I. Matematické základy informatiky

Update: 13. května 2018

Obsah

1 Konečné automaty, regulární výrazy, uzávěrové vlastnosti třídy regulárních jazyků.

1.1 Konečné automaty (KA)

Konečný automat (KA) tvoří množina stavů, vstupní abceda, přechodová funkce, počáteční a koncové stavy. Můžeme jej znázornit jako **tabulku**, **graf** či **strom**.

Konečné automaty se dělí na **determistické** a **nedetermistické**. Deterministický konečný automat má pouze jeden počáteční stav a přechodová funkce vrací jeden stav. Zatímco nedeterministický KA může mít více počátečních stavů a přechodová funkce vrací množinu stavů.

- **Slovo** přijaté automatem je taková sekvence symbolů (ze vstupní abecedy), pro kterou automat skončí v koncovém stavu.
- **Regulární jazyk** je takový jazyk (množina slov) který lze popsat konečným automatem.

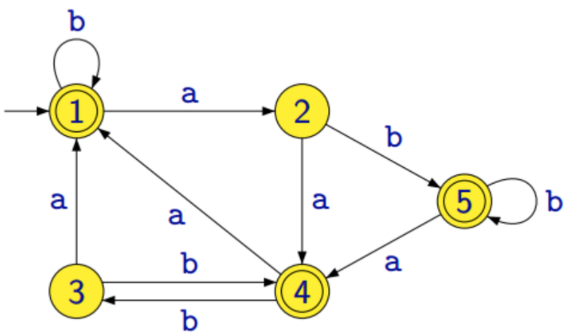
1.1.1 Deterministický konečný automat (DKA)

Skládá se ze **stavů** a **přechodů**. Jeden ze stavů je označen jako **počáteční stav** a některé jsou označeny jako **přijímací**. **Je definován jako uspořádaná pětice** $(Q, \Sigma, \delta, q_0, F)$, kde:

- Q je konečná neprázdná množina **stavů**.
- Σ (*sigma*) je konečná neprázdná množina vstupních symbolů, tzv. **vstupní abeceda**.
- δ (*delta*) je **přechodová funkce**, $\delta : Q \times \Sigma \to Q$.
- q_0 je **počáteční stav**, $q_0 \in Q$.
- F je neprázdná množina **koncových** neboli **přijímajících stavů**, $F \subseteq Q$.

Příklad

- $Q = \{1, 2, 3, 4, 5\}$, $\Sigma = \{a, b\}$, $F = \{1, 4, 5\}$
- $\delta(1, a) = 2$; $\delta(1, b) = 1$; $\delta(3, a) = 1$; $\delta(3, b) = 4$; $\delta(2, a) = 4$; $\delta(2, b) = 5$; $\delta(4, a) = 1$; $\delta(4, b) = 3$; $\delta(5, a) = 4$; $\delta(5, b) = 5$

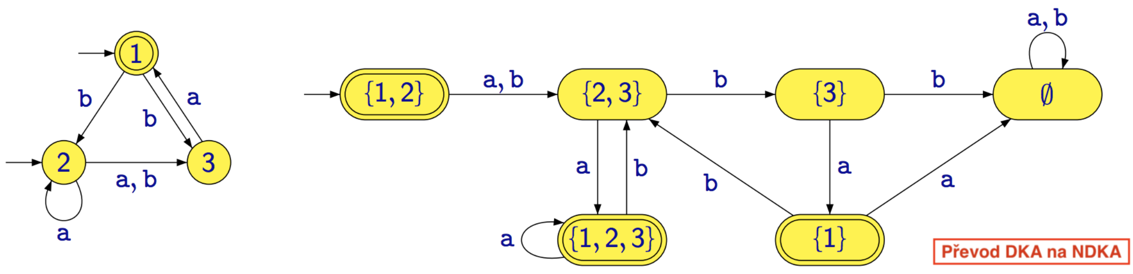


δ	a	b
↔ 1	2	1
2	4	5
3	1	4
← 4	1	3
← 5	4	5

1.1.2 (Zobecněný) Nedeterministický konečný automat ((Z)NKA)

Formálně je NKA definován jako pětice $A = (Q, \Sigma, \delta, I, F)$, s tím rozdílem, že oproti deterministickému KA má **více počátečních stavů** a **přechodová funkce vrací množinu** stavů. V případě ZNKA zde existují navíc **nulové epsilon** (ϵ) přechody:

- δ je přechodová funkce, vrací množinu stavů, $\delta : Q \times \Sigma \rightarrow P(Q)$, v případě **ZNKA** $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$.
- I je konečná množina počátečních stavů, $I \in Q$.



Na rozdíl od deterministického automatu:

- Může z jednoho stavu vést **libovolný počet přechodů** označených stejným symbolem (i **nulové** ϵ v případě ZNKA).
- Není zde nutné, aby z každého stavu vystupovaly všechny symboly, které do něj vstoupily $\rightarrow$ **nemusí ošetřovat všechny varianty**, pouze odhadne, kterou cestou půjde.
- Nedeterministický automat přijímá dané slovo, jestliže **existuje alespoň jeden jeho výpočet**, který vede k přijetí tohoto slova.
- V automatu může být **víc než jeden počáteční stav**.
- Lze ho **převést na deterministický** (formou tabulky). Při převodu automatu, který má n stavů může mít výsledný nedeterministický až 2^n stavů.

1.1.3 Normovaný tvar

Začnu v počátečním stavu a procházím navštívené stavy a vytvářím tabulku. Každý KA má **právě 1** normovaný tvar. Také lze tímto způsobem zjistit, zda jsou automaty **ekvivalentní**.

1.2 Regulární výrazy

Regulární výraz je **řetězec popisující celou množinu řetězců**, konkrétně **regulární jazyk**. Regulární výrazy také můžeme chápat jako jednoduchý způsob, jak **popsat konečný automat** umožňující generovat všechna možná slova patřící do daného jazyka.

V regulárních výrazech využíváme znaky **abecedy** a symboly pro **sjednocení**, **zřetězení** a **iterace** regulárních výrazů. Za regulární výraz se považuje i samotný znak abecedy (např. a) stejně jako **prázdné slovo** ϵ a **prázdný jazyk** $\emptyset$.

1.2.1 Definice regulárních výrazů

Regulární výrazy popisují jazyky nad abecedou $A = \Sigma : \emptyset, \epsilon, a$ (kde $a \in \Sigma$) jsou regulární výrazy:

- $\emptyset$ označuje **prázdný jazyk**,
- ϵ označuje jazyk $\{\epsilon\}$,
- a označuje jazyk $\{a\}$.

Dále, jestliže α, β jsou regulární výrazy, pak i $(\alpha + \beta)$, $(\alpha \cdot \beta)$, $(\alpha *)$ jsou regulární výrazy, kde:

- $(\alpha + \beta)$ označuje **sjednocení** jazyků označených α a β,
- $(\alpha \cdot \beta)$ označuje **zřetězení** jazyků označených α a β,
- $(\alpha *)$ označuje **iteraci** jazyka označeného α.

Neexistují žádné další regulární výrazy než ty definované podle předchozích dvou bodů.

Příklady

Ve všech případech je $\Sigma = \{0, 1\}$:

- **01** (0 a 1) ...jazyk tvořený jedním slovem 01,
- **0+1** (0 nebo 1) ...jazyk tvořený dvěma slovy 0 a 1,
- **(01)*** ...jazyk tvořený slovy $\epsilon, 01, 0101, 010101, \ldots$,
- **(0+1)*** ...jazyk tvořený všemi slovy nad abecedou $\{0, 1\}$,
- **(01)*111(01)*** ...jazyk tvořený všemi slovy obsahující podslovo 111, předcházení i následované libovolným počtem slov 01,
- **(0+1)*00+(01)*111(01)*** ...jazyk tvořený všemi slovy, která buď končí 00 nebo obsahují podslovo 111 předcházené i následované libovolným počtem slov 01,
- **(0+1)*1(0+1)*** ...jazyk tvořený všemi slovy obsahujícími alespoň jeden symbol 1,
- **0*(10*10*)*** ...jazyk tvořený všemi slovy obsahujícími sudý počet symbolů 1.

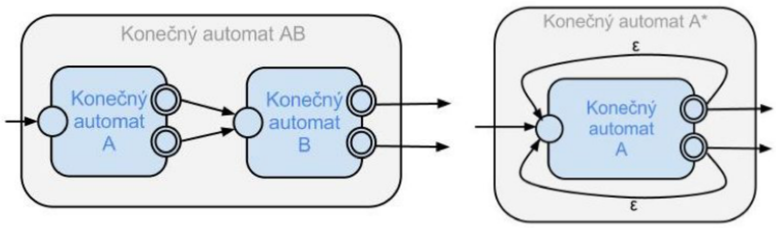
1.3 Uzávěrové vlastnosti třídy regulárních jazyků

Uzavřenost množiny nad operací znamená, že výsledek operace s libovolnými prvky z množiny bude opět spadat do dané množiny. Třídu regulárních jazyků značíme **REG**. Regulární výrazy (tedy i KA) jsou uzavřené vůči operacím:

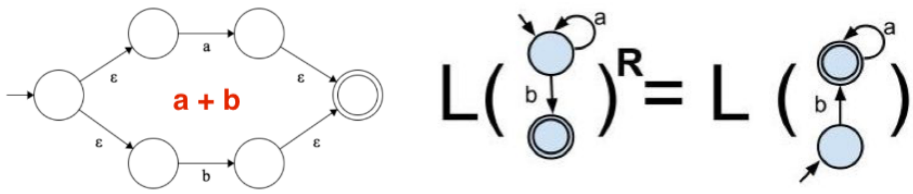
- **Sjednocení**, **průnik**, **doplněk** – je-li $L_1, L_2 \in \text{REG}$, pak také $L_1 \cup L_2, L_1 \cap L_2, L_1'$ jsou v REG.
- **Zřetězení**, **iterace** – je-li $L_1, L_2 \in \text{REG}$, pak také $L_1 \cdot L_2, L_1^*$ jsou v REG.
- **Zrcadlový obraz** – je-li $L \in \text{REG}$, pak také L^R jsou v REG.

1.3.1 Operace sjednocení, zřetězení, iterace a zrcadlový obraz u KA

- **Iterace** – **spojíme koncové stavy** jednoho KA s **počátečními** druhého KA ϵ přechodem. Na obrázku generuje automat A^* jazyk $L(A^*) = L(A)^*$, který je iterací jazyku generovaného modrého automatu A.
- **Zřetězení** – **spojíme koncové stavy jednoho** s **počátečními stavy druhého**. Na obrázku generuje konečný automat AB jazyk $L(AB) = L(A) \cdot L(B)$.



- **Sjednocení** – $L(A + B) = L(A) + L(B)$ získáme tak, že vytvoříme **nový počáteční stav**, ze kterého vedeme ϵ přechody do počátečních stavů obou automatů. Poté obdobě z koncových stavů obou automatů vedeme ϵ přechody do **nového koncového**.
- **Zrcadlový obraz** – pustíme automat pozpátku, celý jej převrátíme. **Přehodíme orientaci všech přechodů**, z počátečních stavů uděláme koncové a naopak.



