak název napovídá, zavádí symboly, které mohou nabývat různých hodnot – nejčastěji je značíme malými písmeny. Formule predikátové logiky mohou obsahovat proměnné a jejich pravdivost může záviset na **ohodnocení** těchto proměnných, tj. na přiřazení konkrétních hodnot proměnným (např. formule „$x = 1$": pokud je proměnné x přiřazena 1, je formule pravdivá, jinak je nepravdivá).

Kvantifikátory nám umožňují výskyty proměnných svázat tak, že pravdivost formule již nebude závislá na ohodnocení. Rozeznáváme dva kvantifikátory:

- **univerzální** ($\forall$), který vyjadřuje, že následující formule platí pro všechna možná ohodnocení proměnné za kvantifikátorem. Např. formule $\forall x(x = 1)$ je nepravdivá, neboť zřejmě existuje ohodnocení, kde např. $x = 0$ a formule $x = 1$ je nepravdivá.

- **existenční** ($\exists$), vyjadřující, že existuje alespoň jedno ohodnocení takové, že následující formule platí. Např. formule $\exists x(x = 1)$ je pravdivá, neboť výrok $x = 1$ platí při valuaci, kde $x = 1$.

Predikáty jsou veškeré symboly, které nepatří do samotného jazyka logiky – tj. vše kromě proměnných, logických spojek, kvantifikátorů a závorek. Význam predikátů záleží na naší definici, není dán samotnou logikou. Příkladem predikátů může být např. $Prime$, kde $Prime(x)$ budeme chápat tak, že x je prvočíslo, nebo symbol $\in$, který budeme dále používat jako symbol příslušnosti do množiny (jako $\in (x, X)$ nebo čitelněji, $x \in X$). Každý predikát má definovánu aritu, což je počet jeho parametrů ($Prime$ má aritu 1, $\in$ má aritu 2). Predikáty arity 0 nazýváme **konstantami** (např. „1" v příkladu výše).

Příkladem složitější predikátové formule může být např. definice prvočísla (predikátu $Prime$) jen s využitím základních operací na přirozených číslech:[3]

$$Prime(x) \equiv x \geq 2 \wedge \forall y(\forall z((y * z = x) \Rightarrow (z = x \vee z = 1)))$$

Po částech lze formuli „přeložit" jako: prvočíslo je číslo větší nebo rovno 2, kde pro všechny možné dělitele (y) platí, že ve všech možných rozkladech x na $(y * z)$ je vždy jeden z členů tohoto rozkladu 1 a druhý x.

[3] rovněž zde prozatím předpokládáme, že všechny proměnné jsou z domény přirozených čísel

3.3 Matematická indukce

Matematická indukce je v prvé řadě důkazová technika, kterou můžeme dokázat nějaké tvrzení pro všechny prvky nějaké posloupnosti (i nekonečné). Princip indukce je však využíván často i v definicích velkých (nekonečných) struktur, jak uvidíme v dalším výkladu.

Mějme nějakou posloupnost libovolných objektů $x_0, ..., x_n, ...$ a chceme dokázat, že pro všechny prvky dané posloupnosti platí nějaká formule F, tedy $\forall i(F(x_i))$, kde $F(x_i)$ chápeme tak, že formule F platí pro prvek posloupnosti x_i. Technika důkazu indukcí se skládá ze dvou základních částí:

- **báze indukce**: Dokážeme, že formule platí pro první prvek posloupnosti.
- **indukční krok**: Dokážeme implikaci $(F(x_i) \Rightarrow F(x_{i+1}))$, tedy pokud formule platí pro nějaký prvek posloupnosti, pak platí i pro jeho bezprostředního následníka. Levou stranu předchozí implikace nazýváme **indukční předpoklad**.

Příklad: Dokážeme, že pro všechna přirozená $n \geq 1$ platí:

$$1 + 2 + ... + n = n/2 * (1 + n)$$

Báze indukce: za n dosadíme první prvek posloupnosti, tedy 1. Dostáváme $1 = 1/2 * (1 + 1)$, což je evidentně pravda.

Indukční krok: Předpokládejme (indukční předpoklad), že pro nějaké k platí

$$1 + 2 + ... + k = k/2 * (1 + k)$$

Dokážeme, že platí

$$1 + 2 + ... + k + (k + 1) = (k + 1)/2 * (1 + (k + 1))$$

Dosazením indukčního předpokladu do levé strany a následujícími úpravami podle aritmetiky nad přirozenými čísly postupně dostáváme:

- $1 + 2 + ... + k + (k + 1)$
- $k/2 * (1 + k) + (k + 1)$
- $(k + k^2)/2 + (k + 1)$

- $(k + k^2 + 2k + 2)/2$
- $(k^2 + 3k + 2)/2$
- $(k + 2) * (k + 1)/2$
- $(k + 1)/2 * (k + 2)$
- $(k + 1)/2 * (1 + (k + 1))$

Tím je věta dokázána.

Je namístě se ptát: Je tento postup korektní? Opravdu touto technikou dokážeme nějaké tvrzení pro všechny prvky příslušné posloupnosti?

Je třeba ověřit, že technika matematické indukce je korektní. Na tomto místě se spokojíme pouze s intuitivním ověřením a prohlášením, že formální důkaz existuje, ale je nad rámec předmětu.

Ověřením báze je dokázáno, že formule F platí pro x_0, platí tedy $F(x_0)$. Protože jsme dokázali indukční krok, platí také formule $F(x_0) \Rightarrow F(x_1)$. Aplikací pravidla modus ponens na předchozí dva výroky dostáváme, že platí i formule $F(x_1)$. Protože indukční krok jsme dokázali pro libovolné k, platí i $F(x_1) \Rightarrow F(x_2)$. Opět aplikací pravidla modus ponens na předchozí dva výroky dostáváme platnost formule $F(x_2)$. Takto můžeme pokračovat do nekonečna a dopracovat se k platnosti formule $F(x_i)$ pro libovolné přirozené i, platí tedy $\forall i(F(x_i))$.

Poměrně často se můžeme setkat i se složitějšími typy indukce; například nám někdy nestačí v indukčním kroku předpokládat, že daná formule platí pro jeden bezprostředně předcházející prvek, ale potřebujeme dva nebo i více předcházejících prvků. Princip indukce se tím nemění, pouze musíme dokázat odpovídající bázi (tj. pro 2 nebo více prvních prvků, abychom byli schopni poprvé aplikovat pravidlo modus ponens – viz předchozí odstavec).

Jak jsme již naznačili výše, princip indukce se dá dobře využít i v definicích, zejména pokud se jedná o nekonečné struktury, množiny a podobně. Zde to vypadá tak, že výčtem definujeme jeden nebo více nejjednodušších prvků dané struktury a analogií indukčního kroku specifikujeme, jak z jednodušších objektů vytvářet objekty složitější. Příkladem může být definice číselných výrazů se sčítáním a násobením:

- každé číslo je výraz (*báze*)
- pokud x a y jsou výrazy, pak $(x + y)$ je výraz
- pokud x a y jsou výrazy, pak $(x * y)$ je výraz

Jak vidíme, definice výše má jednu bázi a dva indukční kroky.

Pokud bychom potřebovali dokázat nějaké tvrzení pro celou takto definovanou strukturu, použijeme techniku zvanou **strukturální indukce**, která postupuje tzv. po struktuře objektu, podle jeho definice. Konkrétně dokážeme dané tvrzení nejprve pro všechny prvky z báze dané definice, a pak dokazujeme implikaci „pokud tvrzení platí pro jednodušší objekty, platí i pro složitější objekt", přesně podle struktury definice. Důkaz podle definice číselných výrazů výše by měl jednu bázi a dva indukční kroky. Důkaz indukcí tedy můžeme využít nejen pro dokazování vlastností posloupností, ale i složitějších struktur jako jsou výrazy, logické formule apod.

Jako cvičení si zkuste dokázat, že libovolný číselný výraz podle výše uvedené definice obsahuje sudý počet závorek.

4 Teorie množin

Spolu s matematickou logikou, která tvoří jakýsi jazyk popisu matematického světa, je teorie množin základním stavebním kamenem matematiky. Téměř všechny matematické objekty je možno definovat jako množiny, včetně čísel, posloupností, funkcí, automatů apod., jak uvidíme v dalším výkladu. Navíc všechny tyto objekty lze zkonstruovat na základě jediného základního objektu, prázdné množiny $\emptyset$, tedy množiny, která neobsahuje žádné prvky.

Teorie množin doplňuje jazyk predikátové logiky o binární predikát **je prvkem ($\in$)**:

$a \in b$ čteme „a je prvkem b";

dále pak o konstantní symbol **prázdné množiny ($\emptyset$)** a dále o prostředky pro zápis množinových objektů, zejména složené závorky ({}).

4.1 Množina

Teorie množin zavádějí množiny jako objekty, které vyhovují určitým axiomům – takovýmto teoriím říkáme *axiomatické teorie množin* a jsou tou „správnou" cestou, jak definovat matematické objekty. Příkladem axiomu z teorie množin je axiom vydělení, který vyjadřuje, že z libovolné množiny můžeme vybrat některé prvky, pro které platí formule F, a tím vytvořit jinou množinu:

$$\forall a(\exists b(\forall x(x \in b \Leftrightarrow (x \in a \wedge F(x)))))$$

(Díky tomuto axiomu např. existuje prázdná množina.) V současnosti existuje více axiomatických teorií množin, které se od sebe mírně liší tím, jaký výčet axiomů je v každé z nich použit.

V tomto textu se nebudeme dopodrobna zabývat žádnou z axiomatických teorií množin. Půjde nám hlavně o to, abychom se naučili množiny prakticky používat při definicích a zápisech matematických faktů a spokojíme se s tzv. „naivní" teorií množin, s níž jste se pravděpodobně setkali na střední škole:[4] **Množinu budeme považovat za *skupinu objektů* (které nemusí být nutně stejného typu), jež neobsahuje duplicity (tj. každý prvek**

[4]Problémy s naivní teorií množin nastávají, pokud začneme používat obskurní definice množin – můžeme si je ilustrovat na příkladu vojenského holiče: Řekneme-li, že holič holí všechny vojáky, kteří se neholí sami (a přitom holič sám je voják), kdo holí holiče? Matematická obdoba této jazykové hříčky se nazývá Russelův paradox a axiomatické teorie množin právě kvůli tomuto paradoxu definují přesné postupy, jak mohou vznikat množiny. Množinám, které nesplňují axiomy teorie množin, typicky říkáme *třídy*.

je v ní obsažen maximálně jednou) a není uspořádaná ($\{1, 2, 3\}$ **a** $\{3, 2, 1\}$ **jsou totožné množiny)**.

Zejména si uvědomme tři základní fakta:

1. existuje prázdná množina $\emptyset$, která neobsahuje žádné prvky;
2. prvky množiny mohou být i jiné množiny;
3. množiny mohou být nekonečné (obsahovat nekonečně mnoho prvků).

Malé konečné množiny obvykle zapisujeme výčtem prvků – např. množina $\{1, 2, 3\}$ obsahuje prvky 1, 2 a 3, množina $\{\emptyset, \{\emptyset\}\}$ obsahuje dva prvky – $\emptyset$ (prázdná množina) a $\{\emptyset\}$ (množina obsahující prázdnou množinu). Velké nebo nekonečné množiny můžeme zapisovat s využitím formule, která musí platit pro všechny členy množiny. Např. množina $\{x \mid x \in \mathbb{N} \quad \wedge \quad x > 5\}$ obsahuje přirozená čísla větší než 5. Jako zkratka pro tento zápis se též často používá$\{x \in \mathbb{N} \mid x > 5\}$.[5]

4.2 Množinové operace

Jak jsme již uvedli, základní operací na množinách je operace příslušnosti do množiny $\in$. Zdůrazněme, že na levé straně je vždy prvek (který může být množinou), na pravé straně je vždy množina. Doplňkem operátoru $\in$ je $\notin$, které definujeme takto:

$$a \notin B \quad \Leftrightarrow \quad \neg(a \in B)$$

Příklady platných formulí s těmito operátory:

- $\forall x(x \notin \emptyset)$
- $\emptyset \in \{\emptyset\}$
- $\emptyset \notin \{\{\emptyset\}\}$
- $\emptyset \notin \emptyset$

Další operací (predikátem), kterou můžeme snadno definovat, je **podmnožina** ($\subseteq$). Definujeme ji takto:

[5]Pokud budeme důsledně používat podmínku, že všechny prvky námi definované množiny patří do nějaké jiné množiny, máme jistotu, že se vyhneme Russelovu paradoxu – tímto postupem de facto využíváme axiom vydělení uvedený výše.

$$A \subseteq B \quad \Leftrightarrow \quad \forall x(x \in A \Rightarrow x \in B)$$

Tedy množina A je podmnožinou množiny B právě tehdy, pokud všechny prvky A jsou zároveň i prvky B.

Na tomto místě poznamenejme, že pro predikátové formule tohoto typu často používáme zkrácený zápis:

$$\forall x \in A\ (x \in B),$$

který by v takovéto podobě nebyl v čisté predikátové logice možný. Podobně, pro existenční kvantifikátor často používáme zkrácený zápis např.

$$\exists x \in A\ (x \in B),$$

ale v trochu jiném významu: Výraz výše je ekvivalentní formuli

$$\exists x(x \in A \wedge x \in B),$$

Na základě pojmu podmnožina definujeme pojem **potenční množina**. Potenční množina množiny A – zapisujeme $\mathcal{P}(A)$ – je možina všech podmnožin této množiny. Formálně:

$$\mathcal{P}(A) = \{x \mid x \subseteq A\}$$

Protože potenční množina má vždy 2^a prvků,[6] kde a je počet prvků množiny A, někdy značíme potenční množinu množiny A jako 2^A.

Platí:

- $\mathcal{P}(\emptyset) = \{\emptyset\}$
- $\mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$
- $\forall x(\emptyset \in \mathcal{P}(x) \wedge x \in \mathcal{P}(x))$

S využitím pojmu podmnožiny se definuje i rovnost množin:

$$A = B \quad \Leftrightarrow \quad (A \subseteq B \ \wedge \ B \subseteq A)$$

Tuto kapitolu zakončíme definicí dvou operací, které znáte ze střední školy, **průnik** ($\cap$) a **sjednocení** ($\cup$). Než budete číst dál, zkuste si je definovat sami.

[6] Rozmyslete si proč.

Správné definice jsou:

$$A \cap B = \{x \mid x \in A \quad \wedge \quad x \in B\}$$
$$A \cup B = \{x \mid x \in A \quad \vee \quad x \in B\}$$

5 Čísla

Na střední škole jsme se seznámili s některými číselnými množinami – přirozenými čísly, celými čísly, racionálními, reálnými a komplexními čísly a učili jsme se s nimi počítat. V minulé kapitole jsme zmínili, že všechny objekty v matematice, včetně čísel, jsou množiny. V této kapitole si tedy ukážeme, jak lze z množin vyrobit přirozená čísla a definovat operace na nich.

5.1 Přirozená čísla

Podobně jako jsou množiny ve formálních teoriích množin definovány axiomy v predikátové logice, jsou i přirozená čísla definována jako objekty, které splňují jisté formální axiomy. Zatímco ve výkladu o množinách jsme tyto axiomy neuváděli, v případě čísel si je již uvedeme – jedná se o jednoduchý axiomatický systém, který nám umožní lépe pochopit, jak funguje formální matematika.

Jazyk přirozených čísel zavádí do predikátové logiky dva nové symboly:

- **nulu** 0 a
- **unární funkční symbol** S, který budeme interpretovat jako **následníka**.

