
Základy matematiky a statistiky pro humanitní obory

studijní text k předmětům PLIN004 a PLIN006

Předměty jsou určeny studentům humanitního zaměření (především oboru Český jazyk se specializací počítačová lingvistika), kteří pro své studium potřebují nejnnutnější 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.

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 spočítat 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 zaměřené 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 otevrou 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.

1 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.

1.1 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.

2 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.

2.1 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
 - $v(A \Rightarrow B) = 0$, pokud $v(A) = 1$ a $v(B) = 0$,
 - $v(A \Rightarrow B) = 1$ 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

¹Nicméně, 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.

(Dejte vždy pozor na to, že je nutné vypsát 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**.

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.

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

2.2 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 definovanou 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

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

Po částech lze formuli „přeložit“ jako: prvočíslo je číslo větší nebo rovno 2, kde pro všechny možné dělitele (y) platí, že pokud existuje rozklad x na $(y * z)$ (neboli pokud y je skutečně dělitelem), pak jeden z členů tohoto rozkladu je 1 (a druhý x ; to už ale není třeba explicitně zmiňovat).

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

2.3 Matematická indukce

Matematická indukce je v první 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, \dots, x_n, \dots 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:

$$1 + 2 + \dots + n = n/2 * (1 + n)$$

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

$$1 + 2 + \dots + k = k/2 * (1 + k)$$

Dokážeme, že platí

$$1 + 2 + \dots + k + (k + 1) = (k + 1)/2 * (1 + (k + 1))$$

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

- $1 + 2 + \dots + k + (k + 1)$
- $k/2 * (1 + k) + (k + 1)$
- $(k + k^2)/2 + (k + 1)$

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

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 indukci 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.

3 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 ($\{\}$).

3.1 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 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 $\{x \mid x \in \mathbb{N} \wedge x > 5\}$ obsahuje přirozená čísla větší než 5. Jako zkratka pro tento zápis se též často používá $\{x \in \mathbb{N} \mid x > 5\}$.⁵

3.2 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. Doplnkem operátoru \in je \notin , které definujeme takto:

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

⁴Problémy 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*.

⁵Pokud 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.

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:

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

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 množina 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:

⁶Rozmyslete si proč.

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

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** (\cup) a **sjednocení** (\cap). Než budete číst dál, zkuste si je definovat sami. Správné definice jsou:

$$\begin{aligned} A \cap B &= \{x \mid x \in A \wedge x \in B\} \\ A \cup B &= \{x \mid x \in A \vee x \in B\} \end{aligned}$$

4 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.

4.1 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?

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

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

4.2 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)

- $1 + 2$
- $1 + S(1)$
- $S(1 + 1)$
- $S(1 + S(0))$
- $S(S(1 + 0))$
- $S(S(1))$
- $S(S(S(0)))$
- $= 3$

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.

4.3 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.

5 Relace a funkce

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

5.1 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 nejprůchoďejší.

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
- $(a_1, a_2, a_3, \dots, a_n) \equiv (a_1, (a_2, (a_3, (\dots, a_n) \dots)))$

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).

5.2 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.

5.3 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.

5.4 Vlastnosti relací

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

5.5 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

$$f : A \rightarrow B$$

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 “.

5.6 Vlastnosti funkcí

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

Injektivita. Funkce $f : A \rightarrow B$ 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 $f : A \rightarrow B$ 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 $f : A \rightarrow B$ 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 $f : A \rightarrow B$, znamená to, že množiny A a B jsou „stejně velké“ (za okamžik tuto vágní poznámku rozvedeme formálně).

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

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

5.7 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 zdefinujeme velikost množiny A (značeno $|A|$) formálně:

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

- množina A má **mohutnost kontinua**, pokud existuje bijekce $f : \mathbb{R} \rightarrow A$ (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.

5.8 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 $a_0, a_1, \dots, a_n, \dots$, což považujeme jen za jiný druh zápisu pro $f(0), f(1), \dots, f(n), \dots$.

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:

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

6 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 lingvistiky.

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 nezákladnějšími konstrukcemi teorie formálních jazyků, neboť je velmi pravděpodobné, že se s nimi a jejich modifikacemi při studiu ještě mnohokrát setkáte.

6.1 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. $\{a, b\}^* = \{\epsilon, a, b, aa, bb, ab, ba, aab, abb, \dots\}$.

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

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

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

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:

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

6.2 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 $V^* \cdot N \cdot V^* \times V^*$ – 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 $\alpha \rightarrow \beta$. 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

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

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?

6.3 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 $\alpha \rightarrow \beta$ podmínku $|\alpha| \leq |\beta|$, tedy levá strana každého pravidla musí být kratší než jeho pravá strana. Výjimkou z tohoto omezení je pravidlo $S \rightarrow \epsilon$, které může být přítomno.

⁷zde 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 $A \rightarrow \beta$ (tak, že $A \in N$), tedy na levé straně je vždy právě jeden neterminál a β je neprázdné. Výjimkou z tohoto omezení je pravidlo $S \rightarrow \epsilon$, které může být přítomno.

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

6.4 Konečný automat

Automaty jsou jiným abstraktním modelem charakterizujícím jazyky. Fungují jako stavové systémy, které čtou po znacích slovo na vstupu, s přečtením každého znaku změni 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

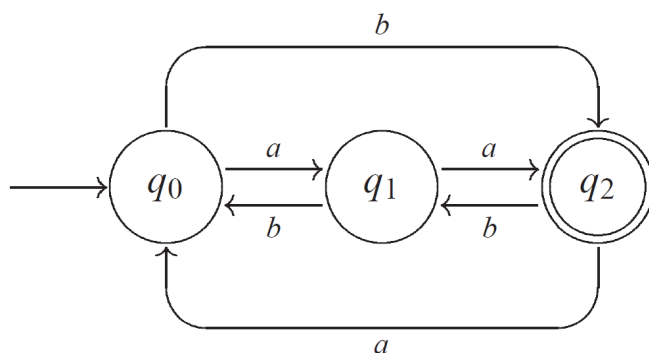
- Q je neprázdná konečná množina stavů automatu
- Σ je konečná množina vstupních symbolů (abeceda)
- $\delta : Q \times \Sigma \rightarrow Q$ je přechodová funkce automatu
- q_0 je počáteční stav
- F je množina koncových stavů

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 $L(A)$.

6.5 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.

7 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.

7.1 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, \dots, A_n o velikostech p_1, p_2, \dots, p_n má množina $A_1 \cup A_2 \cup \dots \cup A_n$ velikost $p_1 + p_2 + \dots + p_n$.

Druhým je **pravidlo součinu** říkájící, 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 $n_1 * n_2 * \dots * n_k$.

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

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

7.2 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,

7.2 Pravděpodobnost 7 KOMBINATORIKA A PRAVDĚPODOBNOST

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