- **Doplněk** – u DKA provedeme prohození označení příjmajících a ostatních stavů, u NKA je nejprve nutné provézt převod na DKA.

2 Bezkontextové gramatiky a jazyky. Zásobníkové automaty, jejich vztah k bezkontextovým gramatikám.

2.1 Bezkontextové gramatiky (BG)

Bezkontextová gramatika definuje **bezkontextový jazyk**. Je tvořena **neterminály** (proměnné), **terminály** (konstanty) a **pravidly**, které každému neterminálu definují přepisovací pravidla. Jeden neterminál označíme jako **startovní**, kde začínáme a podle pravidel je dál přepisujeme na výrazy složené z terminálu a neterminálu. Jakmile už není co přepisovat, výraz obsahuje už jen neterminály, získali jsme **slovo**.

- Je **uzavřená** vůči operacím **sjednocení**, **zřetězení**, **iteraci** a **zrcadlový obraz**.
- Ke každé bezkontextové gramatice existuje **ekvivalentní zásobníkový automat**.

2.1.1 Formální definice BG

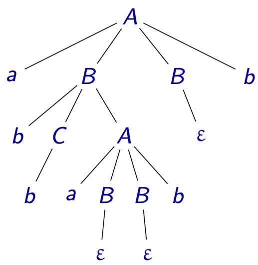
Bezkontextová gramatika je definována jako uspořádaná čtveřice $G = (\Pi, \Sigma, S, P)$, kde:

- Π (*velké pí*) je konečná množina **neterminálních** symbolů (neterminálů).
- Σ je konečná množina **terminálních** symbolů (terminálů), $\Pi \cap \Sigma = \emptyset$.
- S je **počáteční neterminál**, $S \in \Sigma$.
- P je konečná množina **přepisovacích pravidel**, $P \subseteq \Pi \times (\Pi \cup \Sigma)^*$.

2.1.2 Základní pojmy

- **Bezkontextový jazyk** – formální jazyk, který je akceptovaný nějakým zásobníkovým automatem.
- **Derivace slova** – jedno konkrétní odvození slova pomocí gramatiky, tedy záznam postupných přepisů od startovního neterminálu po konečné slovo. Derivace se podle postupu při přepisování dělí na:
  - **levou** – přepisujeme nejprve levé neterminály,
  - **pravou** – přepisujeme nejprve pravé neterminály.
- **Derivačni strom** – grafické znázornění derivace slova stromem. Pro všechny možné derivace (levou, pravou, moji) by měl derivační strom být **stejný**. Není-li tomu tak jedná se o **nejednoznačnou gramatiku**, což je nežádoucí jev.
  - **Špatně** = A $\to$ A | ϵ (lze generovat až N způsoby), **Správně** = A $\to$ ϵ
- **Chomského normální forma** – gramatika může obsahovat pouze pravidla typu: **A** $\to$ **BC** nebo **A** $\to$ **a** nebo **S** $\to$ ϵ (pokud gramatika generuje pouze prázdný řetězec).
- **Nevypouštějící gramatika** – neobsahuje ϵ (*epsilon*) přechody.

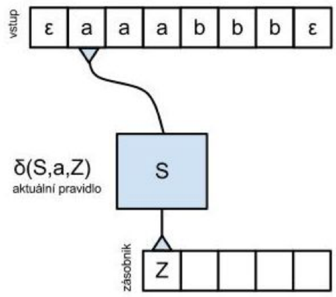
$A \rightarrow aBBb \mid AaA$
$B \rightarrow \varepsilon \mid bCA$
$C \rightarrow AB \mid a \mid b$

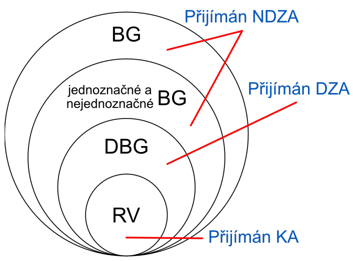


$A \Rightarrow aBBb \Rightarrow abCABb \Rightarrow abCaBBbBb \Rightarrow abCaBbBb \Rightarrow abbaBbBb \Rightarrow$
$abbaBbb \Rightarrow abbabb$

2.2 Zásobníkové automaty (ZA)

Slouží k **rozpoznání bezkontextových jazyků**. S využitím zásobníků si může pamatovat kolik a jaké znaky přečetl, což je potřeba právě k rozpoznání bezkontextového jazyka. Zásobníkový automat je v podstatě konečný automat rozšířený o zásobník.





- ZA na základě **aktuálního znaku** na pásce, **prvního znaku v zásobníku** a **aktuálního stavu** změní svůj stav a **přepíše** znak v zásobníku podle daných pravidel.
- ZA **přijímá** dané slovo, jestliže skončí v konfiguraci (q, ϵ, ϵ), tedy když se přečte celé vstupní slovo a zásobník je **prázdný**.
- **Konfigurace** je dána: aktuálním stavem, obsahem pásky a obsahem zásobníku.
- **Deterministický** – nesmí se objevit **stejná konfigurace vícekrát**.

2.2.1 Formální definice zásobníkového automatu

Zásobníkový automat M je definován jako šestice $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$, kde:

- Q je konečná neprázdná množina **stavů**.
- Σ je konečná neprázdná množina **vstupních symbolů** (vstupní abeceda).
- Γ (*velká gamma*) je konečná neprázdná množina **zásobníkových symbolů**.
- δ je **přechodová funkce** (konečná množina instrukcí), $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P_{\text{fin}}(Q \times \Gamma^*)$.

- q_0 je **počáteční stav**, $q_0 \in Q$.
- Z_0 je **počáteční zásobníkový symbol**, $Z_0 \in \Gamma$.

2.2.2 Definice instrukcí (pravidel) v ZA

Instrukce (sady instrukcí reprezentují přechodovou funkci δ) definují **chování automatu**:

$$(q, a, X) \to (q', \alpha), \text{ kde } a \in \Sigma. \tag{1}$$

Tato instrukce je aplikovatelná jen v situaci (neboli konfiguraci), kdy **řídicí jednotka** je ve stavu q, **čtecí hlava** na vstupní pásce čte symbol a a na vrcholu zásobníku je symbol X. Pokud je **instrukce aplikována**, vykoná se následující:

1. řídicí jednotka **přejde do stavu** q,
2. čtecí hlava na vstupní pásce se **posune o jedno políčko doprava**,
3. vrchní symbol v zásobníku se **odebere** (vymaže),
4. **na vrchol zásobníku se přidá** řetězec α tak, že jeho nejlevější symbol je aktuálním vrcholem zásobníku.

Pravidlo	**Akce (Z = zásobník)**	**Význam**
$\delta(q_1, a, X) \to (q_1, YX)$	**přidání** prvku do Z	na začátek zásobníku se vloží Y
$\delta(q_1, a, X) \to (q_1, Y)$	**přepsání** prvku v Z	první prvek zásobníku se přepíše na Y
$\delta(q_1, a, X) \to (q_1, \epsilon)$	**smazání** prvku ze Z	první prvek zásobníku se smaže neboli nahradí prázdným slovem ϵ
$\delta(q_1, a, X) \to (q_2, X)$	**změna stavu**	stav q_1 se změní na stav q_2
$\delta(q_1, a, X) \to \emptyset$	**pád** automatu	ukončení výpočtu, slovo nebylo přijato

2.3 Převod BG na zásobníkový automat

Využívá se tzv. metody shora-dolů, která obsahuje pouze **1 stav**:

1. pro všechny **neterminály** vypíšu pravidla typu: $(q, \epsilon, A) \to \{(q, B), (q, C)\}$,
2. všechny **terminály** přepíšu na pravidla typu: $(q, a, a) \to (q, \epsilon)$.

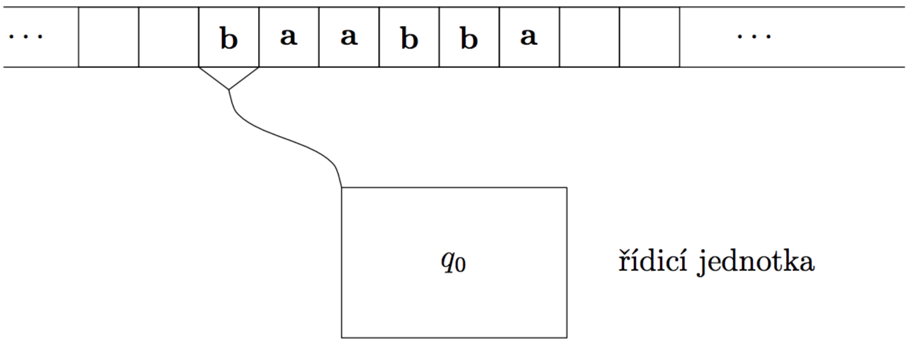
Vstupní gramatika:	**Instrukce, převedené dle výše uvedených pravidel:**
$S \to A \mid B$	$(Q, \epsilon, S) \to \{(q, A), (q, B)\}$
$A \to a$	$(Q, \epsilon, A) \to (q, a)$
$B \to (c)$	$(Q, \epsilon, B) \to (q, (c))$
$\Sigma = \{A, B, S\}$	$(Q, a, a) \to (q, \epsilon)$
$\Gamma = \{a, c, (,)\}$	$(Q, (, () \to (q, \epsilon)$
	$(Q, c, c) \to (q, \epsilon)$
	$(Q,),)) \to (q, \epsilon)$

3 Matematické modely algoritmů - Turingovy stroje a stroje RAM. Složitost algoritmu, asymptotické odhady. Algoritmicky nerozhodnutelné problémy.

Ve snaze **popsat jakýkoliv algoritmus** si vymysleli matematici Turingovy a RAM stroje. Jde o dva různé přístupy (modely) univerzálních počítačů/programovacích jazyků. Jinými slovy těmito stroji lze **definovat** a **provést libovolný algoritmus**.

Historicky prvním „univerzálním programovacím jazykem" byl Turingův stroj. Byl popsán dříve, ještě před rozmachem počítačů, proto se od reálného počítače (programování) podstatně liší, na rozdíl od RAM stroje. Turingův stroj například pracuje s **celou abecedou** zatímco RAM (podobně jako počítač) s **čísly**.

3.1 Turingův stroj (TS)



Turingův stroj je podobný konečnému automatu, ale má **oboustranně nekonečnou pásku** (je na ni zapsáno vstupní slovo), místo symbolu ϵ pro prázdné znaky se používá $\square$, **hlava** je **čtecí** i **zapisovací** a pohybuje se po pásce v **obou směrech**.

3.1.1 Formální definice TS

Turingův stroj, je definován jako šestice $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, kde:

- Q je konečná neprázdná množina **stavů**.
- Σ je konečná neprázdná množina **vstupních symbolů** (vstupní abeceda).
- Γ je konečná neprázdná množina **páskových symbolů**, kde $\Sigma \subseteq \Gamma$ a $\Gamma - \Sigma$ je (přinejmenším) speciální znak $\square$ (prázdný znak [Blank]).
- δ je přechodová funkce, $\delta : (Q - F) \times \Gamma \to Q \times \Gamma \times \{-1, 0, +1\}$.
- q_0 je **počáteční stav**, $q_0 \in Q$.
- F je množina **koncových stavů**, $F \subseteq Q$.