Přirozená čísla jsou pak určena množinou axiomů, které nazýváme **Peanova aritmetika** (po italském matematikovi Peanovi). Tyto axiomy jsou:

- $\exists x(x = 0)$ (existuje nula)
- $\forall x(\exists y(y = S(x)))$ (každé číslo x má následníka $S(x)$)
- $\forall x(\neg(0 = S(x)))$ (nula není následníkem žádného čísla)
- $\forall x(\forall y(S(x) = S(y) \Rightarrow x = y))$ (různá čísla mají různé následníky)

Nyní definujeme množinový systém, který bude splňovat výše uvedené axiomy a tím dostaneme „fyzická” přirozená čísla:

- $0 \equiv \emptyset$
- $S(x) \equiv x \cup \{x\}$

Tedy nulu definujeme jako prázdnou množinu a následníka libovolného čísla definujeme jako sjednocení tohoto čísla (resp. množiny odpovídající tomuto číslu) s jednoprvkovou množinou, která toto číslo (resp. množinu) obsahuje.

Jak tedy vypadají přirozená čísla v množinové notaci?

- $0 \equiv \emptyset$
- $1 = S(0) = \emptyset \cup \{\emptyset\} = \{\emptyset\}$
- $2 = S(1) = \{\emptyset\} \cup \{\{\emptyset\}\} = \{\emptyset, \{\emptyset\}\}$
- $3 = S(2) = ... = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$

atd. Zkuste si jako cvičení vypsat ještě několik dalších. Zjistíte, že pro každé přirozené číslo n platí, že $n = \{0, 1, ..., n-1\}$. Zkuste rovněž navrhnout důkaz, že pro takovýto množinový systém platí Peanovy axiomy uvedené výše.

5.2 Operace na přirozených číslech

Nyní si ukážeme, jak lze na systému přirozených čísel definovat základní operace, sčítání ($+$) a násobení ($*$). Obě definice jsou induktivní:

- $a + 0 = a$
- $a + S(b) = S(a + b)$

- $a * 0 = 0$
- $a * S(b) = (a * b) + a$

Jak vidíme, tyto operace jsou nezávislé na konkrétní (množinové) realizaci a pracují pouze s operací následníka. Nadále tedy množinovou realizaci přirozených čísel již nemusíme používat a můžeme pracovat pouze s abstrakcí následníka. Jak tedy bude vypadat například výpočet 1+2 podle výše uvedené definice? (platí $1 = S(0), 2 = S(1) = S(S(0))$ – tyto rovnosti chápeme pouze jako různé zápisy téhož)

- $1 + 2$
- $1 + \mathrm{S}(1)$
- $\mathrm{S}(1 + 1)$
- $\mathrm{S}(1 + \mathrm{S}(0))$
- $\mathrm{S}(\mathrm{S}(1 + 0))$
- $\mathrm{S}(\mathrm{S}(1))$
- $\mathrm{S}(\mathrm{S}(\mathrm{S}(0)))$
- $= 3$

Vyzkoušejte si i jiné výpočty podle uvedených definic. Zejména si všimněte, že nemůžeme zaměnit $1 + 2$ a $2 + 1$ (i když druhý z těchto výpočtů podle definice by byl kratší a i když ze střední školy víme, že to platí), neboť nemáme definováno žádné pravidlo, podle kterého bychom to mohli udělat, a komutativitu sčítání jsme zatím nedokázali.

5.3 Další číselné množiny

Další číselné množiny (celá, racionální, reálná čísla) jsou v matematice konstruovány s využitím dvojic, ekvivalencí a dalších pokročilejších matematických konstrukcí, kterým se budeme věnovat v dalších kapitolách. Proto i konstrukce dalších číselných množin bude vyložena později.

6 Relace a funkce

V této kapitole se budeme zabývat koncepty relací a funkcí, které lze dále používat pro složitější matematické konstrukce a operace.

6.1 Uspořádané dvojice

Základním pojmem této části je **uspořádaná dvojice**. Jak název napovídá, jedná se o spojení dvou prvků, v němž (narozdíl od dvouprvkové množiny) rozlišujeme první a druhý prvek. Uspořádanou dvojici s prvky a a b zapisujeme (a, b) a lze ji definovat jako množinu $\{\{a\}, \{a, b\}\}$. Touto definicí je zaručeno, že v každém případě víme, který z prvků je první a který druhý, jinými slovy, že dvě různé uspořádané dvojice nebudou reprezentovány stejnou množinou (důkaz je technicky složitější, proto jej zde nebudeme uvádět). Jsou možné i jiné definice, výše uvedená je ale nejpřímočařejší.

Lze definovat i uspořádané trojice nebo obecně n-tice (pro libovolné přirozené n), a to následovně:

- $(a, b, c) \equiv (a, (b, c))$, tedy trojice je totéž jako dvojice, jejímž prvním prvkem je první prvek trojice a druhým uspořádaná dvojice se zbylými dvěma prvky
- $(a_1, a_2, a_3, ..., a_n) \equiv (a_1, (a_2, (a_3, (..., a_n)...)))$

Tato definice n-tic má jisté nevýhody, proto si na konci kapitoly uvedeme alternativní definici s využitím funkcí (kde definujeme n-tice jako konečné posloupnosti).

6.2 Kartézský součin

Na základě pojmu uspořádané dvojice můžeme definovat **kartézský součin množin** $(A \times B)$, definovaný takto:

$$A \times B \equiv \{(a, b) \mid a \in A \ \wedge \ b \in B\}$$

Tedy kartézský součin dvou množin obsahuje (všechny) uspořádané dvojice takové, že první prvek je z první množiny a druhý prvek je z druhé množiny.

Analogicky můžeme definovat kartézský součin více množin, který bude obsahovat uspořádané n-tice.

6.3 Relace

Motivací pro pojem relace je způsob svázání dvou, případně více hodnot, možnost vyjádření toho, že některé hodnoty z obou množin mají něco společného. Relace také tvoří základ pro definici funkcí.

Binární relace mezi množinami A a B je podmnožina kartézského součinu $A \times B$ (tedy množina uspořádaných dvojic – podobně definujeme n-ární relaci jako množinu uspořádaných n-tic). Často říkáme **binární relace na množině A**, čímž rozumíme relaci množiny s ní samotnou, tedy podmnožinu kartézského součinu $A \times A$ (analogicky můžeme opět mluvit o n-ární relaci na množině A).

Příklady relací:

- Relace identity na množině A:
  - $Id(A)$ – binární relace
  - $Id(A) = \{(a, a) \in A \times A \mid a \in A\}$
- Relace větší nebo rovno na přirozených číslech:
  - $\geq (N)$ – binární relace
  - $\geq (N) = \{(a, b) \in N \times N \mid b \subseteq a\}$
  - (srovnejte s množinovou konstrukcí přirozených čísel)
- Relace plus na přirozených číslech:
  - $+(N)$ – ternární relace
  - $+(N) = \{(a, b, c) \in N \times N \times N \mid a + b = c\}$
  - $a+b = c$ pak můžeme prohlásit jen za jiný zápis pro $(a, b, c) \in +(N)$ a všechny operace na číslech považovat za relace

Relace na malých konečných množinách můžeme přehledně zapisovat tabulkou, kde prvky množiny jsou v záhlaví řádků i sloupců a příslušnost dvojice v relaci značíme jedničkou nebo nulou v příslušné buňce tabulky. Přehledný je rovněž zápis grafem, kdy mezi prvky množiny kreslíme orientované šipky, podle toho, které dvojice jsou v relaci.

6.4 Vlastnosti relací

Lze definovat některé vlastnosti relací, které jsou motivovány jejich následným použitím. Všechny vlastnosti jsou definovány výrokovými formulemi, jejich

význam lze ale většinou vyjádřit srozumitelnějším jazykem. n-ární **Reflexivita** vyjadřuje, že každý prvek je v relaci sám se sebou. Relace R na množině A je reflexivní, pokud platí

$$\forall a \in A \; (\; (a, a) \in R \;)$$

Symetrie, jak název napovídá, je vlastnost, která vynucuje, že ke každé dvojici v relaci existuje symetrická dvojice, tedy taková, kde první a druhý prvek jsou prohozeny. Relace R na množině A je symetrická, pokud platí

$$\forall a, b \in A \; (\; (a, b) \in R \Rightarrow (b, a) \in R \;)$$

Antisymetrie je opakem symetrie; říká, že relace neobsahuje žádné dvě symetrické dvojice (s výjimkou takových, kde první a druhý prvek jsou stejné). Formálně, relace R na množině A je antisymetrická, pokud platí

$$\forall a, b \in A \; (\; (a, b) \in R \wedge (b, a) \in R \Rightarrow a = b \;)$$

Tranzitivita je komplikovanější. Relace je tranzitivní, pokud ke každým dvěma dvojicím (a, b) a (b, c), které jsou v relaci, existuje v relaci i dvojice (a, c); tranzitivita tedy vyjadřuje jistý typ souvislosti relace. V případě zápisu grafem si tuto vlastnost lze představit tak, že dojdeme-li z nějakého bodu do jiného podle šipek, pak musí existovat zkratka, tedy šipka přímo ze startu do cíle. Formálně, relace R na množině A je tranzitivní, pokud platí

$$\forall a, b, c \in A(\; (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R \;)$$

Ekvivalence je taková relace, která je současně reflexivní, symetrická i tranzitivní.

Uspořádání je taková relace, kteráje současně reflexivní, antisymetrická i tranzitivní.

Například identita na libovolné množině splňuje všechny uvedené vlastnosti (rozmyslete si postupně proč), je tedy ekvivalencí i uspořádáním. Relace „menší nebo rovno" na přirozených číslech je reflexivní, antisymetrická a tranzitivní (opět si rozmyslete proč), je tedy uspořádáním. Hypotetická relace „sedí vedle" na množině studentů v přednáškové místnosti není tranzitivní, není tedy ekvivalencí ani uspořádáním.

6.5 Funkce

Funkce je speciální typ relace; dalo by se říci, že „býti funkcí" je vlastnost relací podobně jako například tranzitivita. Funkce je taková (n-ární) relace, kde prvních $n-1$ hodnot v n-tici jednoznačně určuje poslední hodnotu.

Alternativně se můžeme na relaci podívat jako na vstupně-výstupní mechanismus: prvních $n-1$ hodnot v n-tici můžeme pokládat za argumenty relace (vstup), poslední hodnotu za její výsledek (výstup – srovnejte např. s relací $+(N)$ výše). Pokud má být taková relace funkcí, výstup musí být jednoznačně určen argumenty. Z tohoto pohledu vychází i speciální zápis funkcí: např. místo

$$(a, b, c) \in +$$

píšeme často v případě funkcí

$$+(a, b) = c$$

(stále uvažujeme ternární relaci $+(N)$). Z tohoto pohledu na funkce vychází i konvence ohledně arity funkcí, která se určuje podle počtu argumentů: Tedy unární funkce je binární relace, ternární relace je binární funkce apod.

Formálně, binární relace f na množině A je funkce, pokud platí:

$$\forall a, b, c \in A \; (\; (a, b) \in f \; \wedge \; (a, c) \in f \Rightarrow b = c \;)$$

Obdobně, ternární relace f na množině A je funkce, pokud:

$$\forall a, b, c, d \in A \; (\; (a, b, c) \in f \; \wedge \; (a, b, d) \in f \Rightarrow c = d \;)$$

Podobně si zkuste definovat funkce více argumentů.

Funkcím také někdy říkáme **zobrazení**. Pro binární relaci f mezi množinami A a B, která je funkcí, říkáme „funkce z A do B", případně „zobrazení z A do B" a zapisujeme

$$f : A \to B$$

Množinu A potom nazýváme **definičním oborem** a někdy značíme D_f nebo $dom(f)$ (ve významu „definiční obor funkce f"), množinu B nazýváme **oborem hodnot** a značíme R_f nebo $f(A)$. Již zmíněný zápis $f(a) = b$ můžeme číst jako „b je obraz prvku a", případně „a je vzor prvku b".

6.6 Vlastnosti funkcí

Také funkce mají některé speciální vlastnosti, které dále využíváme. Mezi nejužitečnější patří:

Injektivita. Funkce $f : A \to B$ je injektivní (též **prostá**), pokud platí

$$\forall a, b \in A \ (\ f(a) = f(b) \Rightarrow a = b\ )$$

neboli *žádné dva prvky nemají stejný obraz.*

Surjektivita. Funkce $f : A \to B$ je surjektivní (též **na**), pokud platí

$$\forall b \in B \ (\ \exists a \in A \ (b = f(a)\ )$$

neboli *každý prvek oboru hodnot má nějaký vzor*, případně můžeme říci, že *celý obor hodnot je pokrytý.*

Úplnost. Funkce $f : A \to B$ je úplná, pokud platí

$$\forall a \in A \ (\ \exists b \in B \ (b = f(a)\ )$$

neboli *celý definiční obor je pokrytý.* Často se můžete setkat s tím, že pojmem *funkce* je myšlena úplná funkce.

Řekneme, že funkce je **bijekce** právě tehdy, je-li současně injektivní, surjektivní a úplná. Bijekce se dobře používají mj. při porovnávání velikostí množin: Existuje-li bijekce $f : A \to B$, znamená to, že množiny A a B jsou „stejně velké" (za okamžik tuto vágní poznámku rozvedeme formálně).

Pro bijekci $f : A \to B$ můžeme dále zavést **inverzní funkci** $f^{-1} : B \to A$, která je definována následovně:

$$f^{-1}(b) = a \ \equiv \ f(a) = b$$

6.7 Velikost množin

Do této doby jsme velikost množiny chápali pouze vágně, jako počet jejích prvků. V případě nekonečných množin jsme neměli představu o velikosti vůbec. Nyní si zadefinujeme velikost množiny A (značeno $|A|$) formálně:

- $|A|$ je definováno jako přirozené číslo n, pokud existuje bijekce $f : n \to A$ (srovnejte s množinovou konstrukcí přirozených čísel)
- množina A je **spočetná**, pokud existuje bijekce $f : \mathbb{N} \to A$

- množina A má **mohutnost kontinua**, pokud existuje bijekce $f : \mathbb{R} \to A$ (kde $\mathbb{R}$ je množina reálných čísel)

Obecně, jak jsme již naznačili, pokud existuje bijekce z jedné množiny do jiné, pak tyto množiny mají stejnou velikost (též říkáme mohutnost).

Množiny $\mathbb{N}$, $\mathbb{Z}$ i $\mathbb{Q}$ (přirozená, celá i racionální čísla, stejně jako např. množina všech sudých čísel nebo množina všech prvočísel) jsou všechny spočetné. Příslušné bijekce zde uvádět nebudeme, zkuste se nad nimi zamyslet sami. Reálná čísla jsou striktně větší než čísla přirozená a jsou nejmenší takovou množinou (čili $|\mathbb{N}|$ a $|\mathbb{R}|$ jsou dvě nejmenší nekonečna). Důkaz je však již nad rámec našeho předmětu.

6.8 Posloupnosti

Posledním pojmem, jímž se v kapitole o funkcích budeme zabývat, jsou posloupnosti. Intuitivně tento pojem chápeme: jedná se o skupinu prvků, v níž (na rozdíl od množin) záleží na pořadí. Prvky se v posloupnosti také mohou opakovat. Již jsme zmínili, že konečné posloupnosti můžeme považovat za uspořádané n-tice; nyní si představíme jinou definici, která dobře generalizuje i na nekonečné posloupnosti.

Konečná posloupnost délky n je úplná funkce, jejímž definičním oborem je množina n (viz množinová konstrukce přirozených čísel). Nekonečná posloupnost je pak úplná funkce, jejímž definičním oborem je množina $\mathbb{N}$.