3.1.2 Definice instrukcí (pravidel) v TS

Podobně jako ZA lze konkrétní Turingův stroj zadat seznamem instrukcí. Tyto instrukce jsou opět dány přechodovou funkcí, význam instrukce: $(q, a) \to (q', a', m)$ je tento:

(akt. stav $[q]$, znak na pásce $[a]$) $\to$ (**nový stav** $[q']$, **nový znak** $[a']$, **posun** $[\{-1; 0; +1\}]$)

3.1.3 Příklad

Tento příklad invertuje slovo, které je uvedené na úvodním obrázku u TS:

$$
\begin{array}{ll}
Q = \{q_1, q_2\} & (q_1, a) \to (q_1, b, +) \\
\Sigma = \{a, b, c, \square\} & (q_1, b) \to (q_1, a, +) \\
q_0 = q_1 & (q_1, \square) \to (q_2, \square, 0) \\
F = \{q_2\} &
\end{array}
$$

3.1.4 Modifikace TS

- **N-páskový TS** – **čte** a **zapisuje** do **více pásek** najednou, jediná změna je v přechodové funkci: $\delta : Q \times \Gamma^n \to Q \times (\Gamma \times \{L, R, N\})^n$.
- **N-hlavový TS** – má více čtecích hlav než klasický TS, každá hlava zapisuje/čte a pohybuje se **nezávisle na ostatních**.
- **Nedeterministický TS** – umožňuje výběr z více možností, pro jednu konfiguraci můžeme definovat **více pravidel**.

3.1.5 Základní pojmy

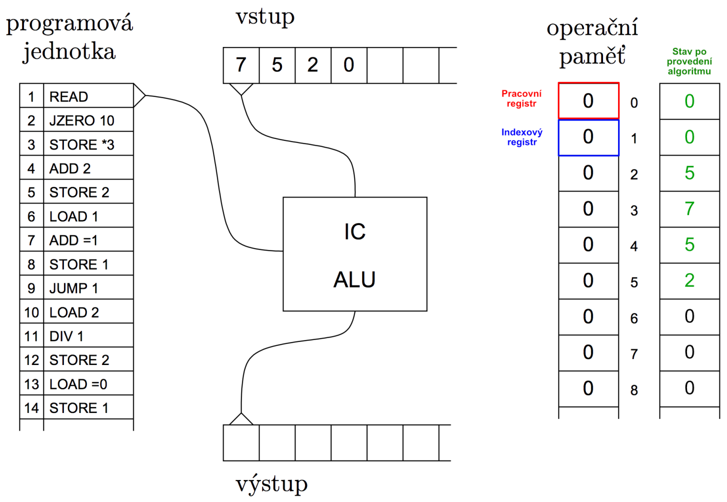
- **Turingovsky úplný** – stroj (počítač, programovací jazyk, úloha, ...), která má stejnou výpočetní sílu jako TS. Lze v něm **odsimulovat** libovolný jiný TS zadaný na vstupu.
- **Church-Turingova teze** – říká, že jakýkoliv výpočet lze úspěšně uskutečnit algoritmem běžícím na počítači, tedy „ke každému algoritmu existuje ekvivalentní TS”.

3.2 Model RAM (Random Access Machine)

RAM stroje již vycházejí ze skutečných počítačů, dá se tedy říct, že se jedná o jednoduchou abstrakci reálného procesoru s jeho strojovým kódem pracujícím s lineárně uspořádanou pamětí. Tento model slouží zejména k analýzy algoritmů z hlediska (**paměťové**, **časové**) **složitosti**. Skládá se z těchto částí:

1. **Programová jednotka** – uchovává program, tvořený konečnou posloupností instrukcí.
2. **Neomezená pracovní paměť** – neomezená lineárně uspořádaná paměť, tvořená buňkami, do který lze zapisovat/číst celá čísla ($\mathbb{Z}$), adresovaná přirozenými čísly ($\mathbb{N}$) ($0 =$ **pracovní** registr, $1 =$ **indexový** registr).

3. **Vstupní a výstupní páska** – lze na ně sekvenčně zapisovat/číst celá čísla ($\mathbb{Z}$).
4. **Centrální jednotka** – obsahuje programový register ukazující, která instrukce má být provedena. Ta se provede a programý registr se příslužně změní (zvýší o 1, o více v případě skoku).



Výše uvedený program vypočítá **aritmetický průměr**, který následně uloží do buňky paměti pod indexem č. **2**. Výsledek po dělení je roven $4,666$ a po zaokrouhlení 5.

3.2.1 Instrukce a typy operandů RAM

Typ	Hodnota operandu
$=i$	přímo číslo udané zápisem i
i	číslo obsažené v buňce s adresou i
$*i$	číslo v buňce s adresou $i+j$, kde j je aktuální obsah indexového registru

Zápis	Význam
`READ`	do pracovního registru (PR) se **načte** vstup a hlava se posune doprava
`WRITE`	na výstup se **zapíše** hodnota PR
`LOAD` *op*	do PR se **načte** hodnota dána operátorem *op*
`STORE` *op*	hodnota PR se **uloží** na do registru daného operátorem *op*
`ADD` *op*	k hodnotě PR se **přičte** hodnota daná operátorem *op*
`SUB` *op*	od hodnoty v PR se **odečte** hodnota daná operátorem *op*
`MUL` *op*	PR se **vynásobí** hodnotou danou operátorem *op*
`DIV` *op*	PR se **vydělí** hodnotou danou operátorem *op*
`JUMP` *návěští*	provede se **skok** na instrukci danou *návěštím*
`JZERO` *návěští*	pokud je hodnota v **PR rovna 0**, provede se skok na *návěští*
`JGTZ` *návěští*	pokud je hodnota v **PR větší než 0**, provede se skok na *návěští*
`HALT`	**korektní ukončení** programu

3.3 Složitost algoritmů

Abychom mohli **porovnávat** různé algoritmy řešící stejný problém, zavádí se pojem složitost algoritmu. Složitost je jinak řečeno **náročnost algoritmu** – čím menší složitost tím je algoritmus lepší. Přičemž nás může zajímat složitost z pohledu **času**, či **paměti**:

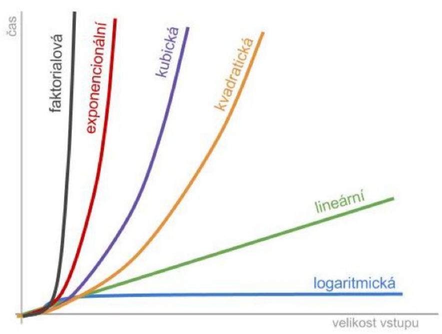
- **Časová složitost** – sleduje jak závisí **doba** výpočtu alg. na množství vstupních dat.
- **Prostorová složitost** – sleduje jak závisí **množství použité paměti** výpočtu alg. na množství vstupních dat.

Jelikož konkrétní čísla (čas, bity) se liší v **závislosti vstupních datech**, množství zpracovávaných dat a použitém programovacím jazyku, neudává se složitost čísly, nýbrž **funkcí závislou na velikosti vstupních dat**. Tato funkce se získá počítáním proběhlých instrukcí algoritmu sestaveném v univerzálním RAM stroji. A počítá se s nejhorším možným případem vstupu. To je důležité například u třídících algoritmů, kde hraje velkou roli to, jak moc už je vstupní pole setříděné (vstupuje-li do algoritmu už setříděná posloupnost čísel, algoritmus skončí okamžitě, zatímco s opačně seřazenými čísly se bude trápit dlouho.

3.4 Asymptotická notace

Je **způsob klasifikace počítačových algoritmů**. Ve většině případů nemusíme znát přesný počet provedených instrukcí a spokojíme se pouze s odhadem toho, jak rychle tento počet narůstá se zvyšujícím se vstupem. Asymptotická notace nám umožní **zanedbat méně důležité detaily** a **odhadnout** přibližně, **jak rychle daná funkce roste**. V souvislosti s asymptotickými odhady složitosti se používjí tyto zapisy:

- $f \in O(f)$ – f roste **nejvýše tak rychle** jako g (f je ohraničena g **shora**) [$\leq$].
- $f \in o(g)$ – f roste **(striktně) pomaleji** než g (f je ohraničena g **shora ostře**) [$<$].
- $f \in \Theta(g)$ – f roste **stejně rychle** jako g [$=$].
- $f \in \omega(g)$ – f roste **(striktně) rychleji** než g (f je ohraničena g **zdola ostře**) [$>$].
- $f \in \Omega(g)$ – f roste **rychleji** než g (f je ohraničena g **zdola**) [$\geq$].



Seřazeno podle složitosti:

- $f(n) \in \Omega(logn)$ – logaritmická funkce (složitost),
- $f(n) \in \Omega(n)$ – lineární funkce (složitost),
- $f(n) \in \Omega(n^2)$– kvadratická funkce (složitost),
- $f(n) \in O(n^k)$ pro nějaké $k > 0$ – polynomiální,
- $f(n) \in \Omega(k^n)$ pro nějaké $k > 1$ – exponenciální.

3.4.1 Úskalí asymptotické notace

Při používání asymptotických odhadů časové složitosti je třeba si uvědomit některá úskalí:

- Asymptotické odhady se týkají pouze toho, **jak roste čas s rostoucí velikostí vstupu** $\rightarrow$ neříkají nic o **konkrétní době výpočtu**. V asymptotické notaci mohou být **skryty velké konstanty**.
- Algoritmus, který má lepší asymptotickou časovou složitost než nějaký jiný algoritmus, **může být ve skutečnosti rychlejší** až pro nějaké hodně velké vstupy.
- Většinou analyzujeme složitost v **nejhorším případě**. Pro některé algoritmy může být doba výpočtu v nejhorším případě mnohem větší než doba výpočtu na „typických" instancích (typicky Quicksort $\rightarrow$ nejhorší: $O(n^2)$, průměrná: $O(n \log n)$).

3.5 Algoritmicky nerozhodnutelné problémy

Rozhodovací problém je rozhodnutelný (řešitelný) pokud pro libovolný vstup z množiny vstupů, skončí algoritmus svůj výpočet a vydá správný výstup (tedy jestliže **existuje turingův stroj, který jej řeší**).

Pokud nalezneme takový vstup, pro který všechny dosavadní algoritmy nejsou schopny nalézt výstup, můžeme tento problém označit za **nerozhodnutelný**. Speciální případ jsou **doplňkové problémy**, které vracejí přesně opačné výsledky než původní problém.

3.5.1 Definice problému

Problém je určen **trojicí** (IN, OUT, p), kde:

- IN je množina (přípustných) **vstupů**,
- OUT je množina **výstupů**,
- $p : IN \rightarrow OUT$ je **funkce** přiřazující každému vstupu odpovídající výstup.

3.5.2 Ano/Ne problémy

Jsou to problémy, jejichž **výstupní množina obsahuje dva prvky** $OUT = \{\text{ano}, \text{ne}\}$. Na ano/ne problémy se dají převést ostatní problémy nepotřebujeme-li znát přesný výsledek:

- Nepotřebuji najít v poli nejmenší číslo, stačí mi vědět **zda pole obsahuje číslo menší než nula**.
- Nepotřebuji znát nejkratší cestu grafem, stačí mi najít cestu, **která je kratší než 8**.

3.5.3 Riceova věta

Tato věta ukazuje nerozhodnutelnost celé třídy problémů, její znění je následující „*Každá netriviální vstupně/výstupní (I/O) vlastnost programů je **nerozhodnutelná***".

- Vlastnost X **je vstupně/výstupní** právě tehdy, když každé dva programy se stejnou I/O tabulkou buď oba vlastnost X mají nebo ji oba nemají.
- Připomeňme tedy ještě, že vlastnost V je **triviální**, když ji mají buď všechny programy nebo ji nemá žádný program; taková vlastnost je podle definice také **vstupně/výstupní**.

Problém	**1**	**2**	**3**
Je triviální?	A	N	N
Je I/O?	A	A	N
Je nerozhodnutelný	N	A	N

3.5.4 Částečná rozhodnutelnost

Částečně rozhodnutelný problém, je takový problém, pro který jsme v případě vstupů, u nichž očekáváme odpověď ANO, **schopni vrátit odpověď ANO**, a v případě NE vrátit buď NE nebo $\perp$ (program se nezastaví a nejsme schopni zjistit, zda by odpověď byla opravdu NE).

3.5.5 Převeditelnost mezi nerozhodnutelnými problémy

Důkaz neřešitelnosti lze provést skrze jiné, **už dokázané**, problémy. Řekneme, že problém P_1 je převeditelný na problém P_2 (značíme $P_1 \rightsquigarrow P_2$), jestliže alg., který k instanci I_1 problému P_1 sestrojí instanci I_2 problému P_2 tak, **že odpověď** P_1, I_1 **je stejná jako** P_2, I_2. Např.: DHP je převeditelný na HP. Z toho vyplývá, že pokud P_1 je nerozhodnutelný tak i P_2 je **nerozhodnutelný**.

3.5.6 Příklady nerozhodnutelných problémů

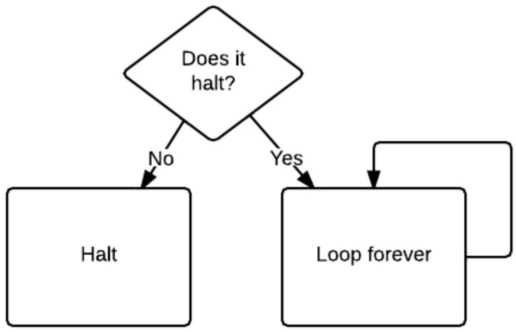
1. **Eq-CFG (Ekvivalence bezkontextových gramatik)**
   - **Vstup**: Dvě bezkontextové gramatiky $G1, G2$.
   - **Otázka**: Platí $L(G1) = L(G2)$? Generují obě gramatiky stejný jazyk?
2. **HP (Problém zastavení [Halting Problem])**
   - **Vstup**: Turingův stroj M a jeho vstup w.
   - **Otázka**: Zastaví se M na w (tzn. je výpočet stroje M pro vstupní slovo w konečný)?

3.6 Optimalizační problémy

Optimalizační problémy **hledají nejlepší řešení** v množině různých řešení. Příkladem je například: hledání nejkratší cesty, nejmenší kostry, apod.

3.6.1 Příklady optimalizačních problémů

1. **Hledání nejkratší cesty v grafu**
   - **Vstup**: Orientovaný graf $G = (V, E)$ a dvojice vrcholů $u, v \in V$.
   - **Výstup**: Nejkratší cesta z u do v.
2. **Hledání minimální kostry v grafu**
   - **Vstup**: Neorientovaný souvislý graf $G = (V_G, E_G)$ s ohodnocenými hranami.
   - **Výstup**: Souvislý graf $H = (V_H, E_H)$, kde $V_H = V_G$ a $E_H \subseteq E_G$, který má součet hodnot všech hran minimální.



4 Třídy složitosti problémů. Třída PTIME a NPTIME, NP-úplné problémy.

4.1 PTIME (PSPACE)

Třída všech problémů, které lze řešit algoritmy s **polynomiální časovou (prostorovou) složitosti**, tj. s časovou složitosti $O(n^k)$, kde k je nějaká konstanta.

Tuto třídu problému považujeme za zvládnutelnou. Je **robustní** – mezi jednotlivými výpočetními modely (RAM, TS) existují vzájemné polynominální simulace, nezáleží tedy jakým modelem budeme algoritmus simulovat, vždy bude patřit do třídy PTIME.

4.1.1 Problémy pařící do třídy PTIME

- Třídění a vyhledávání.
- Nejkratší cesta v grafu a minimální kostra grafu.
- Ekvivalence deterministických konečných automatů.
- Přijatelnost slova bezkontextovou gramatikou.

1. **Výběr aktivit**
   - **Vstup**: Množina aktivit s časovými intervaly, kdy je lze vykonávat.
   - **Výstup**: Největší možný počet kompatibilních aktivit (aktivit, které se nekryjí).
2. **Optimalizace násobení řetězce matic**
   - **Vstup**: Posloupnost matic.
   - **Výstup**: Plně uzávorkovaný součin.
3. **LCS - problém nejdelší společné posloupnosti**
   - **Vstup**: Dvě posloupnosti v, w v nějaké abecedě Σ.
   - **Výstup**: Nejdelší společná podposloupnost posloupností v, w.

4.2 NPTIME

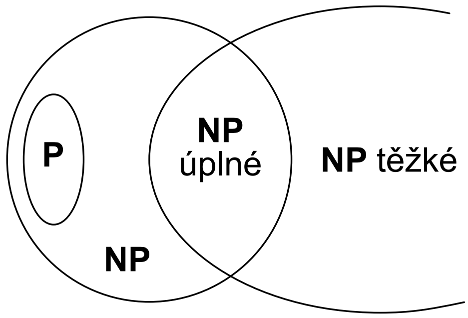
Třída všech rozhodovacích problémů (ano/ne), které jsou rozhodovány **nedeterministickými algoritmy s polynomiální časovou složitosti**. Tedy problémy, u kterých je možné pro daný vstup vždy ověřit zda je odpověď ANO, v případě NE tomu tak být nemusí.

Při odpovědi „ANO" je výsledek vždy správný (našel se alespoň jeden výpočet pro který je odpověď ANO) zatímco odpověď „NE" nemusí být vždy pravdivá a to z důvodu, že by na výstupu muselo být vždy NE (algoritmus by musel otestovat všechny možnosti). Pointou nedeterministických algoritmů je to že **náhodně nastřelují nějaké řešení a ověřují jejich správnost**.

4.3 Třída NP-úplných problémů

NP-úplné problémy jsou takové problémy, na které jsou **polynomiálně redukovatelné všechny ostatní problémy z třídy NP**. To znamená, že třídu NP-úplných úloh tvoří v jistém smyslu ty **nejtěžší úlohy** z NP.

Pokud by byl nalezen deterministický polynomiální algoritmus pro nějakou NP-úplnou úlohu, znamenalo by to, že všechny nedeterministicky polynomiální problémy jsou řešitelné v polynomiálním čase (tedy že NP = P). Otázka, zda nějaký takový algoritmus existuje, zatím nebyla rozhodnuta, předpokládá se však, že NP $\neq$ P (**je však zřejmé, že P $\subseteq$ NP**).



4.3.1 NP-těžký problém

Problém P nazveme NP-těžkým, pokud pro **libovolný problém** A ze třídy NP platí, že A je **polynominálně převeditelné** (redukovatelné) na problém P, tedy pokud platí: $\forall P \in$ NPTIME : $P \triangleright Q$.

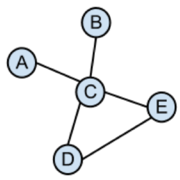
4.3.2 NP-úplný problém

Třída nejtěžších NPTIME problémů. Problém P je NP-úplný, pokud patří do třídy **NP** a je **NP-těžký**.

4.3.3 NP-úplné problémy

1. **IS (problém nezávislé množiny)**

   - **Vstup**: Neorientovaný graf G (o n vrcholech), číslo k ($k \leq n$).
   - **Otázka**: Existuje v G nezávislá množina velikosti k (tj. množina k vrcholů, z nichž žádné dva nejsou spojeny hranou)?

   Program vybere náhodně k vrcholů z grafu a ověří zdali nejsou některé spojeny hranou. Když mezi nimi hranu **nenajde**, vrátí odpověď *„ANO, v grafu existuje* *nezávislá množina o velikosti k"*. Když hranu mezi zvolenou množinou **najde**, vrátí odpověď „NE". Přičemž odpověď ne **může být chybná**.

Třeba pro graf ABCDE na obrázku níže a $k = 3$, vybere algoritmus vrcholy $\{A, B, C\}$, které závislé jsou a vrátí chybnou odpověď „NE". Když to ale algoritmus provede vícekrát, pro různé vrcholy, **pravděpodobnost správnosti odpovědi se zvyšuje**. A o tom to je. Nepotřebujeme znát 100% správnou odpověď, ale chceme se dočkat alespoň nějaké odpovědi.

Výstup pro $k > 3$: NE.
Výstup pro $k = 3$: ANO (ABD, ABE).
Výstup pro $k = 2$: ANO (AB, AD, AE, BD, BE).

2. **Isomorfismus grafů**
   - **VSTUP**: Dva neorientované grafy G a H.
   - **OTÁZKA**: Jsou grafy G a H izomorfní?
3. **CG (Barvení grafu)**
   - **VSTUP**: Neorientovaný graf G a číslo k.
   - **OTÁZKA**: Je možné graf G obarvit k barvami (tj. existuje přiřazení barev vrcholům tak, aby žádné dva sousední vrcholy nebyly obarveny stejnou barvou)?
4. **SAT (problém splnitelnosti booleovských formulí)**
   - **VSTUP**: Booleovská formule v konjunktivní normální formě.
   - **OTÁZKA**: Je daná formule splnitelná (tj. existuje pravdivostní ohodnocení proměnných, při kterém je formule pravdivá)?
5. **3-SAT (problém SAT s omezením na 3 literály)**
   - **VSTUP**: Formule v konjunktivní normální formě, kde každá klauzule obsahuje právě 3 lateraly.
   - **OTÁZKA**: Je formule splnitelná?
6. **HK (problém hamiltonovské kružnice)/HC (problém hamiltonovskho cyklu)**
   - **VSTUP**: Neorientovaný graf G/Orientovaný grav G.
   - **OTÁZKA**: Existuje v G hamiltonovská kružnice (uzavřená cesta, procházející každým vrcholem právě jednou)?
7. **Subset-Sum**
   - **VSTUP**: Množina přirozených čísel $M = x_1, x_2, \ldots, x_n$ a přirozené číslo s.

- **Otázka**: Existuje podmnožina množiny M, pro niž součet jejích prvků je roven s?

8. **Problém obchodního cestujícího (TSP) ANO/NE verze**
   - **Vstup**: Neorientovaný graf G s hranami ohodnocenými přirozenými čísly a číslo k.
   - **Otázka**: lze objet k měst a neujet víc než danou vzdálenost?

9. **Vrcholové pokryti (vertex cover)**
   - **Vstup**: Neorientovaný graf G a přirozené číslo k.
   - **Otázka**: Existuje v grafu G množina vrcholů velikosti k taková, že každá hrana má alespoň jeden svůj vrchol v teto množině?