Takovéto posloupnosti typicky zapisujeme jako $a_0, a_1, ..., a_n, ...$, což považujeme jen za jiný druh zápisu pro $f(0), f(1), ..., f(n), ...$

V případě nekonečných posloupností často pracujeme s induktivními definicemi. V případě posloupností tyto definice vypadají tak, že vypíšeme první člen (případně prvních několik členů), což je analogie báze indukce, a poté určíme předpis, podle kterého dostaneme n-tý prvek pomocí jednoho, případně několika předchozích prvků (analogie indukčního kroku). Podle takovéto definice jsme schopni zkonstruovat libovolný prvek posloupnosti.

Pěkným příkladem nekonečné posloupnosti je tzv. Fibonacciho posloupnost, kde každý další člen je součtem dvou předchozích členů: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Induktivní definice Fibonacciho posloupnosti vypadá takto:

- $a_0 = 0$
- $a_1 = 1$
- $a_n = a_{n-1} + a_{n-2}$

7 Základy formální lingvistiky

Až dosud jsme se zabývali skutečnými základy, pilíři formální matematiky. Nyní si naopak ukážeme část aplikované matematiky, která se intenzivně využívá i v počítačovém zpracování přirozených jazyků – na matematické modely některých rysů jazyka neboli formální lingvistiku.

Tyto matematické modely považují jazyk za množinu slov nad nějakou abecedou, kde prvky abecedy mohou být znaky, slova (pak můžeme za jazyk považovat např. množinu správně utvořených vět), případně i jiné jednotky (morfémy, ...). Modely, jimiž se budeme zabývat, byly původně navrženy pouze k popisu přirozených jazyků; časem ale našly i jiné uplatnění (a naopak analýzu přirozených jazyků dosud uspokojivě nevyřešily) a dnes rozlišujeme tzv. formální jazyky, které jsou dobře analyzovány jednoduchými matematickými modely.

Naším cílem v této části bude seznámení s nejzákladnějšími konstrukcemi teorie formálních jazyků, neboť je velmi pravděpodobné, že se s nimi a jejich modifikacemi při studiu ještě mnohokrát setkáte.

7.1 Základní pojmy

Nejprve zde uvedeme několik základních pojmů, které teorie formálních jazyků používá:

Abeceda je konečná množina symbolů. Značíme ji Σ a pro jednoduchost budeme často pracovat s abecedou $\{a, b\}$.

Slovo je libovolná konečná posloupnost prvků Σ, např. $aabab$.

Délka slova je velikost této posloupnosti, např. $|aabab| = 5$.

Prázdné slovo je slovo nulové délky, značíme je ϵ.

Σ^* je množina všech slov nad abecedou Σ,
např. $\{a, b\}^* = \{\epsilon, a, b, aa, bb, ab, ba, aab, abb, ...\}$.

Operace zřetězení slov, značíme tečkou (.), je pro dvě slova u a v devinována jako $u \,.\, v = uv$, např. $aab \,.\, ab = aabab$.

Mocnina slova je pro slovo u a přirozené číslo i značena u^i a je definována induktivně:

- $u^0 = \epsilon$
- $u^{i+1} = u.u^i$

Např. $(ab)^3 = ababab$.

Jazyk je množina *některých* slov nad danou abecedou, tedy pro každý jazyk L platí $L \subseteq \Sigma^*$ (ale nikoli naopak).

Dále můžeme definovat i operace nad celými jazyky analogicky k operacemi nad slovy. Např. operaci **zřetězení jazyků** lze definovat následovně:

$$L_1 \,.\, L_2 = \{u \,.\, v \mid u \in L_1 \wedge v \in L_2\}$$

7.2 Formální gramatika

Formální gramatika je prvním z nástrojů formální lingvistiky. Můžeme si ji představit jako přepisovací systém, jímž lze vygenerovat slova jazyka (viz dále).

Formálně, gramatika je čtveřice (N, Σ, P, S), kde

- N je **množina neterminálů** (značíme nejčastěji velkými písmeny)
- Σ je **množina terminálů** (symbolů abecedy, pro přehlednost značíme nejčastěji pouze malými písmeny), je disjunktní s množinou N a $N \cup \Sigma$ označujeme jako V (množina všech symbolů)
- P je množina pravidel, formálně podmnožina kartézského součinu $V^* \,.\, N \,.\, V^* \times V^*$ – tj. množina dvojic, kde prvním prvkem je řetězec obsahující alespoň jeden neterminál a druhým prvkem je libovolný řetězec
- S je počáteční symbol gramatiky

Pravidla gramatiky jako dvojice řetězců (slov nad množinou V) (α, β) zapisujeme nejčastěji jako $\alpha \to \beta$. α nazýváme levou stranou pravidla a jak již bylo řečeno, musí obsahovat alespoň jeden neterminál. β nazýváme pravou stranou pravidla.

Jak jsme již naznačili, gramatika je modelem, kterým lze generovat slova jazyka. Toto generování probíhá následovně. Začneme z počátečního symbolu gramatiky S a pravidla gramatiky používáme jako přepisovací systém, to znamená, že v jednom kroku přepisu můžeme nahradit některý řetězec terminálů a neterminálů, který je současně na levé straně nějakého pravidla, pravou stranou tohoto pravidla. Tento postup opakujeme tak dlouho, dokud nedostaneme řetězec terminálních symbolů (čili slovo nad Σ). Tomuto procesu říkáme **odvození slova z gramatiky**.

Při přepisování máme často na výběr z více pravidel. Můžeme se rozhodnout pro kterékoli z nich a vygenerovat tak potenciálně různá slova.

Řekneme, že **gramatika** G **generuje jazyk** L, pokud existuje odvození každého slova jazyka L z gramatiky G. Jazyk generovaný gramatikou G značíme většinou $L(G)$.[7]

Příklad. Mějme gramatiku (N, Σ, P, S), kde

- $\Sigma = \{a, b\}$
- $N = \{S, A\}$
- $P = \{ \quad S \to A, \qquad A \to AA, \qquad A \to a \quad \}$

Příklady odvození z této gramatiky jsou:

- $S \Rightarrow A \Rightarrow a$
- $S \Rightarrow A \Rightarrow AA \Rightarrow aA \Rightarrow aAA \Rightarrow aaA \Rightarrow aaa$

Všimněte si, že jeden krok odvození z gramatiky (jednu aplikaci pravidla) značíme dle konvence stejným symbolem jako implikaci. Jedná se samozřejmě o dvě zcela různé věci a rozeznáme je od sebe jednoznačně na základě kontextu.

Zkuste vymyslet další odvození z této gramatiky. Jaký je jazyk generovaný touto gramatikou?

7.3 Chomského hierarchie jazyků

Pokud budeme zkoumat pravidla v různých gramatikách, získáme rozlišení gramatik na několik typů. Podle typů gramatik rozlišujeme též typy jazyků, podle toho, který jazyk je možné generovat kterým typem gramatiky. Toto rozlišení nazýváme **Chomského hierarchie gramatik**, ekvivalentně Chomského hierarchie jazyků, podle amerického lingvisty Noama Chomského, jenž ji poprvé představil.

Jak název naznačuje, typy gramatik tvoří hierarchii; gramatika vyššího typu je vždy současně i gramatikou nižšího typu. Typy gramatik jsou určeny omezeními na množinu pravidel:

Gramatika typu 0 neklade žádná omezení na množinu pravidel, libovolná gramatika je gramatikou typu 0.

Gramatika typu 1 neboli **kontextová gramatika** klade na všechna pravidla $\alpha \to \beta$ podmínku $|\alpha| \leq |\beta|$, tedy levá strana každého pravidla musí být kratší než jeho pravá strana. Výjimkou z tohoto omezení je pravidlo $S \to \epsilon$, které může být přítomno.

[7] zde L neoznačuje jazyk jako množinu, jedná se o konvenci pro zápis pojmu „jazyk generovaný gramatikou G"

Gramatika typu 2 neboli **bezkontextová gramatika** má všechna pravidla ve tvaru $A \to \beta$ (tak, že $A \in N$), tedy na levé straně je vždy právě jeden neterminál a β je neprázdné. Výjimkou z tohoto omezení je pravidlo $S \to \epsilon$, které může být přítomno.

Gramatika typu 3 neboli **regulární gramatika** má všechna pravidla ve tvaru $A \to aB$ nebo $A \to a$, kde A, B jsou neterminály a a je terminál. Výjimkou z tohoto omezení je pravidlo $S \to \epsilon$, které může být přítomno.

Nejčastěji se pracuje s regulárními a bezkontextovými gramatikami a jazyky. Tyto typy gramatik jsou efektivně zpracovatelné na počítačích a jako základ se používají i při zpracování textů v přirozeném jazyce.

7.4 Konečný automat

Automaty jsou jiným abstraktním modelem charakterizujícím jazyky. Fungují jako stavové systémy, které čtou po znacích slovo na vstupu, s přečtením každého znaku změní stav a na konci rozhodnou, zda slovo je akceptováno či nikoliv. Podobně jako říkáme, že gramatika vygeneruje slovo (resp. existuje odvození slova z gramatiky), řekneme, že **automat akceptuje slovo**. Zde si představíme nejjednodušší typ automatu, **konečný automat**.

Konečný automat je pětice $(Q, \Sigma, \delta, q_0 F)$, kde

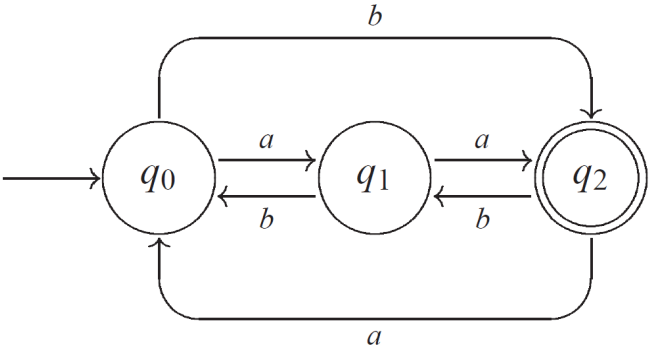
- Q je neprázdná konečná množina stavů automatu
- Σ je konečná množina vstupních symbolů (abeceda)
- $\delta : Q \times \Sigma \to Q$ je přechodová funkce automatu
- q_0 je počáteční stav
- F je množina koncových stavů

Běh automatu probíhá následovně. Začneme v počátečním stavu. Přečteme první symbol na vstupu a na základě přechodové funkce se přesuneme do dalšího stavu. Dále po jednom čteme další symboly na vstupu a podle přechodové funkce, aktuálního stavu a symbolu na vstupu se přesouváme do dalších stavů. Pokud se po dočtení posledního symbolu nacházíme v některém z množiny akceptujících stavů, automat slovo akceptuje. Pokud se nacházíme ve stavu, který do množiny F nepatří, automat slovo neakceptuje.

Konečný automat lze názorně zakreslit tzv. *přechodovým diagramem*, kde stavy znázorňujeme kroužky, přechodovou funkci šipkami mezi kroužky ohodnocenými symboly z abecedy, koncové stavy vyznačujeme dvojitým kroužkem

a počáteční stav šipkou „z prostoru". Takovýto diagram je dostatečnou specifikací konečného automatu, můžeme z něj odvodit hodnoty všech prvků pětice.

Příklad. Ukážeme si běh následujícího automatu na slově *abaa*:



Začínáme výpočet v počátečním stavu q_0. Přečteme první symbol vstupního slova, „a", a podle přechodové funkce znázorněné šipkou s příslušným symbolem přejdeme z počátečního stavu do stavu q_1. Dalším symbolem na vstupu je „b" a na základě přechodové funkce přejdeme ze stavu q_1 do stavu q_0. Dalším symbolem je „a", přejdeme tedy ze stavu q_0 do stavu q_1. Posledním symbolem je opět „a" a automat na základě přechodové funkce přejde ze stavu q_1 do stavu q_2. Jsme na konci vstupního slova a nacházíme se v akceptujícím stavu, automat tedy toto slovo akceptuje.

Pokud automat akceptuje právě všechna slova nějakého jazyka L (všechna slova daného jazyka a žádná jiná), řekneme, že **automat rozpoznává jazyk** L. Podobně jako v případě gramatiky značíme jazyk rozpoznávaný určitým automatem A jako $L(A)$.

7.5 Ekvivalence formalismů

Konečný automat je ekvivalentním formalismem k regulárním gramatikám, to znamená, že rozpoznávají stejné třídy jazyků. Jinými slovy, ke každé regulární gramatice existuje konečný automat, který rozpoznává jazyk generovaný touto gramatikou; a platí to i naopak, tedy ke každému konečnému automatu existuje regulární gramatika, která generuje jazyk rozpoznávaný tímto automatem. Důkaz tohoto faktu je konstruktivní, tj. přímo představuje převod gramatiky na automat a naopak. Nebudeme jej zde uvádět, zkuste se ale zamyslet nad tím, jak by tento převod mohl vypadat.

Existují i další typy automatů; některé z nich jsou ekvivalentní s jinými typy gramatik. Pravděpodobně ještě uslyšíte např. pojmy *zásobníkový automat* nebo *Turingův stroj*. Tyto formalismy jsou již nad rámec našeho výkladu. Více se o nich můžete dozvědět v kurzu *Formální jazyky a automaty* na FI.

8 Kombinatorika a pravděpodobnost

V dalším výkladu se budeme zabývat statistikou a pravděpodobností. K úvahám o pravděpodobnosti často potřebujeme vědět, kolik různých možností v různých případech může nastat. Tímto počítáním se zabývá kombinatorika; a kromě zmíněné motivace je rovněž jedním z klíčů k pochopení matematického uvažování.

Ze střední školy si možná pamatujete některé vzorečky pro variace, permutace a kombinace, s opakováním nebo bez. Tyto vzorečky si tady připomínat nebudeme, vlastně by bylo lepší je zapomenout. Často je obtížné rozhodnout, který vzoreček použít pro kterou situaci, navíc tyto vzorečky často nejsou jednoduché a navádí nás spíše k jejich mechanickému používání než ke skutečnému přemýšlení o důvodech, které k výsledkům vedou.

8.1 Základní kombinatorická pravidla

Místo vzorečků si představíme dvě jednoduchá pravidla, které je relativně snadné „napasovat" na konkrétní případy, a konkrétní výpočty budeme řešit spíše úvahou, která bude zase na druhou stranu komplikovanější než v případě použití vzorečků. Dobrý způsob, jak řešit komplikovanější případy, je vyzkoušet si, jak věci fungují v jednoduchých případech, kdy např. můžeme vypsat všechny možnosti, a na základě těchto jednoduchých případů zobecňovat.

Prvním ze zmiňovaných pravidel je tzv. **pravidlo součtu**, které říká, že pro disjunktní množiny $A_1, A_2, ..., A_n$ o velikostech $p_1, p_2, ..., p_n$ má množina $A_1 \cup A_2 \cup ... \cup A_n$ velikost $p_1 + p_2 + ... + p_n$.

Druhým je **pravidlo součinu** říkající, že počet všech uspořádaných k-tic, takových, že 1. člen lze vybrat n_1 způsoby, druhý člen n_2 způsoby, ..., k-tý člen n_k způsoby, je $n_1 * n_2 * ... * n_k$.

Tato téměř triviální pravidla jsou vše, co potřebujeme ke kombinatorickým úvahám. Ukážeme si jedno použití na příkladě.

Příklad. Kolika způsoby lze seřadit množinu {1, 2, ..., n}? První prvek vybíráme z n prvků, druhý z $n-1$ prvků atd. Podle pravidla součinu je počet všech seřazení $n * (n-1) * (n-2) * ... = n!$.

8.2 Pravděpodobnost

Pravděpodobnost nějakého jevu je definována jako podíl m/n, kde m je počet možností, kdy daný jev nastal, a n je počet všech možností, které potenciálně nastat mohly či mohou. Tato čísla někdy vyvozujeme na základě kombinatorických úvah a jistých předpokladů (např. předpokládáme, že kostka,

kterou házeme, je vyvážená), často je však také odvozujeme na základě statistických dat, jak brzy poznáme v dalším výkladu.

9 Základy teorie grafů

Ještě než se však začneme podrobněji zabývat statistikou, ukážeme si ještě něco z jiného důležitého podoboru diskrétní matematiky, a tím je teorie grafů. Jak brzy poznáme, nejedná se zde o grafy funkcí, i když s nimi lze také pracovat v podobě obrázků. Můžeme si je představit jako skupiny bodů (které budeme nazývat vrcholy nebo uzly, v praxi představují např. města nebo počítače v síti) propojené cestami (které budeme nazývat hrany, v praxi např. silnice nebo síťová spojení).

V praxi představují grafy užitečnou teoretickou základnu pro velké množství aplikací – s jejich pomocí jsou modelovány např. mapy (vč. vyhledání nejrychlejších cest), počítačové či elektrické sítě, v lingvistické oblasti je pak využíváme k modelování syntaxe (syntaktické stromy) či sémantiky (sémantické sítě).

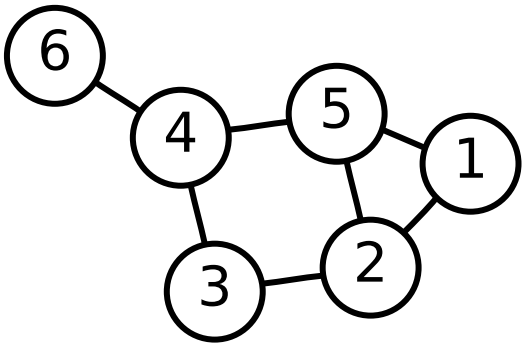
9.1 Základní pojmy

Graf G je definován jako uspořádaná dvojice (V, E), kde V je množina vrcholů grafu (někdy značíme i $(G(V))$ a E, značená někdy též $G(E)$, je množina hran. Každá hrana je dvouprvkovou podmnožinou množiny vrcholů (tedy množina E se skládá z dvouprvkových množin) a její sémantiku si můžeme představit tak, že hrana „spojuje" vrcholy, které jsou v ní obsaženy.

Příkladem grafu může být dvojice (V, E), kde

$$V = \{1, 2, 3, 4, 5, 6\}$$
$$H = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}$$

Takový graf bychom pak zakreslili např. takto:



Nyní si zavedeme několik základních pojmů nad grafy, které nám pak usnadní formulaci dalších definic a faktů.

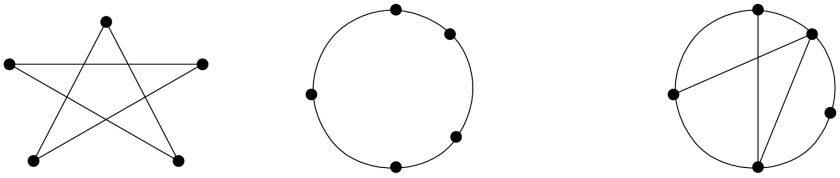
Sousední vrcholy. Řekneme, že dva vrcholy jsou sousední, pokud mezi nimi vede hrana (jinými slovy, $u, v \in V$ jsou sousední, pokud $\{u, v\} \in E$).

Stupeň vrcholu je počet hran, které z daného vrcholu vychází (neboli stupeň vrcholu x je $|\{\{u, v\} \in E \mid u = x \wedge v = x\}|$).

Podgraf grafu $G = (V, E)$ je graf $G' = (V', E')$ takový, že $V' \subseteq V$ a $E' \subseteq E$. Je důležité, aby i G' byl graf, tedy aby platilo $\forall\{u, v\} \in E'$ $(u \in V' \wedge v \in V')$. Jinými slovy, podgraf tvoří vybrané vrcholy a hrany původního grafu.

Izomorfismus mezi grafy G a G' je bijekce $f : V(G) \to V(G')$, kde platí $\forall\{u, v\} \in G(E)$ $(\{f(u), f(v)\} \in G(E'))$. Izomorfismus je způsob, jak říct, že grafy jsou shodné, až na pojmenování vrcholů, čili dokážeme přejmenovat vrcholy tak, abychom z jednoho grafu dostali druhý. Řekneme, že grafy jsou **izomorfní** (shodné), pokud mezi nimi existuje izomorfismus (tedy jsme schopni nalézt bijekci f, která splňuje kritérium izomorfismu).

Příklad. Které z následujících grafů jsou izomorfní?



9.2 Typy grafů

Rozeznáváme několik základních typů grafů:

- **Kružnice** nebo **cyklus** délky n je graf (případně libovolný s ním izomorfní) $(V = \{1, 2, ..., n\}, E = \{\{1, 2\}, \{2, 3\}, ..., \{n-1, n\}, \{n, 1\}\})$. Má stejný počet vrcholů a hran, všechny vrcholy jsou stupně 2 a nákres grafu tvoří kružnici.

- **Cesta** délky n je kružnice s jednou chybějící hranou, tedy $(V = \{1, 2, ..., n\}, E = \{\{1, 2\}, \{2, 3\}, ..., \{n-1, n\}\})$.

- **Úplný graf** je takový, kde mezi každými dvěma vrcholy vede hrana, tedy $(V = \{1, 2, ..., n\}, E = \{\{u, v\} \mid u \in V \ \wedge \ v \in V \ \wedge \ u \neq v\})$.

Na základě těchto typů rozeznáváme též specifické podgrafy. Cyklus v grafu je jeho podgraf, který je kružnicí; cesta v grafu je podgraf, který je cestou; **klika** v grafu je podgraf, který je úplným grafem.

Mezi další typy grafů patří:

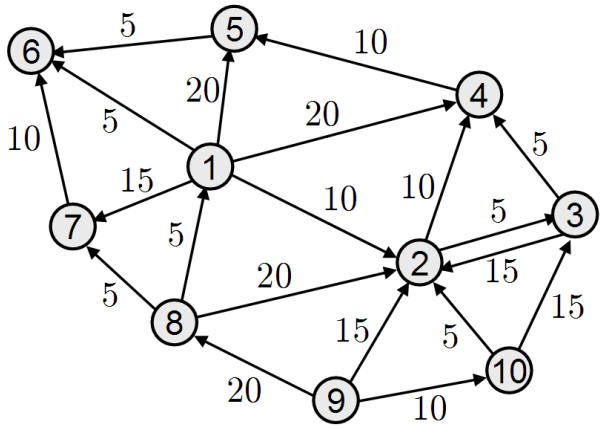
- **Souvislý graf** je takový graf ve kterém existuje cesta mezi každými dvěma vrcholy. (Opakem je graf **nesouvislý**)
- **Acyklický graf** je takový, který neobsahuje cyklus. Někdy ho nazýváme též **les**, protože jeho souvislé části jsou stromy.
- **Strom** je acyklický souvislý graf. (Dochází zde k menšímu paradoxu, neboť lze říct, že souvislý les je strom :-))

9.3 Rozšíření pojmu graf

Ve většině aplikací není jednoduchý graf, jak jsme si ho představili, dostatečným modelem, vznikly proto mnohé matematické objekty, které základní pojem grafu rozšiřují. Některé z nich si zde představíme.

- **Orientovaný graf.** V takovémto grafu jsou hrany orientovány. U každé hrany tedy rozeznáváme zdrojový a cílový vrchol; tato vlastnost nám např. umožňuje modelovat jednosměrné ulice v mapě. Formální definici musíme změnit tak, že množina hran již nebude množinou (neuspořádaných) dvouprvkových podmnožin, ale uspořádaných dvojic, kde je jednoznačně dán první (zdrojový vrchol) a druhý (cílový vrchol) prvek.
- **Ohodnocený graf.** Takovýto graf přiřazuje každé hraně reálné číslo, které nazýváme *ohodnocení*. Tato vlastnost nám umožní modelovat např. různé vzdálenosti mezi místy v mapě nebo různé kapacity síťových kanálů. Formálně k definici grafu přidáme ohodnocovací funkci $e : E(G) \to \mathbb{R}$.
- **Multigraf** povoluje více hran mezi stejnou dvojicí vrcholů, čímž umožňuje např. modelovat více různých cest mezi dvěma městy, servery apod. Rovněž tento formalismus povoluje hrany začínající i končící ve stejném vrcholu, tzv. „smyčky". Možnosti formální reprezentace takovéto struktury jsou různé, např. ke každé hraně (dvouprvkové množině) můžeme přidat třetí prvek, index, který odliší dvě hrany mezi stejnou dvojicí vrcholů. Přidaný prvek pak nesmí být obsažen v množině vrcholů (Proč?).

Všechna výše uvedená rozšíření se mohou vzájemně kombinovat, můžeme (a často se s tím i setkáváme) mít např. orientovaný ohodnocený graf, který lze znázornit jako následující obrázek:



nebo např. orientovaný ohodnocený multigraf, který je základem pro přechodový graf konečného automatu, s nímž jsme se již setkali (zde navíc máme ještě různé typy vrcholů).

9.4 Graf jako relace

To, že množina hran v orientovaném grafu je množina uspořádaných dvojic, nám může napovědět, že na graf můžeme pohlížet (a můžeme ho tak i formálně reprezentovat) jako na binární relaci na množině vrcholů. Tím pádem ho např. můžeme reprezentovat jako tabulku příslušné relace, tzv. **matici sousednosti**.

Neorientovaný graf lze v tomto formátu zapsat jako symetrickou relaci (ve významu, že mezi každou dvojicí vrcholů vede hrana tam i zpět, což lze považovat za totéž, jako bychom orientaci hran neuvažovali), matice sousednosti by tedy v tomto případě byla symetrická.

Následující tabulka je příkladem matice sousednosti pro graf ze začátku této kapitoly.

vrchol	**1**	**2**	**3**	**4**	**5**	**6**
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

9.5 Další pojmy z teorie grafů

Dále si představíme několik dalších pojmů z teorie grafů, které pravděpodobně ještě někdy uslyšíte.

Souvislé komponenty v grafu jsou jeho největší souvislé podgrafy – na obrázku je poznáme snadno, jde o souvislé „ostrůvky" které vzájemně nejsou spojeny žádnou hranou. Jinými slovy lze také říct, že se jedná o největší podgrafy takové že mezi každými dvěma vrcholy každého takového podgrafu vede cesta.

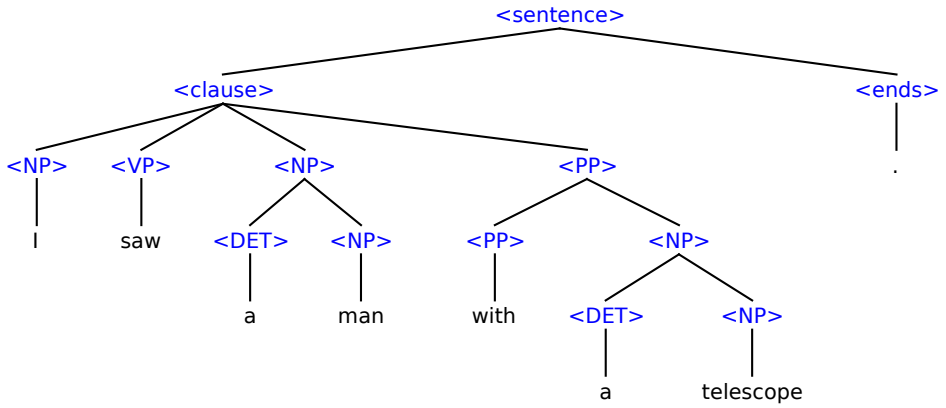
V případě orientovaných grafů mluvíme o tzv. **silně souvislých komponentách**, kde požadujeme, aby v každé komponentě (podgrafu) vedla cesta mezi každými dvěma vrcholy tam i zpět (z každého vrcholu se „po šipkách" dostaneme do každého jiného). Analogicky můžeme mluvit o slabě souvislých komponentách, kde požadujeme, aby existovala cesta alespoň jedním směrem. Jako cvičení si můžete zkusit najít silně a slabě souvislé komponenty v grafu ze sekce 9.3

Délkou cesty v grafu rozumíme počet hran v této cestě. V případě ohodnoceného grafu délkou cesty obvykle rozumíme součet ohodnocení hran této cesty. Délku nejkratší cesty mezi dvěma vrcholy pak označujeme jako **vzdálenost mezi vrcholy**.

V případě stromů často z praktických důvodů rozeznáváme více typů vrcholů. Obvykle strom obsahuje jeden význačný vrchol, ve kterém „začíná" – tomuto vrcholu říkáme **kořen**. Volba kořene je obvykle dána aplikací, tj. není možné určit kořen u obecného stromu – nejedná se o vlastnost grafu, ale o volbu danou konvencemi. Pokud informatici kreslí strom, kreslí jeho kořen většinou nahoře.

Dále, všechny vrcholy stupně 1, které nejsou kořenem, obvykle nazýváme **listy stromu**. Listy naopak kreslíme většinou dole.

Příkladem lingvistického využití teorie grafů je **syntaktický strom**, kterým znázorňujeme syntaktické odvození věty (se syntaktickými stromy se také jistě ještě setkáte). Příkladem je strom na následujícím obrázku – zde je kořenem vrchol *<sentence>*, listy jsou pak slova dané věty.



Kostra grafu je podgraf, který obsahuje všechny vrcholy původního grafu a zároveň je stromem. K tomu, abychom z grafu dostali jeho kostru, musíme tedy odstranit všechny cykly v tomto grafu, neboli odebrat hranu z každého jeho cyklu.

Pro ohodnocený graf dále můžeme mluvit o **minimální, resp. maximální kostře**, což je taková kostra, která má minimální, resp. maximální možný součet ohodnocení hran, které jsou v ní obsaženy. Pojem maximální kostry má opodstatnění např. při hledání páteřních rozvodů v sítích nebo při hledání nejpravděpodobnější syntaktické analýzy v závislostním formalismu.

9.6 Grafové algoritmy

Protože grafy jsou v drtivé většině určeny ke zpracování počítači a protože počítače nevidí graf jako obrázek, ale mají k dispozici pouze jeho číselnou reprezentaci s pomocí množin, polí, případně reprezentaci maticí sousednosti, uvedeme si zde několik příkladů ilustrujících způsob, jakým počítače s grafy pracují.

Nejjednoduššími z grafových algoritmů jsou algoritmy procházení (též prohledávání) grafu – slouží obvykle jako prostředek k navštívení všech vrcholů a k provedení nějaké akce nad každým z nich. My zde budeme pro ilustraci používat akci „označ".

Rozeznáváme dva základní způsoby procházení grafu – do hloubky a do šířky. Liší se zejména v pořadí, v jakém navštěvujeme jednotlivé vrcholy. Vždy začínáme z jednoho určeného vrcholu a po hranách vedoucích z tohoto vrcholu postupujeme dále v grafu.

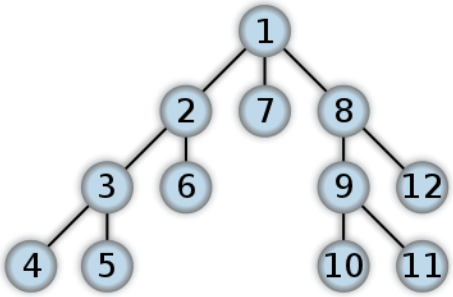
Procházení do hloubky (depth-first search) z určeného vrcholu probíhá tak, že označíme tento vrchol, poté se přesuneme na libovolný sousední neoznačený vrchol a provedeme znovu celý algoritmus s novým počátečním vrcholem (tedy opět jej označíme a přesuneme se dále na libovolný sousední

neoznačený vrchol). Pokud neexistuje žádný sousední neoznačený vrchol, vracíme se postupně na vrcholy, které jsme již navštívili, v opačném pořadí a zkoušíme, zda můžeme pokračovat dále z těchto vrcholů.