5 Jazyk predikátové logiky prvního řádu. Práce s kvantifikátory a ekvivalentní transformace formulí.

Predikátová logika (PL) pracuje s primitivními formulemi (**predikáty**) vypovídajícími o **vlastnostech** a **vztazích** mezi **předměty** jistého **univerza** (individui). Je rozšířením výrokové logiky. Na rozdíl od výrokové logiky si všímá i struktury vět samotných a obsahuje predikáty a kvantifikátory. Pouze jen malá část úsudku může být formalizována pomocí výrokové logiky:

Všechny opice mají rády banány
Judy je opice

Judy má ráda banány

Z hlediska VL jsou to jednoduché výroky p, q, r a z p, q nevyplývá r.

5.1 Predikátová logika 1. řádu

Predikátová logika umožňuje uvažování nad **vlastnostmi**, jež jsou **sdíleny mnoha objekty**, díky použití **proměnných** a **kvantifikátorů**. V predikátové logice by byl výše uvedený úsudek formalizován takto:

Každé individuum, je-li **O**pice pak má rádo **B**anány.
Judy je individuum s vlastností být **O**pice.

Judy je individuum s vlastností mít rádo **B**anány.

$\forall x(O(x) \rightarrow B(x)); O(J) \neq B(J)$, kde x je **individuová proměnná**; O, B **predikátové symboly** a J **funkční symbol**.

Poznámka Pokud bychom chtěli formalizovat úsudky, které navíc vypovídají i o vlastnostech vlastností a vztahů a o vztazích mezi vlastnostmi a vztahy, museli bychom použít predikátovou logiku druhého řádu a vyššího. Tou se ale nebudeme zabývat.

5.1.1 Formální jazyk PL1 – Abeceda

Logické symboly:

- Individuové proměnné: $x, y, z, \ldots$
- Logické spojky: $\wedge$ konjunkce, $\vee$ disjunkce, $\rightarrow$ implikace, $\leftrightarrow$ ekvivalence, $\neg$ negace.
- Kvantifikační symboly: $\forall, \exists$.

Speciální symboly: (n-arita = počet argumentů)

- Predikátové: $P^n, Q^n, \ldots$
- Funkční: $f^n, g^n, h^n, \ldots$

Pomocné symboly: závorky a jiná interpunkční znaménka (,), …

5.1.2 Formální jazyk PL1 – Gramatika

Termy:

1. každý symbol proměnné $x, y, \ldots$ je **term**,
2. jsou-li $t_1, \ldots, t_n$ $(n \geq 0)$ termy a je-li f n-ární **funkční symbol**, pak výraz $f(t_1, \ldots, t_n)$ je term; pro $n = 0$ se jedná o **individuovou konstantu** (značíme $a, b, c, \ldots$),
3. jen výrazy dle 1. a 2. jsou termy.

Atomické formule:

- je-li P n-ární predikátový symbol a jsou-li $t_1, \ldots, t_n$ termy, pak výraz $P(t_1, \ldots, t_n)$ je **atomická formule** (na vstupu jsou pouze termy).

Formule:

- každá atomická formule je formule,
- je-li výraz A formule, pak $\neg A$ je formule,
- jsou-li výrazy A a B formule, pak výrazy $(A \wedge B), (A \vee B), (A \rightarrow B), (A \leftrightarrow B)$ jsou formule, je-li x proměnná a A formule, pak výrazy $\forall x A$ a $\exists x A$ jsou formule.

5.2 Převod z přirozeného jazyka do PL1

- $\forall$ – „všichni", „žádný", „nikdo", ...
- $\exists$ – „někdo", „něco", „někteří", „existuje", ...

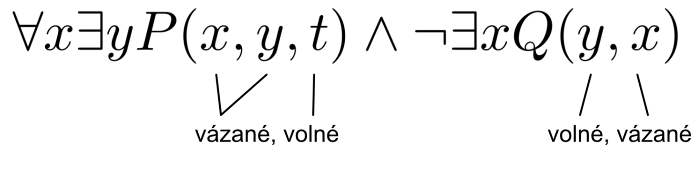
Větu musíme často ekvivalentně přeformulovat, pozor: v češtině **dvojí zápor**!

- Žádný student není důchodce: $\forall x(S(x) \rightarrow \neg D(x))$.
- Ale, „všichni studenti nejsou důchodci" čteme jako „ne všichni studenti jsou důchodci": $\neg\forall x(S(x) \rightarrow D(x)) \leftrightarrow x(S(x) \rightarrow \neg D(x))$

Jako **pomůcka** k řešení může sloužit tato zásada:

- Po **všeobecném** kvantifikátoru následuje formule ve tvaru implikace: $\forall \ldots \rightarrow$.
- Po **existenčním** kvantifikátoru formule ve tvaru konjunkce: $\exists \ldots \wedge$.

5.2.1 Volné a vázané proměnné



5.3 Ekvivalentní úpravy

Při ekvivalentních úpravách se používají **de Morganovy** zákony v PL1: $\neg\forall x A \leftrightarrow \exists x \neg A$ $\neg\exists x A \leftrightarrow \forall x \neg A$.

Příklady

- Není pravda, že všichni vodníci jsou zelení. $\leftrightarrow$ Někteří vodníci nejsou zelení.
  $\neg\forall x(V(x) \to Z(x)) \leftrightarrow \exists x(V(x) \wedge \neg Z(x))$
- Není pravda, že někteří vodníci jsou zelení. $\leftrightarrow$ Žádný vodník není zelený.
  $\neg\exists x(V(x) \wedge Z(x)) \leftrightarrow \forall x(V(x) \to \neg Z(x))$
- Everybody loves somebody sometimes.
  $\forall x \forall y \forall z L(x, y, z)$
- Marie má ráda pouze vítěze.
  $\forall x(R(m, x) \to V(x))$

5.4 Ekvivalentní transformace

- Aplikace negace: $\neg\forall x[V(x) \to Z(x)] \quad \leftrightarrow \quad \exists x[V(x) \wedge \neg Z(x)]$.
- **De morganovy** zákony: $\forall x[((P(x) \wedge Q(x)) \vee D(x)] \quad \leftrightarrow \quad \forall x[(P(x) \vee D(x)) \wedge (Q(x) \vee D(x))]$.

  $$\neg(A \vee B) \iff (\neg A) \wedge (\neg B)$$
  $$\neg(A \wedge B) \iff (\neg A) \vee (\neg B)$$

- Převod **implikace**: $\forall x(P(x) \to G(x)) \quad \leftrightarrow \quad \forall x(\neg P(x) \vee G(x))$.

Patří zde i část z převodu do **Skolemovy Klauzární formy**:

1. eliminace nadbytečných kvantifikátorů,
2. eliminace spojek $\to, \leftrightarrow$,
3. přesun negace dovnitř,
4. přejmenování proměnných,
5. přesun kvantifikátorů doprava,
6. přesun všeobecných kvantifikátorů doleva,
7. použití distributivních zákonů.

5.5 Sémantika v PL1

- Při substituci termů za proměnné (A(x/t)), je třeba dbát na nahrazování pouze **volných proměnných**.
- Při definici formule, je nutné vysvětlit co **znamenají** jednotlivé predikátové symboly, termy atd.

6 Pojem relace, operace s relacemi, vlastnosti relací. Typy binárních relací. Relace ekvivalence a relace uspořádání.

6.1 Relace

- **N-ární relace** nad množinami $A_1, \ldots, A_n$ je **libovolná podmnožina kartézského** součinu $A_1 \times \ldots \times A_n$ (tyto množiny jsou **nosičemi** relace).
- **Kartézský součin** množin$A$ a B, označovaný $A \times B$, je množina všech uspořádaných dvojic, kde první prvek z dvojice patří do množiny A a druhý do množiny$B$. Příklad: $\{a, b\} \times \{a, b, c\} = \{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c)\}$.

6.1.1 Typy relací

- **homogenní** – jediný druh nosiče ($A \times A$),
- **heterogenní** – alespoň dva různé druhy nosiče ($A \times B$),
- **unární** (n = 1), **binární** (n = 2), **ternární** (n = 3), **n-ární** – podle arity,
- **triviální** – **úplná** ($\rho = A_1 \times A_n$), **prázdná** ($\rho = \emptyset$),
- **netriviální** – $\emptyset \subset \rho \subset A_1 \times \ldots \times A_n$.

6.1.2 Vlastnosti relací

Binární relace $R \subseteq A \times A$ je:

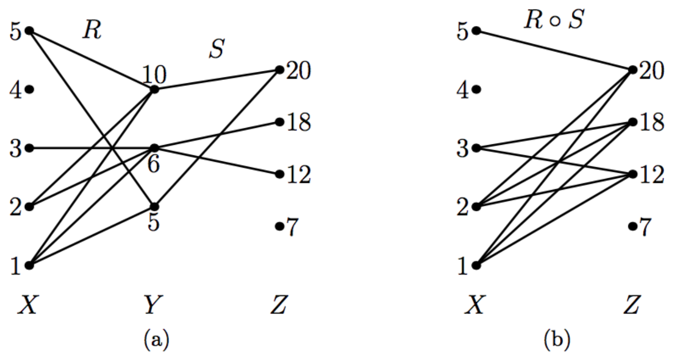
- **Reflexivní** – $\forall x \in A : (x, x) \in R$.
- **Ireflexivní** – $\forall x \in A : (x, x) \notin R$.
- **Symetrická** – $\forall x \in A : (x, y) \in R \Rightarrow (y, x) \in R$.
- **Asymetrická** – $\forall x \in A : (x, y) \in R \Rightarrow (y, x) \notin R$.
- **Antisymetrická** – $\forall x \in A : (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y$.
- **Tranzitivní** – $\forall x \in A : (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$.

Příklad

- Relace „=" na $\mathbb{N}$ je reflexivní, symetrická, antisymetrická a tranzitivní, ale není ireflexivní ani asymetrická.
- Relace „$\leq$" na $\mathbb{N}$ je reflexivní, antisymetrická a tranzitivní, ale není ireflexivní, symetrická ani asymetrická.
- Relace „$<$" na $\mathbb{N}$ je ireflexivní, asymetrická, antisymetrická a tranzitivní, ale není reflexivní ani symetrická.

6.2 Operace s relacemi

- **Průnik** – Prvek x náleží do průniku relací $R1 \cap R2$, pokud patří do množiny $R1(x \in R1)$ **a zároveň** do $R2(x \in R1)$.
- **Sjednocení** – Prvek x náleží do sjednocení relací $R1 \cup R2$, pokud patří do množiny $R1(x \in R1)$ **nebo** $R2(x \in R1)$.
- **Doplněk** – Doplňkem $R1'$ k relaci $R1$ rozumíme **všechny prvky které nepatří** do $R1$.
- **Inverze** – Relace $R^{-1} \subseteq B \times A$ je inverzní k relaci $R \subseteq A \times B$, pokud $xR^{-1}y \Leftrightarrow yRx$.
- **Skládání relací** – Výsledkem je množina dvojic, kde pokud existují dvojice $(a, b) \in R$ a $(b, c) \in S$, pak jejich složení $(a, c) \in R \circ S$.



Obrázek 1.2: (a) Relace R a S, (b) jejich složení.