Následuje formálnější zápis (tzv. pseudokód) nastíněného algoritmu. Vidíme, že algoritmus je rekurzivní, tedy volá sebe sama.

```
DFS (G, u)
============
označ
u for všechny hrany (u, v) vycházející z vrcholu u:
    if v není označen:
        DFS (G, v)
```

Pořadí, v jakém jsou jednotlivé vrcholy grafu navštíveny, pak ilustruje následující obrázek (začínáme ve vrcholu 1 a pro přehlednost v případě více hran vedoucích z daného vrcholu jdeme vždy napřed doleva).

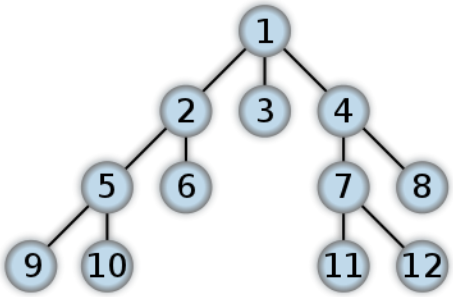


Procházení do šířky (breadth-first search) z určeného vrcholu začíná opět označením tohoto vrcholu. Poté všechny neoznačené sousední vrcholy přidáme do seznamu (tzv. fronty). Dokud seznam není prázdný, vybereme vždy první vrchol seznamu, smažeme jej ze seznamu, označíme jej a všechny neoznačené sousední vrcholy přidáme na konec seznamu. Procházení končí, když je seznam prázdný.

Pseudokód procházení do šířky:

```
BFS (G, u)
============
Q = [u]
while Q je neprázdný:
    odstraň první prvek z Q a přiřaď jej do t
    označ t
    přidej všechny neoznačené sousedy t na konec Q
```

A ilustrace pořadí navštívených vrcholů (srovnejte s pořadím při procházení do hloubky):



Dalším algoritmem, který stojí za zmínku, je **Kruskalův algoritmus** pro hledání minimální kostry. Vstupem algoritmu je ohodnocený graf, výstupem je jeho minimální kostra. Algoritmus nejprve setřídí všechny hrany vzestupně podle ohodnocení, poté je postupně podle tohoto pořadí zkouší přidat do výsledného grafu a v každém kroku kontroluje, zda nevznikl cyklus. To, že nevznikne cyklus, algoritmus zajistí tím, že si stále pamatuje a v každém kroku aktualizuje seznam souvislých komponent výsledného grafu a testuje, zda nově přidávaná hrana spojuje dvě různé souvislé komponenty (v opačném případě by vznikl cyklus a hrana nebude přidána).

Pseudokód Kruskalova algoritmu:

```
přiřaď K = []; přiřaď comp = {}
for u in G(V):
    přiřaď comp[u] = set(u)
setřiď G(E) podle ohodnocovací funkce
for (u, v) in G(E):
    if comp[u] != comp[v]:
        K.append((u,v))
        newset = union(comp[u], comp[v])
        for x in newset:
            přiřaď comp[x] = newset
(G(V), K) je minimální kostra grafu
```

Na internetu lze nalézt spoustu názorných animací Kruskalova algoritmu, pro lepší pochopení doporučujeme některé vyhledat, shlédnout a porovnat s pseudokódem algoritmu. Nejlepší způsob procvičení je zkusit si algoritmus napsat a spustit.

Posledním algoritmem, kterým se budeme zabývat, je **Dijkstrův algoritmus** pro nalezení nejkratší cesty mezi zadanou dvojicí vrcholů – fakticky

algoritmus ovšem počítá všechny vzdálenosti z počátečního vrcholu do všech ostatních vrcholů grafu (a je zajímavým faktem, že nikdo dosud nepřišel na žádný způsob, jak to udělat bez toho, tj. vždy musíme spočítat vzdálenosti mezi počátečním vrcholem a všemi zbývajícími vrcholy v grafu). Tento algoritmus, resp. jeho modifikace, používají např. služby na vyhledávání nejkratších či nejrychlejších cest, jako např. portál *mapy.cz*.

Vstupem algoritmu je graf ohodnocený nezápornými čísly a počáteční vrchol s, výstupem je pole vzdáleností z vrcholu s do všech dalších vrcholů grafu. Algoritmus si po celou dobu běhu udržuje aktuální nejmenší známé vzdálenosti do všech vrcholů (na začátku všechny tyto vzdálenosti nastaví na nekonečno), postupně prochází hrany, hledá kratší cesty a hodnoty aktualizuje.

Pseudokód Dijkstrova algoritmu:

```
for u in G(V):
    přiřaď d[u] = infinity
přiřaď d[s] = 0
přiřaď N = G(V)
přiřaď p = {}
while N != []:
    přiřaď u = vrchol z N s nejmenší hodnotou d[u]
    for všechny hrany (u,x) vycházející z vrcholu u:
        přiřaď alt = d[u] + w((u,x))
                                # kde w je ohodnocovací funkce
        if alt < d(x):
            přiřaď d[x] = alt
            přiřaď p[x] = u
odstraň vrchol u z množiny N
d jsou vzdálenosti vrcholů z vrcholu s
p obsahuje předchozí vrcholy na nejkratší cestě z vrcholu s
```

10 Základy popisné statistiky

Počínaje touto kapitolou se dostáváme na půdu statistiky, matematického odvětví, které se zabývá převážně zpracováním velkých datových souborů, a to zejména sumarizací obsažených informací, odhady informací o velkých souborech dat na základě menšího vzorku, vytvářením statistických modelů objektů (např. jazyka) na základě dostupných dat.

Při počítačovém zpracování přirozeného jazyka se používá velké množství metod založených na statistice, přičemž se často jedná o nejlepší výsledky, které byly v dané oblasti dosaženy. Proto je žádoucí, abyste se i vy zevrubně seznámili se základními pojmy popisné statistiky. Využijete je nejen v navazujících předmětech či při projektech z počítačového zpracování jazyka, ale i v běžném životě, neboť se statistickými údaji se setkáváme téměř každý den.

10.1 Statistický soubor

Objektem zkoumání matematické statistiky je **statistický soubor**, což chápeme jako posloupnost údajů (většinou číselných nebo výjádřených číselně) o nějakých objektech. Typy těchto údajů nazýváme **statistické znaky** a jejich počet pak určuje **rozměr statistického souboru**.

Příkladem statistického souboru může být množina údajů o výšce a hmotnosti slonů v Africe – v tomto případě výška a hmotnost jsou statistické znaky a rozměr souboru je 2.

Dále rozlišujeme tzv. **základní soubor** (též **populace**), který uvažuje všechny objekty daného typu, v našem příkladě by se jednalo o množinu všech slonů v Africe. O tomto souboru většinou chceme shromažďovat údaje a vyvozovat fakta.

Protože však většinou nejsme schopni změřit a zvážit všechny slony v Africe, obvykle pracujeme jen s omezeným výběrem objektů ze základního souboru – v našem příkladě by se jednalo např. o množinu slonů, které se podařilo změřit a zvážit. Teprve tomuto výběru pak říkáme **statistický soubor**.

Aby vybraný vzorek správně vypovídal o základním souboru, výběr by měl být reprezentativní. Pro nedostatek informací o základním souboru (cílem statistického zkoumání je obvykle právě tyto informace získat) se však většinou spoléháme na **náhodný výběr** a na to, že náhoda nám zaručí reprezentativnost vzorku. Je však třeba dávat pozor na to, jaké vlivy mohou náhodnost našeho výběru ohrozit a tím způsobem zkreslit výsledky zkoumání. Například jsme mohli měřit a vážit pouze příliš tlusté slony, protože pomaleji běhají a lépe se chytají.

10.2 Jednorozměrný statistický soubor

Často pracujeme se statistickým souborem omezeným pouze na jeden statistický znak (např. soubor informací o hmotnosti slonů). Pro lepší ilustraci uvažujme následující příklad: Podařilo se nám zvážit šest slonů, ti měli hmotnosti postupně 2, 4, 4, 4, 5 a 11 tun. Statistický soubor pak bude šestice $(2, 4, 4, 4, 5, 11)$.

Rozsah statistického souboru je počet jeho prvků, tedy rozsah našeho souboru je 6.

Absolutní četnost hodnoty (někdy též pouze četnost) v souboru je počet jejích výskytů, např. četnost hodnoty 4 v našem souboru je 3. Dále rozeznáváme **relativní četnost**, což je absolutní četnost podělená rozsahem souboru a udává se obvykle v procentech. Tedy relativní četnost hodnoty 4 je 50 %.

Kumulativní četnost hodnoty je četnost hodnoty souboru plus četnost všech menších hodnot. Rozeznáváme opět absolutní a relativní kumulativní četnost. Kumulativní absolutní četnost hodnoty 4 je 4, kumulativní relativní četnost hodnoty 4 je 66 %.

Často je možných hodnot příliš mnoho (např. při vážení na kilogramy nedostaneme třikrát přesně 4000 kg, ale budeme mít tři různé hodnoty, všechny s četností 1), a proto se může hodit rozdělit tyto hodnoty do nějakých tříd, které již budou obsahovat větší počet prvků. Např. můžeme rozeznávat třídu „3500 - 4500 kg", což je v podstatě ekvivalentní zaokrouhlení hodnot na celé tuny. Rozdělení hodnot do tříd budeme často dělat implicitně, i v následujícím textu – pokud budeme psát „3 lidé s výškou 170 centimetrů", bude to nejspíše znamenat, že uvažujeme přesnost na celé desítky nebo analogicky že uvažujeme třídu lidí od 165 do 175 centimetrů, která má četnost 3.

Četnosti můžeme zaznamenat do sloupcového grafu – graf funkce, který má na ose x jména tříd a na ose y jejich absolutní nebo relativní četnosti, se nazývá **histogram**.

10.2.1 Charakteristiky polohy

Pro jednorozměrný statistický soubor zavádíme tzv. charakteristiky polohy a charakteristiky variability. S většinou z nich jste se jistě už setkali. Charakteristiky polohy nám shrnují potenciálně velké množství dat do několika málo čísel, které lze snadno interpretovat a vytvořit si tak hrubý úsudek o celém vzorku dat (např. jak vysoký je běžný slon). Charakteristiky variability nám ukazují, jak je statistický soubor vnitřně konzistentní, čili jak moc se od sebe vzájemně liší hodnoty obsažené v souboru.

Mezi důležité charakteristiky polohy patří:

- **Modus**, což je hodnota či třída s největší četností (v našem ukázkovém souboru je modus 4).
- **Aritmetický průměr** (někdy též značený *avg* jako „average") je součet hodnot ve statistickém souboru, podělený velikostí souboru. Pokud bychom uvažovali sloupce v histogramu jako závaží, pak průměr je těžištěm histogramu. Pro náš soubor je průměr 5.
- **Medián** je „prostřední" hodnota v souboru po jeho setřídění. V případě, že datový soubor má sudý počet prvků, je to průměr ze dvou prostředních, pro náš ukázkový soubor je medián 4. Medián není citlivý na extrémní odchylky jako průměr – např. relativně málo hodně vysokých hodnot může průměr vychýlit i hodně nahoru, zatímco u mediánu se neprojeví vůbec. (Zamyslete se, proč se zveřejňuje celorepublikový průměr platů a ne celorepublikový medián platů.)

10.2.2 Charakteristiky variability

Hlavní charakteristikou variability statistického souboru je **rozptyl** (též **disperze** nebo **variance**) značený s^2. Je definován jako

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - avg)^2 = ((x_1 - avg)^2 + (x_2 - avg)^2 + ... + (x_n - avg)^2)/n$$

Jedná se tedy o aritmetický průměr druhých mocnin odchylek jednotlivých hodnot od průměru. Pokud je rozptyl 0, všechny hodnoty v souboru jsou stejné. Čím větší je rozptyl, tím více se jednotlivé hodnoty od sebe liší.

Odvozenou charakteristikou variability je **směrodatná odchylka** s, která je pouze odmocninou z rozptylu a vyjadřuje téměř totéž, jen jiným číslem.

10.3 Dvourozměrný statistický soubor

Dvourozměrný statistický soubor (např. data o výšce a hmotnosti slonů) lze chápat jako dva jednorozměrné soubory, vzájemně provázané. Formálně jej můžeme reprezentovat jako posloupnost uspořádaných dvojic, např.

$$((x_1, y_1), (x_2, y_2), ..., (x_n, y_n))$$

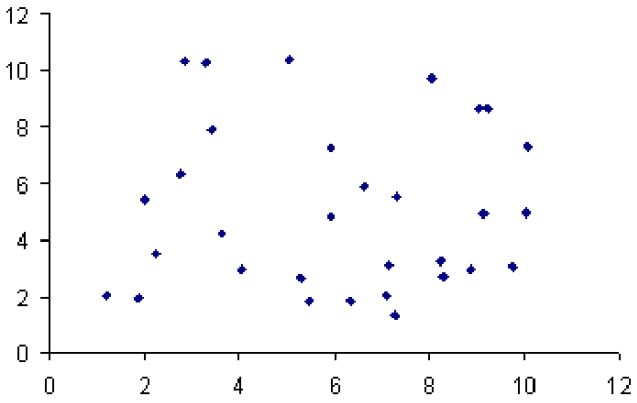
Důležitou vlastností dvourozměrného statistického souboru je **korelace** statistických znaků. Pojmem korelace rozumíme **stupeň lineární závislosti** znaků x a y, tedy to, do jaké míry hodnoty znaku x lineárně závisí na hodnotách znaku y. Jinými slovy, to, jak dobře lze grafem závislosti x na y proložit přímku. Vzorec pro výpočet korelace je

$$\frac{\sum_{i=1}^{n}(x_i - avg(x))(y_i - avg(y))}{n * s(x) * s(y)}$$

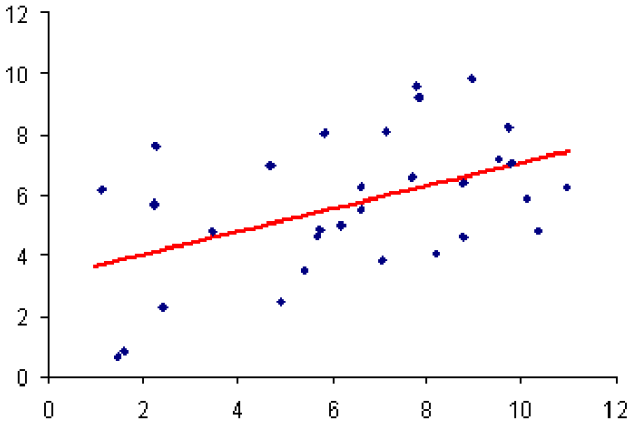
kde $s(x), s(y)$ jsou směrodatné odchylky jednorozměrných datových souborů pro znaky x a y.

Hodnoty korelace se pohybují od -1 do 1. Pokud je korelace 0, jsou hodnoty znaků dokonale nezávislé; pokud je korelace 1, jedná se o přímou úměrnost (čím větší je x, tím větší je y a hodnoty y lze z hodnot x získat jednoduše vynásobením nějakou kladnou konstantou); pokud je korelace -1, jedná se o nepřímou úměrnost (čím větší je x, tím menší je y a hodnoty y lze z hodnot x získat jednoduše vynásobením nějakou zápornou konstantou).

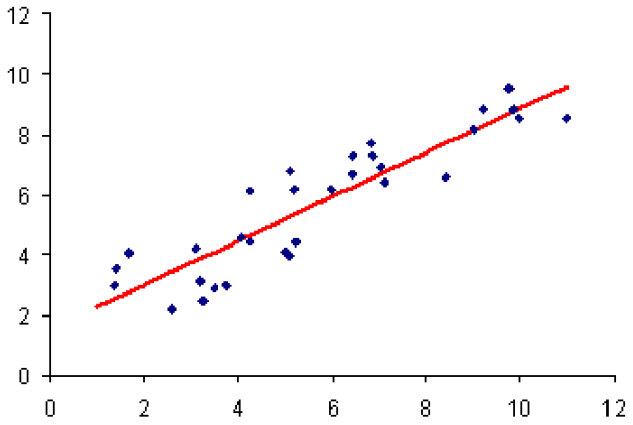
Pro lepší představu uvádíme ukázky grafů závislosti dvou statistických jevů pro různé hodnoty korelace:



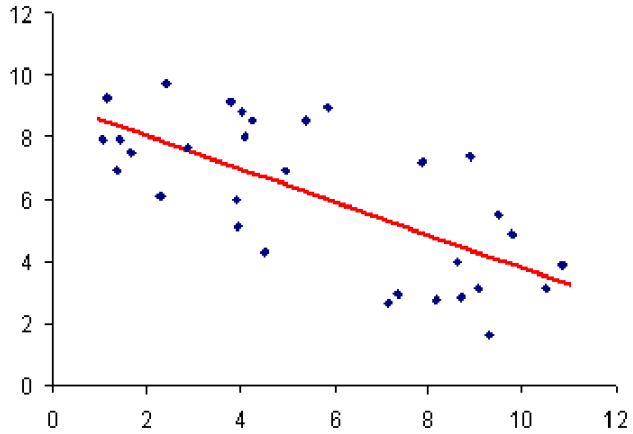
korelace 0



korelace 0.5



korelace 0.9



korelace -0.7

11 Statistika a pravděpodobnost

V kapitole 8 jsme si představili základní pojem pravděpodobnosti, v minulé kapitole jsme si představili základní pojmy popisné statistiky. V této kapitole si vysvětlíme, jak spolu pravděpodobnost a statistika souvisí. K tomu ale potřebujeme zavést několik dalších pojmů z teorie pravděpodobnosti.

11.1 Pravděpodobnostní rozložení

Náhodná proměnná, např. A je vlastnost, jejíž hodnotu z nějakého důvodu neznáme, např. protože o ní nemáme dost informací nebo protože vlastnost dosud žádné hodnoty nenabyla. Příkladem náhodné proměnné může být výsledek hodu kostkou, která je ukrytá pod kelímkem nebo třeba teplota vzduchu v Brně zítra v poledne. (Kdybychom měli dostatek informací a dokázali přesně modelovat počasí nebo hod kostkou, byli bychom možná schopni hodnoty příslušné vlastnosti *vypočítat*, toho však schopni nejsme, a proto zavádíme pojem náhodné veličiny.)

Většinou ale máme *nějaké* informace o dané vlastnosti, které nám mohou např. vyloučit nebo téměř vyloučit některé hodnoty. Např. víme, že na kostce nejspíš nepadne osmička, že zítřejší teplota patrně nebude větší než 50 stupňů apod. Minulá pozorování nám mohou napovědět ještě více. Např. pokud jsme už viděli deset hodů toutéž kostkou a z toho devětkrát padla šestka, s kostkou patrně nebude něco v pořádku a pokud bychom si měli vsadit na výsledek dalšího hodu, nejspíše bychom volili opět šestku – řekli bychom, že možnost, že padne šestka, je „pravděpodobnější" než možnost, že padne jednička, i když jistotu nemáme. Podobně ze zkušenosti víme, že pokud je zrovna červen, polední teplota v Brně se bude pohybovat nejspíše mezi 20 a 40 stupni a pravděpodobnost, že bude stupňů méně nebo více, je „menší".

Intuitivně tedy cítíme, že pravděpodobnosti jednotlivých možných hodnot náhodné veličiny nemusí být stejné. Jak můžeme toto vágní pozorování formalizovat, zpřesnit a využít např. pro přesnější předpovědi?

Pravděpodobnostní rozdělení, **pravděpodobnostní rozložení** nebo **pravděpodobnostní distribuce** jevu či vlastnosti A, anglicky **probability distribution**, je funkce, která pro jednotlivé možné hodnoty vrací pravděpodobnost, s jakou vlastnost A nabude této hodnoty. Formálně se jedná o funkci

$$p : X \to [0, 1]$$

kde X je množina možných hodnot příslušné vlastnosti a $[0, 1]$ je uzavřený interval od nuly do jedné, čili

$$\forall x \in X(p(x) \leq 1 \wedge p(x) \geq 0)$$

Dále musí platit, že součet hodnot funkce pro všechny možné hodnoty je 1, tedy

$$\sum_{x \in X} p(x) = 1$$

Rovněž si uveďme formálně, jaká je sémantika pravděpodobnostního rozložení:

$$p(x) = P(A = x)$$

Tedy hodnota pravděpodobnostního rozložení (malé p) je rovna pravděpodobnosti (velké P, tedy obecná pravděpodobnost), s jakou vlastnost A nabude hodnoty x. Dvojice (X, p), tedy množina všech možných hodnot vlastnosti spolu s pravděpodobnostním rozložením, se nazývá **pravděpodobnostní prostor**.

Fukncí, které splňují podmínky pravděpodobnostního rozložení, je samozřejmě celá řada. Dají se rozlišit na diskrétní (množina možných hodnot dané vlastnosti je konečná nebo spočetná) a spojité (množina možných hodnot je nespočetná) – my se budeme zabývat výhradně diskrétními pravděpodobnostními rozloženími.

Z čeho ale odvodíme vhodné pravděpodobnostní rozložení, které bude dobře charakterizovat naši náhodnou veličinu a bude prakticky použitelné?

11.2 Určení pravděpodobnostního rozložení

Možnosti jsou v zásadě dvě. První z nich je použití nějaké „ideální" funkce, které vychází z našich předpokladů o dané vlastnosti. Např. předpokládáme, že kostka, kterou se háže, je férová; hodnoty 1 až 6 tedy musí mít všechny stejnou pravděpodobnost, ostatní hodnoty budou mít pravděpodobnost 0. Z omezení kladená na pravděpodobnostní rozložení pak dostaneme, že pravděpodobnost pro hodnoty 1 až 6 bude 1/6. Nebo předpokládáme, že rozložení výšky lidí odpovídá nějakému ideálnímu *normálnímu* (viz dále) pravděpodobnostnímu rozložení s určitým průměrem a rozptylem, a rozhodneme se aproximovat výšku lidí tímto pravděpodobnostním rozložením.

Druhou možností je pravděpodobnostní rozložení určovat na základě měření provedeného v minulosti, které bylo zachyceno *ve statistickém souboru*. Tímto způsobem určujeme pravděpodobnost neznámých dat na základě jiných dat stejného typu, která jsme viděli a zaznamenali, např. určíme pravděpodobnostní rozložení teploty zítra v poledne na základě záznamů o měření teploty

v minulých dnech a letech nebo můžeme určit pravděpodobnostní rozložení slov (dvojic slov, slovních druhů apod.) v jazyce na základě dostatečně velkého vzorku textů v daném jazyce.

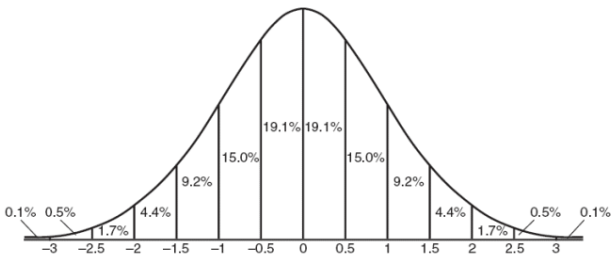
Možnosti určení pravděpodobnostního rozložení na základě statistického souboru jsou různé a obecně platí, že různé metody se hodí pro různé aplikace. Základní metodou nicméně je odvozování pravděpodobnostního rozložení na základě četností pozorovaných ve statistickém souboru. Jinými slovy, **relativní četnosti ve statistickém souboru odpovídají hodnotám pravděpodobnostního rozložení v pravděpodobnostním prostoru.**

11.3 Typy pravděpodobnostních rozložení

Na základě tvaru grafu rozlišujeme několik základních typů pravděpodobnostních rozložení. Představíme si zde tři nejdůležitější.

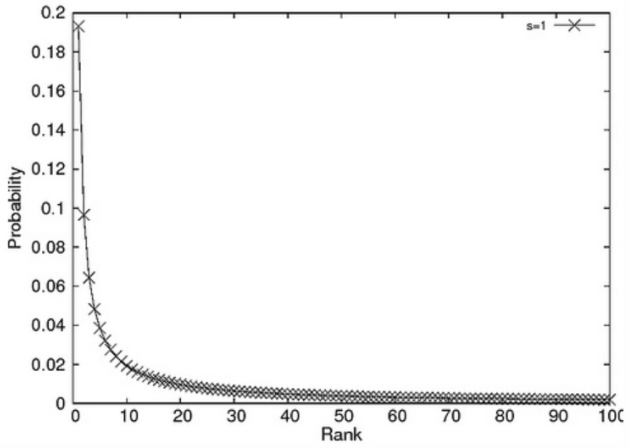
Uniformní rozložení je takové, v němž všechny hodnoty mají přibližně stejnou pravděpodobnost. Grafem jsou tedy body uspořádané přibližně do přímky vodorovné s osou x. Příkladem může být pravděpodobnostní rozložení výsledků hodu vyváženou kostkou.

Normální rozložení se vyznačuje tím, že nejpravděpodobnější hodnoty jsou blízké průměru a s větší odchylkou od průměru pravděpodobnost klesá. Graf takového rozložení má tvar zvonu, např.:



„Normální" se jmenuje proto, že dobře vystihuje velké množství vlastností různých přírodních populací – např. rozložení výšek a hmotností zvířat či lidí.

Zipfovo rozložení je typické tím, že několik málo nejčastějších hodnot má velkou pravděpodobnost a s každou další hodnotou (při setřídění od nejčastější) tato pravděpodobnost prudce klesá – v ideálním Zipfově rozložení má nejčastější hodnota pravděpodobnost x, druhá $x/2$, třetí $x/3$ atd. Graf takového rozložení nejčastěji kreslíme tak, že hodnoty na ose x jsou uspořádány podle jejich četnosti – graf potom tvoří prudce klesající křivku, jako je např. tato:



Toto rozložení velmi často výstižně popisuje nejrůznější frekvenční seznamy – např. pokud vytvoříme pravděpodobnostní rozložení velikostí světových měst, dojdeme ke křivce Zipfova rozložení. Stejnou křivku dostaneme, pokud vytvoříme pravděpodobnostní rozložení slov nebo jejich morfologických kategorií v jazyce.

Pro ilustraci uveďme frekvenční seznam morfologických značek pro angličtinu (zdrojem byl americký korpus Brown):

tag	Freq
NN	161881
NP	62669
NNS	56629
VVN	27545
VV	27481
VVD	27391
VVG	16922
VBD	13275
VBZ	11321
VVZ	8254
VVP	7912
VB	6377
VBP	5211
VHD	5190
VHZ	2497
VBN	2470
VHP	2445
VH	1780
NPS	1524
VBG	674
VHG	279
VHN	194

11.3.1 Zipfův zákon

Skutečnost, že Zipfovo rozložení výstižně popisuje opravdu velké množství jevů, se někdy označuje jako „Zipfův zákon". Zejména v přirozeném jazyce tento zákon platí téměř všude: téměř vždy je frekvence (nebo ekvivalentně – pravděpodobnost výskytu) zhruba nepřímo úměrná pořadí podle této frekvence; to platí pro slova, dvojice slov, slovní druhy, syntaktické vztahy, sémantické kategorie a mnohá další. Při jakémkoli zpracování jazyka je tedy třeba s ním počítat.

Pro ilustraci uveďme frekvence nejčastějších slovních tvarů v angličtině: „the" má relativní četnost 7 %, druhé „of" má 3,5 % a více než polovina anglických korpusů je pokryta 135 nejčastějšími slovy (u těchto slov se můžete setkat s technickým označením *stoplist*).

Zipfův zákon platí dokonce i pro „náhodně generovaný" jazyk – pokud budeme náhodně generovat znaky včetně mezery, pravděpodobnostní rozložení takto vygenerovaných „slov" bude odpovídat Zipfovu rozložení.

Zipfův zákon platí ale i v mnoha oblastech mimo přirozený jazyk – již jsme zmínili např. počet obyvatel měst; dále jej lze aplikovat např. na velikost platů, počet zaměstnanců společností a mnohá další. Rovněž známé ekonomické pravidlo „80 : 20" – 80 % problému vyřešíme s 20% úsilím – je vlastně alternativní formulací Zipfova zákona.

11.4 Distribuční funkce

Jak jsme si řekli, pravděpodobnostní rozložení je pravděpodobnost, že náhodná veličina nabude určité hodnoty (resp. zda patří do dané třídy), čili $p(x) = P(A = x)$, a její hodnoty odpovídají relativním četnostem ve statistickém souboru.

Distribuční funkce (cumulative distribution function), často značena F, je pravděpodobnost, že náhodná veličina nabude určité hodnoty nebo menší, čili $F(x) = P(A \leq x)$. Její hodnoty odpovídají kumulativním relativním četnostem ve statistickém souboru. Hodnoty distribuční funkce jsou také dobře známé jako tzv. **percentil**. Hodnota distribuční funkce (percentil) mediánu statistického souboru je 0,5.

11.5 Náhodný vektor

Náhodný vektor chápeme jako posloupnost náhodných veličin, např. náhodný vektor „počasí v Brně zítra v poledne" můžeme chápat jako trojici $(teplota, tlak, vlhkost)$. Jeho pravděpodobnostní rozložení můžeme modelovat s využitím vícerozměrného statistického souboru.

Pozor na to, že pravděpodobnost jednoho určitého vektoru hodnot může být obecně jiná než součin pravděpodobností jednotlivých jeho složek, což by se mohlo na první pohled (a po absolvování středoškolské pravděpodobnosti) zdát. Aby se jednalo o součin, musely by být jednotlivé složky vektoru dokonale nezávislé (viz dále), což v praxi téměř nikdy nenastává (i když to pro jednoduchost často předpokládáme). Pro dvourozměrný náhodný vektor (A, B) je hodnota pravděpodobnostního rozložení

$$p(x, y) = P(A = x \ \wedge \ B = y)$$

(srov. s pravděpodobnostním rozložením pro jednoduchou náhodnou veličinu).

Lze definovat i distribuční funkci pro náhodný vektor, např. pro dvourozměrný náhodný vektor je distribuční funkce analogicky definována jako

$$F(x, y) = P(A \leq x \quad \wedge \quad B \leq y).$$

12 Podmíněná pravděpodobnost

Po předvedení, jak souvisí pravděpodobnost se statistikou, se dále ve výkladu budeme zabývat některými dalšími, většinou prakticky motivovanými pojmy z této oblasti. Také si ukážeme některé praktické aplikace teorie pravděpodobnosti na praktické, i lingvistické, problémy.

Podmíněná pravděpodobnost je motivována potřebou formalizovat to, že často máme kromě pravděpodobnostního rozložení daného jevu další informace, např. o jiném jevu, který s původním může, ale nemusí souviset. Např. pokud budeme počítat pravděpodobnostní rozložení součtu dvou hodů kostkou až poté, co budeme znát výsledky prvního hodu, bude výsledné pravděpodobnostní rozložení jiné, než pokud budeme počítat toto rozložení bez jakýchkoli znalostí (zkuste si např. vypočíst pravděpodobnost. že padne součet 8 a pak pravděpodobnost, že padne součet 8 za předpokladu, že výsledek prvního hodu byl 1).