6.3 Typy binárních relací

Mezi nejznámější typy binárních relací patří **ekvivalenece** $(=)$ [Re, Sy, Tr], **uspořádání** $(<, >, \leq, \geq)$ [Re, An, Tr] a **tolerance** [Re, Sy].

6.3.1 Ekvivalence [Re, Sy, Tr]

Relace ekvivalence představuje jakési zjemnění relace rovnosti. Vždy můžeme rozhodnout, že jsou dva prvky množiny stejné, tj. že a = a. Ale někdy se nám hodí zjistit, zda jsou si dva prvky **pouze podobné**, ne nutně stejné. Neboli zda mají stejnou nějakou zásadní vlastnost. Například dvě knihy můžeme považovat za podobné, pokud mají stejný žánr nebo **pomocí ekvivalence**: dvě knihy jsou ekvivalentní pokud mají stejný žánr.

- Binární relace na množině X je ekvivalentní, pokud je R na A: **reflexivní**, **symetrická** a **tranzitivní**.
- **Třída ekvivalence** prvku a je množina všech prvků ekvivalentních s daným prvkem a.
- **Průnik** dvou ekvivalencí je zase ekvivalence.
- **Sjednocení** dvou ekvivalencí nemusí znamenat, že výsledek bude ekvivalentní.

6.3.2 Uspořádání [Re, An, Tr]

- Binární relace na množině X je **neostrým uspořádání**, pokud je R na A: **reflexivní**, **antisymetrická** a **tranzitivní**.
- Binární relace na množině X je **ostrým uspořádání**, pokud je R na A: **ireflexivní**, **antisymetrická** a **tranzitivní**.
- Uspořádání je **úplné** pokud neexistují neporovnatelné prvky.

7 Pojem operace a obecný pojem algebra. Algebry s jednou a dvěma binárními operacemi.

7.1 Algebra

Algebra je naukou o **algebraických strukturách**, tedy **množinách**, na nichž jsou zavedeny nějaké **operace**. Slouží pro popis objektů reálného světa a operací prováděných s těmito objekty. Příklady algeber:

- $(\mathbb{N}, \{+^2\})$ - sčítání nad množinou přirozených čísel,
- $(2^M, \{\cup, \cap\})$ - množina všech podmnožin M s operací průnik a sjednocení.

7.1.1 Definice

Každý objekt algebry je reprezentován **datovým nosičem** (množina popisující data, se kterými pracujeme). A **operacemi** – nejjednoduššími transformacemi, které nad daty můžeme realizovat. **Algebraická struktura** je definována jako $(A, \circ)$, kde:

- A – nosič algebry (množina objektů – čísel, proměnných, ...),
- $\circ$ – množina operací nad nosičem X.

7.2 Operace

Operace na množině A je definována jako zobrazení

$$f : A^n \to A, \tag{2}$$

tedy zobrazení, které každé n-tici prvků množiny A, jednoznačně přiřazuje prvek z množiny A.Číslo n nazýváme **arita operace** a podle něj operace označujeme jako **nulární** (n = 0), **unární** (n = 1), **binární** (n = 2), **ternární** (n = 3).

7.3 Algebraické struktury s jednou binární operací

Definována jako $(A, \circ)$ s jedním nosičem (A) a jednou homogenní binární operací $(\circ)$. Nejprve je nutné zmínit **vlastnosti binárních operací**:

- **asociativita**: $a * (b * c) = (a * b) * c$,
- **komutativita**: $a * b = b * a$.

Kromě již zmíněné asociativity a komutativity algebraické struktury také zavádí existenci:

- **Jednotkového prvku**: e takové, že $\forall x \in X : x \circ e = e \circ x = x$. Tedy prvek, který nezmění výsledek (1 u násobení, 0 u sčítání).
- **Inverzního prvku**: $\overline{x}$ takové, že $\forall x \in X : x \circ \overline{x} = \overline{x} \circ x = e$. Tedy prvek, který převede výsledek na jednotkový prvek.

neutrálního či **inverzního prvku**, a další charakteristiky.

7.3.1 Klasifikace algebraických struktur

Všechny níže uvedené klasifikace algebraické struktury $(A, \circ)$ zahrnují i ty co jsou pod nimi. Tedy pokud je nějaká algebraická struktura (AS) Monoid, je i Pologrupa a Grupoid.

1. **Grupoid** – **uzavřenost** (univerzalita) na nosiči (po výpočtu je výsledek stále v množině A).
2. **Pologrupa** – splňuje vlastnost **asociativity**.
3. **Monoid** – existence **jednotkového prvku**.
4. **Grupa** – existence **inverzního prvku**.
5. **Abelova grupa** – splňuje vlastnost **komutativity** (symetrická podle diagonály).

Kongruence – označuje ekvivalenci na algebře, která je slučitelná se všemi operacemi na algebře.

7.3.2 Morfismy

- **Homomorfismus** – zobrazení, které převádí jednu algebraickou strukturu na jinou: $f(a_1 \cdot a_2) \to f(a_1) \circ f(a_2)$.
- **Izomorfismus** – bijektivní homomorfismus.
- **Epimorfismus** – subjektivní homomorfismus.
- **Monomorfismus** – injektivní homomorfismus.
- **Endomorfismus** – homomorfismus z objektu do sebe sama (stejná množina).
- **Automorfismus** – endorfismus, který je izomorfní.

7.4 Okruhy (Algebraické struktury s dvěma binárními operací)

Okruh je algebraický systém $(A, +, \cdot)$ se dvěma základními binárními operacemi, kde první $(A, +)$ je **abelova grupa** a druhá $(A, \cdot)$ je alespoň **pologrupa**. Podobně jako u předchozí AS, i zde se zavádí nový pojem:

- **Existence dělitele nuly** – říká, že ve struktuře existují 2 nenulové prvky, pro něž platí $a \circ b = 0$.

U všech typů okruhů musí být splněna podmínka první struktury, která musí být **abelova grupa**, a druhá musí být:

- **Okruh** - uzavřená (U), asociativní (A) [pologrupa].
- **Unitární okruh** - U, A, existence jednotkového prvku (J) [monoid].
- **Obor Integrity** - U, A, J [monoid] + **nesmí** obsahovat dělitele nuly.
- **Těleso** - U, A, J a existence inverzního prvku (I) [grupa] + **nesmí** obsahovat dělitele nuly.
- **Pole** - U, A, J, I a komutativita [grupa] + **nesmí** obsahovat dělitele nuly.

8 FCA – formální kontext, formální koncept, konceptuální svazy. Asociační pravidla, hledání často se opakujících množin položek.

8.1 Formální konceptuální analýza (FCA)

Metoda analýzy tabulkových dat (objektů a jejich vlastností), umožňuje jiný pohled na data (využívá se např. u data miningu). **Vstupem** pro FCA jsou **tabulková data**, která jsou uspořádána následovně: **objekty** (řádky) a **atributy** (sloupce). Tyto tabulková data vytváří tzv. **kontexty**.

	červené	**bílé**
jablko	×	
zelí	×	×

8.2 Formální Kontext

Formální kontext K obsahuje objekty z množiny O a atributy z množiny A. Vztahy mezi objekty a atributy jsou charakterizovány binární relací R. Obecně se pro popis kontextu používá výraz:

$$K = (O, A, I). \tag{3}$$

Takto vymezený formální kontext je dobře zobrazitelný tabulkou, ve které jsou **řádky** obsazeny **objekty**, **sloupce atributy** a incidenční data ($I \subseteq O \times A$) vyjadřují relaci R (I je relace incidence).

8.2.1 Galoisovy konexe

Umožňují přecházet z množiny objektů na jejich společné atributy (↑ **intent**) a naopak (↓ **extent**).

- **Intent** ↑ – $2^X \to 2^Y; A \subseteq X; A^\uparrow = \{y \in Y; \forall x \in A(x, y) \in I\}$ [*z objektu na atributy*].
- **Extent** ↓ – $2^Y \to 2^Y; B \subseteq Y; B^\downarrow = \{x \in X; \forall y \in B(x, y) \in I\}$ [*z atributů na objekt*].

Příklad

$A_1 = \{$jablka, zelí$\}, A_1^\uparrow = \{$červené$\}$ $A_2 = \{$zelí$\}, A_2^\uparrow = \{$červené, bílé$\}$
$B_1 = \{$červené, bílé$\}, B_1^\downarrow = \{$zelí$\}$ $B_2 = \{$červené$\}, B_2^\downarrow = \{$zelí, jablko$\}$

8.3 Formální koncept

Formální koncept je dvojice (A, B), kde A je množina objektů, a B je množina atributů, které jsou **společné pro všechny objekty** z množiny A. Koncept (A, B) je tedy dán jako: $(A, B) \Leftrightarrow A = B^\downarrow \wedge B = A^\uparrow$, kde $A \subseteq X$, $B \subseteq Y$, kde X a Y jsou objekty a atributy výše uvedené tabulky.

8.3.1 Uzávěrový operátor $\uparrow\downarrow$

Jak již znační $A^{\uparrow\downarrow} = C(A)$ vypovídá, k určení uzávěru atributů množiny A se nejprve provede **intent** a poté **extent**. Uzávěr má tyto vlastnosti:

1. **Idempotence** – $C(C(A)) = C(A)$.
2. **Extensionalita** – $A \subseteq C(A)$.
3. **Monotonie** – $A_1 \subseteq A_2 \Rightarrow C(A_1) \subseteq C(A_2)$.

8.4 Konceptuální svazy

Uspořádaná množina konceptů tvoří tzv. konceptuální svaz. Ten lze graficky znázornit **Hasseovým diagramem** – každý vrchol grafu reprezentuje **jeden koncept**. Koncepty K_i a K_j jsou v grafu spojeny, pokud $K_i \leq K_j$, přičemž K_j je umístěn výše než K_i a neexistuje takové K_k, že $K_i \leq K_k \leq K_j$.

8.4.1 Návod k vytvoření konceptuálního svazu

1. Vytvořím a postupně si zapíšu množinu všech extentů na jednotlivých atributech (v grafu zapisuji **zdola nahoru**).
2. Vytvořím jedinečné průniky extentů.
3. Nesmím zapomenout na zahrnutí průniku s prázdnou množinou = $\emptyset$.
4. Přidám extent zahrnující všechny objekty.
5. Pro odpovídající extenty vytvořím intenty (v grafu zapisuji **shora dolů**).