Jak jsme řekli, dva jevy, které zde bereme v potaz, spolu obecně mohou, ale nemusí, kauzálně souviset. Můžeme např. počítat pravděpodobnost deště zítra v poledne za předpokladu deště dnes v poledne (kde nejspíš nějaká souvislost existuje), ale i např. pravděpodobnost, že člověk je bezdomovec, pokud má vousy delší než 5 cm (zde bychom zřejmě také vypozorovali „souvislost", nicméně nemůžeme vyvodit nic takového jako „dlouhé vousy způsobují bezdomovectví" apod.). Pozor na fakt, že v běžném životě se změna pravděpodobnostního rozložení při určité podmínce často jako kauzální souvislost prezentuje. Většina zpráv typu „vědci zjistili, že ... (např. konzumace červeného masa způsobuje rakovinu)" přitom obsahuje pouze informaci o tom, že někdo zjistil, že mezi těmito dvěma jevy existuje korelace nebo že pravděpodobnostní rozložení druhého jevu se nějakým způsobem změní, pokud své zkoumání omezíme jen na objekty, pro které nastává první jev. To však nic nenapovídá o kauzálních vztazích těchto dvou jevů – souvislost může být náhodná nebo mít komplikovanější charakter (např. v případě červeného masa se později zjistilo, že jeho konzumenti také častěji kouří, což asi s rakovinou bude mít společného více než červené maso). Pozor tedy na zaručené „vědecké" informace; sami vědci velmi často nerozumí číslům, se kterými operují, nemluvě o reportérech, kteří nám jejich zjištění zprostředkují.

Ale zpět k formálním pojmům. Podmíněnou pravděpodobnost zapisujeme

$$P(A|B)$$

a čteme „pravděpodobnost jevu A za předpokladu, že nastal jev B". Podmíněnou pravděpodobnost lze vypočítat následovně:

$$P(A|B) = P(A, B)/P(B)$$

kde $P(A, B)$ je pravděpodobnost, že jevy A a B nastaly současně. Alternativně, pokud máme k dispozici výchozí statistický soubor, můžeme podmíněnou pravděpodobnost (resp. celé **podmíněné pravděpodobnostní rozložení**) odvodit tak, že se omezíme jen na objekty, u kterých platí jev B, a provést výpočet pravděpodobnostního rozložení pouze na nich. (Vzorec výše de facto funguje stejně.)

12.1 Nezávislé jevy

Skrze podmíněnou pravděpodobnost definujeme i tzv. **nezávislé jevy**. Intuitivně platí, že pokud jsou jevy nezávislé, pak by nám informace o jednom z nich neměla dát žádnou informaci o druhém z nich. Matematicky zapsáno, jevy A a B jsou nezávislé, pokud

$$P(A|B) = P(A) \quad \wedge \quad P(B|A) = P(B)$$

čili jsou nezávislé, pokud to, jestli nastal jev B, nijak neovlivní pravděpodobnost jevu A a naopak (ve skutečnosti stačí, aby platila jedna strana této konjunkce, neboť pak platí zcela jistě i druhá strana – zkuste si rozmyslet proč a dokázat).

Jen a pouze pro nezávislé jevy pak platí vzorec, který se snadno odvodí z nezávislosti jevů a z definice podmíněné pravděpodobnosti (Vyzkoušejte si jako cvičení!) a se kterým jsme často počítali na střední škole:

$$P(A, B) = P(A) * P(B)$$

Ještě jednou zdůrazněme: Pozor, toto platí pouze pro nezávislé jevy! Reálné jevy nebývají téměř nikdy dokonale nezávislé (i když v některých případech je oprávněné nezávislost předpokládat, např. při hodu dvěma mincemi to, co padne na první minci nejspíše neovlivní to, co padne na druhé).

12.2 Bayesův vzorec

Někdy nemáme přístup přímo ke statistickému souboru, z něhož vycházíme, ale známe některé pravděpodobnosti z něj odvozené. Např. když má test na drogy přesnost 99 %, znamená to obvykle:

$$P(\textit{pozitivní test} \mid \textit{testovaný požil drogu}) = 0,99$$
$$P(\textit{negativní test} \mid \textit{testovaný nepožil drogu}) = 0,99$$

Přitom nevíme, z jakých statistických dat byla tato čísla odvozena, ale důvěřujeme jim, protože např. pocházejí z důvěryhodného zdroje.

Pak nám často vzniká potřeba podmíněné pravděpodobnosti převádět, např. pokud uděláme někomu test, který vyjde pozitivně, chceme spočítat pravědpodobnost, s jakou testovaný jedinec opravdu požil drogu. Jinými slovy, chceme spočítat

$$P(\textit{testovaný požil drogu} \mid \textit{pozitivní test})$$

a obecně chceme převádět $P(A|B)$ na $P(B|A)$. Pokud známe také pravděpodobnosti $P(A)$ a $P(B)$, můžeme použít tzv. **Bayesův vzorec**:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Tento vzorec lze snadno dokázat s využitím definice podmíněné pravděpodobnosti (viz výše) pro $P(A|B)$ a $P(B|A)$ – zkuste si sami.

Pokud bychom chtěli spočítat pravděpodobnost z našeho příkladu, tedy zda testovaný požil drogu za předpokladu, že má pozitivní test, musíme tedy znát pravděpodobnosti

$P(\textit{testovaný požil drogu})$
$P(\textit{pozitivní test})$

Dejme tomu, že 0,5 % lidí užívá příslušnou drogu, tedy

$P(\textit{testovaný požil drogu}) = 0,005.$

Druhou z pravděpodobností můžeme vypočítat následovně:

$$\begin{aligned}
&P(\textit{pozitivní test}) = \\
&= P(\textit{pozitivní test}, \textit{požil drogu}) + P(\textit{pozitivní test}, \textit{nepožil drogu}) = \\
&= P(\textit{pozitivní test} \mid \textit{požil drogu}) * P(\textit{požil drogu}) \\
&\quad + \ P(\textit{pozitivní test} \mid \textit{nepožil drogu}) * P(\textit{nepožil drogu}) = \\
&= 0,99 * 0,005 + 0,01 * 0,995 \\
&= 0,0149
\end{aligned}$$

První krok je pouze vyjádření pravděpodobnosti jednoho jevu pomocí součtu pravděpodobností dvou jevů přes všechny hodnoty druhého jevu (rozmyslete a vyzkoušejte si, proč toto platí), druhý krok je opět použití definice podmíněné pravděpodobnosti.

Po dosazení do Bayesova vzorce dostáváme

$$P(\textit{testovaný požil drogu} \mid \textit{pozitivní test}) =$$

$$= \frac{P(\textit{pozitivní test} \mid \textit{požil drogu}) \ * \ P(\textit{požil drogu})}{P(\textit{pozitivní test})} = \frac{0{,}99 * 0{,}005}{0{,}0149} =$$

$$= 0,3322$$

Tedy pravdpodobnost, že testovaný s pozitivním testem skutečně požil drogu, je zhruba třetinová, jinými slovy dva ze tří lidí, kterým vyšel test pozitivně, drogu vůbec nepožili. Jak je to možné, když úspěšnost testu je téměř stoprocentní? Co by se muselo změnit, aby test pro nás fungoval dobře? (Nápověda: Zkuste předpokládat, že 50 % populace užívá drogu a provést stejný výpočet.)

Podobných zdánlivých paradoxů existuje více – zmíníme zde ještě např. tzv. *Monty Hall problem*.[8]

Obecně je důležité je uvědomit si význam čísel, ptát se po jejich přesném významu a nenechat se zmást na první pohled vysokými hodnotami – v praxi se používají testy i s daleko horší úspěšností než 99 %, např. při testech na různé choroby, prenatální diagnostice apod.; pravděpodobnost, že test vypovídá spolehlivě, může být tedy často o dost menší.

[8] http://en.wikipedia.org/wiki/Monty_Hall_problem

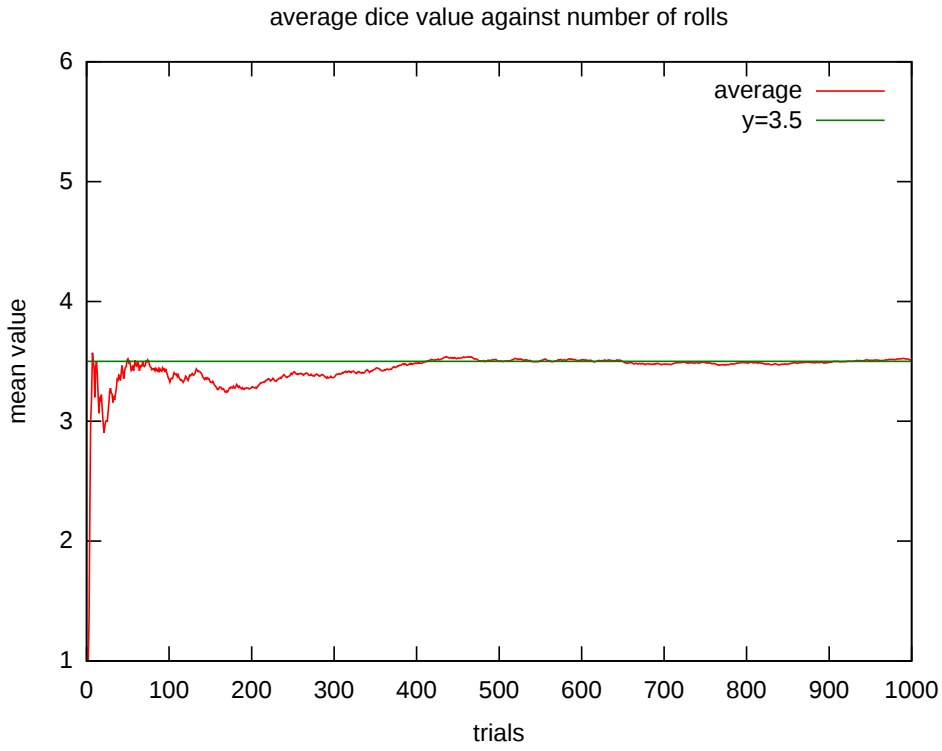
13 Zákon velkých čísel

Dále budeme pokračovat třemi kratšími kapitolami ze statistiky a pravděpodobnosti, které s látkou výše souvisí spíše volně, nicméně jsou velmi důležité.

Zákon velkých čísel říká v podstatě to, že aby výsledky založené na statistických metodách byly kvalitní a reprezentativní, potřebujeme je odvozovat z velkých statistických souborů. (To samozřejmě mimo jiné znamená, že všechny naše jednoduché příklady na statistiku tomuto zákonu odporují – pro účely procvičení nám to nevadí, je ale třeba, abychom si toho byli vědomi.)

Pro intuitivní ověření tohoto zákona můžeme udělat následující myšlenkový pokus: Pokud budeme pozorovat pohyb několika málo molekul v moři, bude se jejich pohyb jevit náhodný; naproti tomu pokud budeme sledovat významný podíl molekul, budeme schopni pozorovat jevy jako mořské proudy, vlny, příliv...

Matematická formulace zákona zní tak, že čím větší počet pokusů provedeme, tím větší je pravděpodobnost, že průměr výsledků pokusů bude odpovídat očekávanému vypočtenému průměru. Tato formulace věty je i formálně matematicky dokázána; tento důkaz však už přesahuje rámec našeho zájmu. Místo toho uvedeme jeden další myšlenkový pokus: Pokud budeme postupně házet vyváženou kostkou, průměr prvních dvou pokusů bude zcela náhodný; průměr 50 pokusů bude již velmi pravděpodobně blízký očekávanému 3,5, ale stále je zde poměrně velká možnost náhodných odchylek. Pokud provedeme 1000 pokusů, pak pravědpodobnost, že průměr nebude velmi blízko očekávané hodnotě 3,5, je mizivá. Pro ilustraci zde uvádíme graf jednoho z takových pokusů (zdroj: Wikipedia):



Podobně je tomu např. u zkoumání jazykových korpusů: Pokud se slovo (značka, kolokace, ...) v korpusu vyskytuje dvakrát, není o něm statistika schopna říci téměř nic. Pokud se vyskytuje padesátkrát, pak už lze vyvozovat nějaké závěry, nicméně stále hrozí poměrně velké náhodné odchylky. Pokud se vyskytuje tisíckrát, můžeme naše statistické výpočty považovat za spolehlivé. Tento fakt je také příčinou zájmu o budování korpusů o velikosti desítek miliard slov nebo i větších.

Vždy tedy musíme naše statistické závěry zakládat na velkém počtu výskytů.

14 Statistické testování hypotéz

Statistické testování hypotéz je standardní exaktní metodika, s jejíž pomocí je možné statisticky prokázat nějakou hypotézu s pomocí pozorovaných dat. Jinými slovy chceme zjistit, zda naše statistická data potvrzují nějakou hypotézu. Slovo „prokázat" zde budeme chápat nikoli ve smyslu prokázat dokonale a nade vši pochybnost, ale tak, že pravděpodobnost chyby našeho úsudku bude minimalizována.

Vysvětlíme si celou věc na následujícím pokusu: Ukazujeme člověku postupně hrací karty z rubu, tak, aby neviděl lícovou stranu. Úkolem člověka je uhodnout jednu ze čtyř barev karty, která je mu ukázána, aniž by se podíval na líc. Necháme ho hádat postupně např. 25 karet. Otázka zní: Kolikrát musí odpovědět správně, abychom ho prohlásili za „jasnovidce" – člověka, který skutečně *ví* něco o tom, která karta je mu zrovna ukázána, a nehádá pouze náhodně?

Selským rozumem nejspíš usoudíme, že pokud uhodne např. 5x, bude to asi náhoda; pokud uhodne 24x, bude tento člověk nejspíš jasnovidcem. Ale jsou tyto naše „selské" úsudky správné? Již jsme měli možnost se přesvědčit, že neplatí vždy to, co je zdánlivě zřejmé na první pohled. Další otázkou je – co usoudíme, když uhodne např. 12x? Jak určit hranici, odkdy je člověk „jasnovidec"?

Z toho, co už víme o pravděpodobnosti je jasné, že např. i 25 úspěšných pokusů při hádání karet může být náhoda; může se stát, že člověk uhodl 25x a přitom o skutečné barvě karet opravdu nic nevěděl. Co dále umíme, je stanovit *pravděpodobnost takové události*. A právě vyjádření pravděpodobnosti toho, že dané výsledky vznikly náhodně, je základem metodiky testování hypotéz.

Statistické testování hypotéz zavádí několik základních pojmů:

- **Nulová hypotéza**, značeno H_0, je předpoklad, který obvykle platí a který chceme našimi statistickými daty vyvrátit. V případě hádání karet by H_0 byla „pokusná osoba hádá náhodně".
- **Alternativní hypotéza**, značeno H_1, je tvrzení, pro které hledáme oporu v našich datech. Musí být doplňková k H_0, tedy výrok „$H_0 \vee H_1$" musí být tautologie, žádná třetí možnost není přípustná. V našem případě by H_1 byla „pokusná osoba nehádá náhodně".
- **Chyba typu I** nastává, pokud potvrdíme H_1, neboli **vyvrátíme nulovou hypotézu**, ale přitom platí H_0. Úkolem testování hypotéz je **minimalizovat pravděpodobnost této chyby**.
- **Chyba typu II** nastává, pokud nepotvrdíme H_1 a ta je přitom platná. Tato chyba není tak závažná.