Příklad

	2 nohy	**4 nohy**	**mléko**	**vlna**
pes		×		
ovce		×	×	×
koza		×	×	
slepice	×			

$e_7 = \{\text{pes, ovce, koza, slepice}\}$ $\quad$ $i_7 = e_7^{\uparrow} = \emptyset$

$e_2 = \{\text{4 nohy}\}^{\downarrow} = \{\text{pes, ovce, koza}\}$ $\quad$ $i_2 = e_2^{\uparrow} = \{\text{4 nohy}\}$

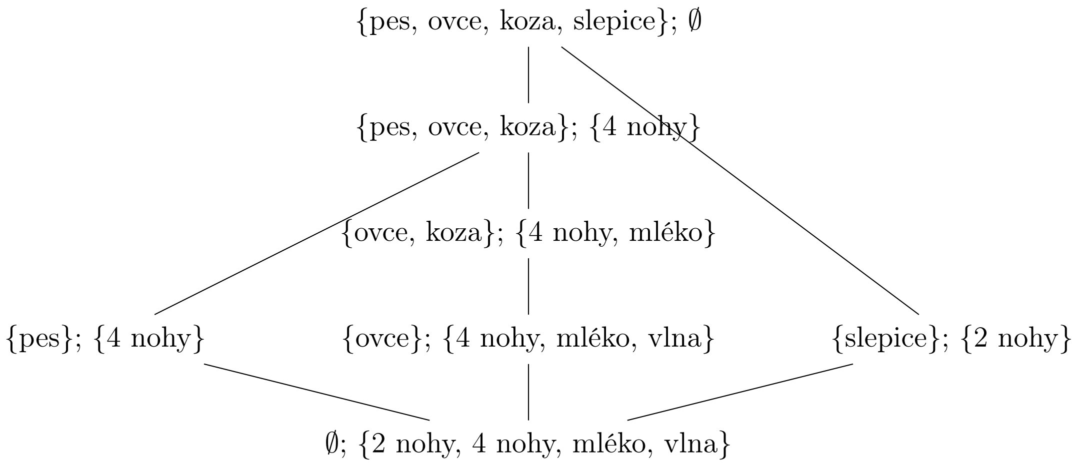
$e_3 = \{\text{mléko}\}^{\downarrow} = \{\text{ovce, koza}\}$ $\quad$ $i_3 = e_3^{\uparrow} = \{\text{4 nohy, mléko}\}$

$e_1 = \{\text{2 nohy}\}^{\downarrow} = \{\text{slepice}\}$ $\quad$ $i_1 = e_1^{\uparrow} = \{\text{2 nohy}\}$

$e_4 = \{\text{vlna}\}^{\downarrow} = \{\text{ovce}\}$ $\quad$ $i_4 = e_4^{\uparrow} = \{\text{4 nohy, mléko, vlna}\}$

$e_5 = e_2 \cap e_3 = \{\text{pes}\}$ $\quad$ $i_5 = e_5^{\uparrow} = \{\text{4 nohy}\}$

$e_6 = \emptyset$ $\quad$ $i_6 = e_6^{\uparrow} = \{\text{2 nohy, 4 nohy, mléko, vlna}\}$



8.5 Svazy (pro doplnění, asi není nutné úplně znák k FCA)

Svaz je algebra $(L, \cap, \cup)$ s dvěma základními binárními operacemi $x \cap y$ **spojení (suprémum)** (sup(x, y)) a $x \cup y$ **průsek (infimum)** (inf(x, y)), které mají následující vlastnosti:

1. **Univerzalita (jednoznačnost)** – $\forall x, y \, \exists z \;\; x \cap y = z \quad | \quad \forall x, y \exists \, z \;\; x \cup y = z.$
2. **Asociativita** – $x \cap (y \cap z) = (x \cap y) \cap z \quad | \quad x \cup (y \cup z) = (x \cup y) \cup z.$
3. **Komutativita** – $x \cap y = y \cap x \quad | \quad x \cup y = y \cup x.$
4. **Absorbce** – $x \cap (x \cup y) = x \quad | \quad x \cup (x \cap y) = x.$

8.5.1 Typy svazů

1. **Distributivní** – platí zde axiomy distributivity a neobsahuje ani **diamant** ani **pentagon**: $x \cup (x \cap z) = (x \cup y) \cap (x \cup z) \quad | \quad x \cap (x \cup z) = (x \cap y) \cup (x \cap z).$

2. **Modulární** – slabší reprezentace distributivity, **nesmí** obsahovat **pentagon**, $a \geq c:$ $a \wedge (b \vee c) = (a \wedge b) \vee c.$
3. **Komplementární** – platí zde, že pro každý prvek x existuje komplement x', kdy: $x \cap x' =$ [svazová 0] a $x \cup x' =$ [svazová 1]
4. **Boleaovský svaz** – komplementární $\wedge$ distributivní

8.5.2 Vlastnosti svazů

Pro každé dva prvky v množině existuje **sup** a **inf**. **Úplný svaz** – nastane tehdy, zda pro **libovolné neprázdné podmnožiny existuje** sup a inf. U svazů můžeme dále získat tyto vlastnosti:

- **Minimum a maximum** – žádný není menší/větší než a (vrchol diagramu).
- **Nejmenší a největší** – **pouze jeden** nejmenší/největší prvek, pokud je jich více, **neexistuje** největší/nejmenší prvek.
- **Dolní (L(A)) a horní (U(A)) závora** – všechny prvky jsou $\leq$ / $\geq$ než A,
- **Infimum** – nejmenší prvek dolní závory.
- **Supremum** – nejmenší prvek horní závory.

8.6 Asociační pravidla

Termín asociační pravidla široce zpopularizoval počátkem 90. let v souvislosti s analýzou nákupního košíku. Při této analýze se zjišťuje, jaké druhy zboží si současně kupují zákazníci v supermarketech (např. pivo a párek). **Jde** tedy **o hledání vzájemných vazeb** (**asociací**) **mezi různými položkami** sortimentu prodejny. Přitom není upřednostňován žádný speciální druh zboží jako závěr pravidla.

8.6.1 Základní charakteristika pravidel

U pravidel vytvořených z dat nás obvykle zajímá kolik příkladů splňuje **předpoklad** a kolik **závěr** pravidla, kolik příkladů splňuje předpoklad i závěr **současně**, kolik příkladů splňuje předpoklad a **nesplňuje** závěr.... Tedy, zajímá nás, jak pro pravidlo:

$$Ant \Rightarrow Suc, \qquad \text{kde } Ant, Suc \subseteq I \text{ (položky)} \tag{4}$$

kde Ant (**předpoklad**, levá strana pravidla, **antecedent**) a Suc (**závěr**, pravá strana pravidla, **sukcedent**) jsou kombinace kategorií, pro něž příslušná **kontingenční tabulka** vypadá následovně:

	Suc	$\neg Suc$
Ant	a	b
$\neg Ant$	c	d

- $Ant \wedge Suc$ – **a** je počet objektů pokrytých současně předpokladem i závěrem,
- $Ant \wedge \neg Suc$) – **b** je počet objektů pokrytých předpokladem a nepokrytých závěrem,
- $\neg Ant \wedge Suc$) – **c** je počet příkladů nepokrytých předpokladem ale pokrytých závěrem,
- $\neg Ant \wedge \neg Suc$) – **d** je počet příkladů nepokrytých ani předpokladem ani závěrem.

8.6.2 Základní charakteristiky asociačních pravidel

- **Support (podpora)** – relativní četnost objektů splňující předpoklad i závěr, jinými slovy $\frac{\text{počet splňující výstup}}{\text{počet položek}}$:

$$sup(Ant \Rightarrow Suc) = \frac{a}{a+b+c+d}, \in \langle 0; 1 \rangle. \tag{5}$$

- **Confidence (spolehlivost)** – podmíněná pravděpodobnost závěru pokud platí předpoklad, tedy $\frac{\text{podpora obou}}{\text{podpora závěru}}$:

$$conf(Ant \Rightarrow Suc) = \frac{sup(Ant \cup Suc)}{sup(Suc)}. \tag{6}$$

- Další: **pokrytí**, **zajímavost**, **závislost**.

8.7 Hledání často se opakujících množin položek)

Frequent item set je množina, kde $sup(K) \geq \gamma$, máme tedy stanovenou **minimální podporu**. Pokud je např. $\gamma = 0,3$ pak je minimální podpora 30%.

8.7.1 Generování kombinací

Základem všech algoritmů pro hledání asociačních pravidel je **generování kombinací (konjunkcí) hodnot atributů**. Při generování vlastně procházíme (prohledáváme) prostor všech přípustných konjunkcí. Metod je několik:

- do **hloubky**,
- do **šířky**,
- **heuristicky**,

8.7.2 Algoritmus apriori

Jedná se o nejznámějším algoritmus pro hledání asociačních pravidel Jádrem algoritmu je **hledání často se opakujících množin** položek (frequent itemsets). Jedná se kombinace (konjunkce) kategorií, které dosahují předem zadané četnosti (**minimální podpory**) v datech.

Algoritmus apriori

1. do L_1 přiřaď všechny kategorie, které dosahují alespoň požadované četnosti
2. polož k=2
3. dokud $L_{k-1} \neq \emptyset$
   3.1. pomocí funkce *apriori-gen* vygeneruj na základě L_{k-1} množinu kandidátů C_k
   3.2. do L_k zařaď ty kombinace z C_k, které dosáhly alespoň požadovanou četnost
   3.3. zvětš počítadlo k

Funkce apriori-gen(L_{k-1})

1. pro všechny dvojce kombinací $Comb_p$, $Comb_q$ z L_{k-1}
   1.1. pokud $Comb_p$ a $Comb_q$ se shodují v *k-2* kategoriích přidej $Comb_p \wedge Comb_q$ do C_k
2. pro každou kombinaci *Comb* z C_k
   2.1. pokud některá z jejich podkombinací délky *k-1* není obsažena v L_{k-1} odstraň *Comb* z C_k

Při hledání kombinací délky k, které mají vysokou četnost se využívá toho, že **již známe kombinace délky** $k-1$. Při vytváření kombinace délky k spojujeme kombinace délky $k-1$.

Jde tedy o **generování kombinací „do šířky"**. Přitom pro vytvoření jedné kombinace délky k požadujeme, aby všechny její podkombinace délky $k-1$ **splňovaly požadavek na četnosti**. Tedy např. ze tříčlenných kombinací $\{A_1A_2A_3,\ A_1A_2A_4,\ A_1A_3A_4,\ A_1A_3A_5,\ A_2A_3A_4\}$ dosahujících požadované četnosti vytvoříme **pouze jedinou čtyřčlennou** kombinaci $A_1A_2A_3A_4$. Kombinaci $A_1A_3A_4A_5$ sice lze vytvořit spojením $A_1A_3A_4$ a $A_1A_3A_5$, ale mezi tříčlennými kombinacemi chybí $A_1A_4A_5$ i $A_3A_4A_5$.

8.7.3 Algoritmus Next Closure

Slouží k vytváření formálních kontextů, **vyhledáváním nejmenších intentů**, postup:

1. Začnu s následující tabulkou:

A	i	$A \cap \{1 \dots i-1\} \cup \{i\} = B'$	$cl(B') = B$	$B \setminus A$	je-li $B \setminus A = \{i\}$ nebo větší $\rightarrow$ ANO je-li $B \setminus A = \{j\}$, kde $j < i \rightarrow$ NE
$\emptyset$	5				

2. Do A vložím prázdnou množinu a i nastavím na nejvyšší intent (pořadí).
3. Udělám průnik s $A \cap \{1 \dots i-1\} \cup \{i\} = B' \rightarrow$ průnik Áčka s intenty od 1 do $i-1$, k tomu přidám i. Př.: $A = \{1,3,4\}, i = \{3\} \Rightarrow B' = \{1,3\}$.
4. Udělám closure $(B') \rightarrow$ intent na extent $\rightarrow$ extent na intent $\Rightarrow B'^{\downarrow\uparrow}$.
5. Od B odečtu A.
6. Je-li:
   - $B \setminus A = \{i\}$ a větší tak ANO [je-li nejmenší prvek z $B \setminus A$ roven nebo větší než $\{i\}$],
   - $B \setminus A = \{j\}$ kde $j < i$ tak NE [je-li nejmenší prvek z $B \setminus A$ menší než i pak NE].
7. Pokud:
   - ANO $\rightarrow$ do A dosadíme $cl(B') = B$ a i nastavíme na nejvyšší intent,
   - NE $\rightarrow$ do A neměním a snížíme i o -1.
8. Skončím když je A rovno celé množině extentů.