Analogii chyb typu I a II můžeme spatřovat v trestním právu – výchozím předpokladem je nevina obžalovaného (nulová hypotéza). Pokud uvězníme nevinného (chyba typu I, vina byla prokázána a dotyčný byl přitom nevinen), dopustíme se horšího činu než když osvobodíme viníka (chyba typu II) na základě nedostatečných důkazů. Podobně pokud se při testování hypotéz nepodaří vyvrátit nulovou hypotézu, neznamená to, že nulová hypotéza platí, podobně jako osvobození obžalovaného neprokazuje jeho nevinu.

Tento fakt bývá rovněž velmi často opomíjen (i ve vědeckých kruzích), takže ho zopakujeme ještě v alternativní formulaci: To, že se nepodařilo prokázat souvislost neznamená, že souvislost neexistuje (pouze ji dostupná data nepotvrzují s dostatečnou jistotou).

Doveďme nyní náš příklad do konce: Máme dánu nulovou a alternativní hypotézu a předpokládejme, že pokus dopadl tak, že dotyčný uhodl barvu ve všech 25 případech správně. Na základě toho jsme usoudili, že nehádá barvu náhodně (řekli jsme, že nulová hypotéza je vyvrácena). Pravděpodobnost chyby typu I je pak za předpokladu nezávislosti pokusů $(1/4)^{25} = cca\ 10^{-15}$ (1 tip má pravděpodobnost úspěchu 1/4 a bylo provedeno 25 nezávislých pokusů). Vidíme, že tato pravděpodobnost je naprosto zanedbatelná, čili řekneme, že se podařilo prokázat, že dotyčný nehádal karty náhodně.

Obvykle pro vyvrácení nulové hypotézy požadujeme nějakou maximální přípustnou pravděpodobnost chyby typu I. Tato mez se může lišit v závislosti na aplikaci – nejčastěji bývá požadováno max. 1 %, můžeme být ale striktnější a požadovat např. desetinu nebo i setinu procenta, nebo naopak méně striktní a požadovat 5 %. Výsledku, který byl takovýmto způsobem ověřen (tj. nulová hypotéza byla vyvrácena s pravděpodobností chyby typu I méně než X %), pak říkáme **statisticky průkazný**.

Již bez výpočtu (který by byl v tomto případě komplikovanější) doplňme, že v případě našeho pokusu by stačilo 12 úspěšně uhodnutých karet k tomu, abychom vyvrátili nulovou hypotézu s pravděpodobností chyby typu I méně než 1 %.

Závěrem této kapitoly upozorníme na jedno nebezpečí v případě testování hypotéz, v duchu principu, že nad čísly musíme přemýšlet. Pokud provedeme pokus o vyvrácení nulové hypotézy vícekrát, logicky se zvyšuje pravděpodobnost, že se nám to alespoň jednou podaří i přes to, že nulová hypotéza platí (zvyšuje se pravděpodobnost chyby typu I). Při opakování pokusů musíme tedy brát v potaz *všechna* data, která jsou k dispozici, nikoliv jen ten vzorek, na němž to (dost možná náhodou) vyšlo správně.

To také souvisí s problematikou publikací vědeckých výsledků: pokud není nalezena souvislost, výsledek obvykle není významný, není tedy publikován a příslušná data často zapadnou. Pokud *je* nalezena souvislost (relativně nedávno např. probíhaly diskuse tohoto typu o vlivu oxidu uhličitého na globální

oteplování), výsledek je obvykle publikován a považován za průkazný. Téměř nikdy ale nejsou brána v potaz předchozí data, která souvislost neprokázala, což zvyšuje pravděpodobnost chyby publikovaných výsledků. Zůstává přitom otevřenou otázkou, zda je možné v tomto ohledu něco výrazně změnit.

15 Entropie

Entropie náhodné veličiny je zjednodušeně řečeno **míra informace** obsažená v této náhodné veličině, tedy číslo vyjadřující, kolik informace získáme, když se dozvíme skutečnou hodnotu (tu, která nastala) náhodné veličiny. Entropie je vždy nezáporná a měří se v bitech. Nulová entropie znamená, že hodnotu náhodné veličiny jsme schopni určit se stoprocentní jistotou (takovou veličinou může být např. výsledek hodu falešnou kostkou, na které padají pouze šestky). Čím vyšší je entropie, tím více informace je obsaženo v náhodné veličině – jinými slovy tím více jednotek informace je třeba k přenesení výsledku náhodného pokusu.

Historické počátky úvah o statistické entropii sahají do 40. let 20. století, kdy byly zkoumány metody, jak lze s minimální přenosovou kapacitou přenést určitou zprávu, a kdy byly též vyvíjeny první kompresní algoritmy. Tyto počátky jsou spojeny se jménem Claude Shannon. Samotná entropie skutečně vyjadřuje minimální průměrný počet bitů (jedniček a nul), který je potřeba pro přenesení určité informace, v našem případě uvažované jako výsledek náhodného pokusu.

Entropie náhodné veličiny X – $H(X)$ – nebo ekvivalentně entropie jejího pravděpodobnostního rozložení p – $H(p)$ – se spočítá následovně:

$$-\sum_{x \in X} p(x) log_2 p(x)$$

Např. entropie počtu orlů při házení 2 mincemi se vypočte následovně. Nejprve vyjádříme pravděpodobnosti jednotlivých možných hodnot:

$$p(0) = 1/4, \quad p(1) = 1/2, \quad p(2) = 1/4$$

a pak dosadíme do vzorce:

$$H(p) = -(1/4 log_2(1/4) + 1/2 log_2(1/2)) + 1/4 log_2(1/4)) = \\ -(-2/4 - 1/2 - 2/4) = 1,5 \; bitu$$

V případě, kdy uvažujeme entropii jako nutnou informaci pro přenos, lze tuto hodnotu interpretovat tak, že při přenášení informace o sérii hodů dvěma mincemi potřebujeme průměrně 1,5 bitu na jeden pokus – jednu možnost můžeme kódovat 1 bitem, dvě další (ty méně pravděpodobné) 2 bity.

Zkuste si sami spočíst entropii pro případ hodu jedinou mincí. Očekávaná hodnota výsledku je 1 bit, neboť např. 1 orla můžeme reprezentovat jedničkou, žádného nulou – více informace nepotřebujeme.

15.1 Podmíněná entropie

Podobně jako jsme zavedli podmíněnou pravděpodobnost lze zavést podmíněnou entropii s analogickým významem – míra informace obsažená v náhodné veličině za předpokladu, že známe hodnoty jiné náhodné veličiny. Je definována jako

$$H(X|Y) = \sum_{x \in X} p(x) H(Y \mid X = x)$$

Podobně jako u podmíněné pravděpodobnosti, lze entropii dvojice náhodných veličin převést na podmíněnou entropii a naopak. K tomu slouží tzv. **řetízkové pravidlo** (chain rule):

$$H(X, Y) = H(X) + H(Y|X)$$

15.2 Mutual information

Mutual information – česky **vzájemná informace** – je statistickou veličinou založenou na entropii, která určuje míru informace, kterou jedna náhodná proměnná říká o jiné. Je rovna nule, pokud jsou veličiny nezávislé, a čím vyšší její hodnota je, tím hodnoty jedné vlastnosti určují hodnoty druhé vlastnosti. Vzorec pro *mutual information* je:

$$MI(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Ve statistickém zpracování přirozeného jazyka vzájemnou informaci můžeme použít k měření *síly kolokace* dvou slov – v tomto případě uvažujeme výskyt jednoho slova (např. „základní") za jednu náhodnou veličinu a výskyt druhého slova (např. „škola") v textu. Ze vzorce *mutual information* vyplývá, že je tím vyšší, čím vyšší je počet souvýskytů těchto slov v korpusu a tím nižší, čím jsou jednotlivá slova častější. *Mutual information* pro slova „základní" a „škola" bude tedy zřejmě mnohem vyšší než např. pro „počítač" a „škola".

16 Statistika a zpracování přirozeného jazyka

Na závěr našeho výkladu si ukážeme několik aplikací statistiky při automatickém zpracování přirozeného jazyka. Jak už víme, statistika je nástroj, který nám umožňuje uchopit a popsat velké množství dat a na tomto základě lze vyvozovat další informace o objektech, které jsou v našich datech obsaženy, např. za účelem stanovení pravděpodobností různých jevů nebo predikce událostí.

Pro účely statistického zpracování přirozeného jazyka jsou k dispozici velké soubory jazykových dat v podobě jazykových korpusů o velikosti i více než 10 miliard slov. Tyto korpusy umožňují aplikovat statistické metody tak, jak jsme si je přiblížili, na přirozený jazyk, predikovat jeho chování a modelovat jej. Využití statistiky ve zpracování přirozeného jazyka je obrovské a nástroje založené na statistickém modelování jazyka jsou v současnosti v mnoha oblastech ty vůbec nejlepší.

My si ukážeme využití statistiky na dvou ukázkách – první bude vyhledávání kolokací, druhou pak n-gramové jazykové modely.

16.1 Vyhledávání kolokací

Definice pojmu kolokace se v různých zdrojích liší. Podle některých je to fráze, jejíž význam se neskládá z významů jejích částí, či idiom, podle jiných je to pouze nějakým způsobem významné spojení dvou slov. Statistika nabízí svůj vlastní pohled, který v mnoha ohledech koresponduje s oběma výše nastíněnými definicemi – kolokaci můžeme považovat za *statisticky významné* spojení dvou slov (v korpusu), a to podle různých metrik. Navíc podle většiny těchto metrik umíme určit i „sílu" kolokace libovolných dvou slov.

Jakým způsobem můžeme tedy hledat v korpusu kolokace? Můžeme začít s prostým počítáním pravděpodobností výskytů dvojic slov v korpusu. Pak se nám ale (např. v případě angličtiny) na nejvyšší příčky dostanou dvojice jako „of the" nebo „in the", což asi úplně neodpovídá tomu, co si představujeme pod pojmem kolokace. Zřejmě potřebujeme nějakým způsobem znevýhodnit slova z tzv. *stoplistu*.

Jeden z možných způsobů, jak tohoto docílit, je uvažovat při vyhledávání kolokací pouze dvojice složené z určitých slovních druhů, např. podstatných jmen, přídavných jmen a sloves, popřípadě příslovcí. Již s takovýmto jednoduchým omezením dostaneme mnohem lepší výsledky; mezi nejlepšími však stále ještě budou dvojice jako např. „last week", kde je otázka, zda je chceme považovat za kolokace nebo ne. Jisté je, že slova jako „last", „new" apod. budou mít časté souvýskyty s velkým množstvím slov a ne všechny z nich budeme považovat za významné. Co tedy dále můžeme potřebovat, je

znevýhodnit nejen slova ze *stoplistu*, ale všechna častější slova, a to poměrně.

K tomuto účelu slouží celá řada statistických metrik (je jich více, protože pro různé aplikace se hodí různé z nich a nelze vybrat jednu univerzálně nejlepší) – např. T-test, což je aplikace testování hypotéz. Nulová hypotéza v tomto případě říká, že slova se chovají náhodně podle svých obvyklých pravděpodobnostních rozložení. Pokud je nulová hypotéza vyvrácena (slova se spolu vyskytují častěji než náhodně), prohlásíme příslušnou dvojici slov za kolokaci. Z pravděpodobnosti chyby typu I můžeme odvodit sílu kolokace a podle této pravděpodobnosti můžeme kolokace setřídit.

Mezi další podobné metriky patří vzájemná informace, představená v minulé kapitole, nebo tzv. *dice* a *logdice*, které používá pro sestavování kolokačních profilů (tzv. *word sketches*) komerční korpusový nástroj *Sketch Engine*.

16.2 N-gramové jazykové modely

Jako n-gram označujeme obvykle posloupnost slov délky n, která se vyskytuje v korpusu. Nejjednodušší představa n-gramového jazykového modelu vede přes následující úlohu: Známe posloupnost $n-1$ slov v korpusu. Jaké slovo za nimi bude následovat?

Formálně se n-gramový jazykový model skládá z podmíněných pravděpodobností

$$P(w_n \mid w_1, ..., w_{n-1})$$

čili pravděpodobností, že se vyskytlo slovo w_n za předpokladu, že *před ním* se vyskytla slova $w_1, ..., w_{n-1}$. Souhrn všech takovýchto pravděpodobností pro všechny možné kombinace slov v korpusu (tento souhrn snadno odvodíme z textu korpusu) se nazývá **n-gramový jazykový model**.

$w_1, ..., w_n$ přitom nemusí být pouze slova – můžeme vytvořit n-gramový model znaků, fonémů, pádů, morfologických značek apod., případně i komplikovanější modely, kde např. w_n bude morfologická značka a $w_1, ..., w_{n-1}$ budou slova. Využití n-gramových modelů ve zpracování jazyka je obrovské, proto existuje nepřeberné množství, z nichž nejpoužívanější je pravděpodobně trigramový ($n = 3$) jazykový model.

Například bigramový ($n = 2$) jazykový model na znacích pro text „mama mele maso" by byl následující (mezery v tomto případě pro přehlednost ignorujeme):

$$P(a|m) = 3/4$$
$$P(e|m) = 1/4$$

$$P(m|a) = 2/3$$
$$P(s|a) = 1/3$$
$$P(l|e) = 1/2$$
$$P(m|e) = 1/2$$
$$P(e|l) = 1$$
$$P(o|s) = 1$$
$$P(\$|o) = 1$$
$$P(x|y) = 0 \textit{ pro všechny ostatní kombinace}$$

Všimněte si, že podle omezení kladená na pravděpodobnostní rozložení musí součet všech pravděpodobností se stejnou posloupností za svislítkem být roven 1. Rovněž upozorňujeme na symbol „$", který používáme ve významu „konec textu", abychom mohli mít model úplný. Na závěr ještě zopakujeme, že stejně jako všechny „malé" příklady, i tento odporuje zákonu velkých čísel a pro rozumně reprezentativní jazykový model potřebujeme mnohem větší vzorek textů.

Úloha „hádání dalšího slova" samozřejmě nemá sama o sobě žádný praktický význam. I přesto jsou n-gramové modely ve zpracování jazyka používány téměř všude – od rozpoznávání řeči přes morfologické značkování i syntaktickou analýzu až po strojový překlad. Jazykový model nám totiž umožňuje odhadnout nejpravděpodobnější sekvence symbolů, které odpovídají danému (víceznačnému) vstupu – např. při rozpoznávání řeči se od sebe špatně rozeznávají jednotlivé souhlásky, čili jeden zvuk může teoreticky mít několik různých významů. Určení nejpravděpodobnější sekvence na základě jazykového modelu a daného vstupu řeší algoritmy zpracování přirozeného jazyka, které jsou však již nad rámec našeho výkladu.

16.2.1 Nedostatky jazykových modelů

Prvním problémem jazykových modelů je, že mohou být velké – často pro větší n dostaneme tolik různých pravděpodobností, že počítačová práce s takovýmto modelem je výpočetně velmi náročná. Malé n nám zase často neposkytuje dostatečný kontext na to, abychom na základě příslušného modelu prováděli kvalitní odhady – často se vzájemně ovlivňují slova, která jsou jednoduše příliš daleko od sebe. Efektivita modelu samozřejmě také závisí na základních jednotkách (tzv. parametrech) modelu – např. znaků je mnohem méně než slov, a tedy i modely na znacích budou mnohem menší než analogické modely na slovech.

Ještě větším problémem je tzv. *data sparseness* (řídkost dat), což znamená, že pro slova a n-gramy, které se nevyskytují tak často (a takových je i v desetimiliardových korpusech stále velké množství) dostaneme nekvalitní

odhady pravděpodobností a podle zákona velkých čísel i podle praktických experimentů fungují části jazykových modelů obsahující tato slova a n-gramy skutečně špatně.