DÚLEŽITÉ V i přeskakuju hodnoty, které jsou v A [výjde pro ně $\emptyset \rightarrow$ neřeším].

Příklad

	y_1	y_2	y_3	y_4	y_5
x_1	0	1	0	1	1
x_2	0	1	1	0	0
x_3	1	1	0	1	1
x_4	1	1	1	0	0
x_5	1	0	1	1	0
x_6	1	0	0	1	0

A	i	$A \cap \{1 \ldots i-1\} \cup \{i\} = B'$	$cl(B') = B$	$B \setminus A$	ANO / NE
$\emptyset$	5	5	2, 4, 5	2, 4, 5	N $[2 < 5]$
$\emptyset$	4	4	4	4	A $[4 \geq 4] \leftarrow i_n$
4	5	4, 5	2, 4, 5	2, 5	N $[2 < 5]$
4	3 [skip i]	3	3	3	A $[3 \geq 3] \leftarrow i_{n-1}$
3	5	3, 5	1, 2, 3, 4, 5	1, 2, 4, 5	N $[1 < 5]$
		$\vdots$			
1, 2, 3, 4, 5		KONEC			A i_1

9 Metrické a topologické prostory – metriky a podobnosti.

9.1 Metrický prostor

Metrický prostor je **matematická struktura**, pomocí které lze formálním způsobem definovat pojem **vzdálenosti**. Na metrických prostorech se poté definují další topologické vlastnosti jako např. **otevřenost** a **uzavřenost** množin, jejichž zobecnění pak vede na ještě abstraktnější matematický pojem **topologického prostoru**.

9.1.1 Formální definice

Metrický prostor je dvojice (M, ρ), kde M je libovolná neprázdná množina a ρ je **metrika**, což je zobrazení:

$$\rho : M \times M \to \mathbb{R},$$

které splňuje následující axiomy:

1. **Nezápornost**: $\forall x, y\, \rho(x, y) \geq 0$.
2. **Totožnost**: $\forall x, y\, \rho(x, y) = 0 \Leftrightarrow x = y$.
3. **Symetrie**: $\forall x, y \in M : \rho(x, y) = \rho(y, x)$.
4. **Trojúhelníková nerovnost**: $\forall x, y \in M : \rho(x, y) + \rho(y, z) \geq \rho(x, y)$.

9.1.2 Metriky v $\mathbb{R}^n$

U metrik v $\mathbb{R}^n$ platí:

- Každý normovaný vektorový prostor je metrickým prostorem.
- Množina reálných čísel spolu s metrikou $\rho(x, y) = |x - y|$, kde x, y jsou libovolné body množiny $\mathbb{R}$ tvoří **úplný metrický prostor**.

Mezi nejpoužívanější **metriky** na Euklidovském prostoru $\mathbb{R}^n$ patří:

1. **Manhattanská** – $\rho_1(x, y) = \sum_{i=0}^{n} |x_i - y_i|$.
2. **Euklidovská** – $\rho_2(x, y) = \sqrt{\sum_{i=0}^{n} |x_i - y_i|^2}$.
3. **Minkowského** – $\rho_3(x, y) = (\sum_{i=0}^{n} |x_i - y_i|^P)^{1/P}$, kde $P \geq 1$; $P \in \mathbb{R}$.
4. **Čebyševova (Maximova)** – $\rho_{\max}(x, y) = \max_{\forall i} |x_i - y_i|$, speciální případ Minkowského metriky pro $P = \infty$.

9.1.3 Podobnosti a nepodobnosti

- **Cosinova podobnost** – míra podobnosti dvou vektorů, která se získá výpočtem kosinu úhlu těchto vektorů:

$$S_c(x, y) = \frac{x \cdot y}{||x||\,||y||} = \frac{\sum_{i=0}^{n} x_i - y_i}{\sqrt{\sum_{i=0}^{n} x_i^2}\sqrt{\sum_{i=0}^{n} y_i^2}}$$

- **Jaccardova podobnost** – $S_j(X,Y) = \frac{|X \cap Y|}{X \cup Y}$ používá se pro porovnání podobnosti dvou množin.
- **Jaccardova nepodobnost** (svým způsobem vzdálenost) – $d = 1 - S$, $d = \frac{1}{S} - 1$ pro $S \neq 0$.

9.1.4 Vzdálenost mezi slovy

K určení vzdálenosti mezi textovými řetězci definujeme tyto **vzdálenosti**:

1. **Hammingova** – počet rozdílných písmen na stejných pozicích: d_h(Karolin, Kathrin) = 3, lze použít pouze pro **stejně dlouhá slova!**.
2. **LCS (Longest common subsequence)** – počet operací **vkládání** a **mazání** nutných k převodu jednoho slova na druhé: d_{LCS}(kitten, sitting) ⇒ itten [-K] → sitten [+S] → sittn [-E] → sittin [+I] → sitting [+G] = **5 operací**.
3. **Levenshteinova** – počet operací **vkládání**, **mazání**, **substituce** nutných k převodu jednoho slova na druhé: d_l(kittne, sitting) ⇒ sitten [K ∼ S] → sittin [E ∼ I] → sitting [+G] = **3 operace**.

Editační vzdálenost patří zde LSC a Levenstheinova vzdálenost (při určení se používají úpravy).

9.1.5 Normalizace

Snažíme se dostat všechny hodnoty atributů ve sloupci do intervalu $\langle 0, 1 \rangle$, proto dělíme celý sloupce jeho maximem (dělíme v rámci daného sloupce, pro každý sloupec zvlášť vlastním maximem).

Příklad

d'_1 a d'_2 reprezentují **normovaný tvar** vzdáleností d_1 a d_2:

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
d_1	1	0	0	1	3	0	1
d_2	1	0	0	2	2	0	0
d'_1	1	0	0	$\frac{1}{2}$	1	0	1
d'_2	1	0	0	1	$\frac{2}{3}$	0	0

	$i=1$	$i=2$	$i=3$	$i=\infty$	Cosinova	Jaccard
$\rho_i(d_1, d_2)$	3	$\sqrt{3}$	$\sqrt{3}^3$	1	$\frac{1}{6\sqrt{(3)}}$	$\frac{3}{4}$?
$\rho_i(d'_1, d'_2)$	$\frac{11}{6}$	$\frac{7}{6}$	$\frac{3\sqrt{251}}{6}$	1	$\frac{5\sqrt{(154)}}{3\sqrt{(2)}}$	$\frac{3}{4}$?

9.2 Topologický prostor

Jedná se o **rozšíření (zobecnění) metrického prostoru**. Cílem topologie je studium **vlastností prostorů**. Na rozdíl od teorie metrických prostorů se v topologii **nezajímáme o vzdálenosti mezi body** prostoru a prostory považujeme za stejné, pokud **se na sebe dají vzájemně přeměnit** nějakou spojitou deformací. Takže např. nerozlišujeme mezi koulí a krychlí, ostatně koule se změní v krychli již při přechodu mezi dvěma ekvivalentními metrikami v $\mathbb{R}^3$.

Základním pojmem, který se v topologii studuje je **spojitost zobrazení**. Proto není úplně potřeba vědět přesně, jak jsou od sebe které body daleko. Vystačíme si s informacemi, že jisté body se nekonečně blíží k nějakému bodu prostoru.

9.2.1 Formální definice

Topologickým prostorem nazveme množinu X společně s kolekcí τ podmnožin X, tedy **dvojici** (X, τ), splňující následující axiomy:

1. $\emptyset, X \in \tau$,
2. $\forall A, B \in \tau \Rightarrow A \cup B \in \tau$, tedy **sjednocení** libovolného počtu (tj. konečného, spočetného i nespočetného) množin z τ leží v τ,
3. $\forall A, B \in \tau \Rightarrow A \cap B \in \tau$, tedy **průnik** konečného počtu množin z τ leží v τ.

9.2.2 Uzavřená, otevřená množina

$A \in \tau$ ... A je **otevřená** pokud:

- $\emptyset, X \in \tau$,
- $\forall A, B \in \tau \Rightarrow A \cup B \in \tau$,
- $\forall A, B \in \tau \Rightarrow A \cap B \in \tau$.

$A \in \tau$ ... A je **uzavřená** pokud:

- $\emptyset, X \in \tau$,
- $\forall A, B \in \tau \Rightarrow A \cap B \in \tau$,
- $\forall A, B \in \tau \Rightarrow A \cup B \in \tau$.

9.2.3 Souvislost nesouvislost

Pokud sjednocením dvou neprázdných množin z τ získáme všechny prvky topologie (celé X), tak je topologie **nesouvislá**. Příklad:

$X = \{a, b, c\}$
$\tau = \{\emptyset, X, \{a, b\}, \{c\}\}$
$\{a, b\} \cup \{c\} = \{a, b, c\} \Rightarrow \tau$ je **nesouvislá**

Poznámka: sjednocované množiny musí být **disjunktní**, tedy nesmí mít společné prvky, např.: $\{a, b\}$ a $\{a, c\}$ již disjunktní nejsou, ale nenaruší souvislost.

Topologický prostor τ, je **souvislý** právě tehdy, když jedině podmnožiny v τ, které jsou současně otevřené i uzavřené jsou X a $\emptyset$. V opačném případě je τ **nesouvislý**.

9.2.4 Uzávěrový systém

Uzávěrový systém C nad množinou X obsahuje X a $\forall A, B \in C$ platí, že $A \cap B \in C$.

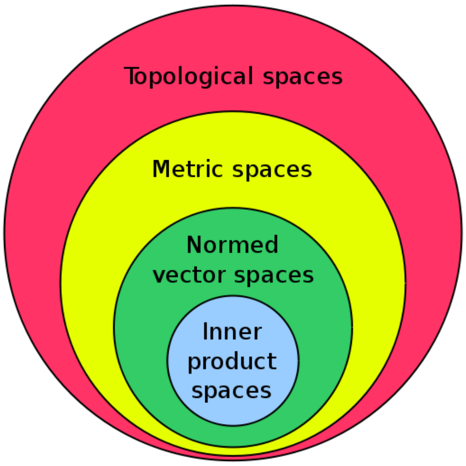
$$R_i \subseteq A \times A \quad R_i^* = [\text{tranzitivně-reflexivní uzávěr}]$$
$$\text{pro } R_i^* \text{ platí} \quad R_1^* \neq R_2^*; \quad R_1^* \cap R_2^* \in G; \quad R_1^* \cup R_2^* \notin G$$

9.2.5 Uzávěrový operátor $cl(A)$

Uzávěr (*closure*) $A \cup C \to cl(A)$ je **nejmenší uzavřená množina obsahující daný prvek**. Analogie u konceptů $cl(B) = B^{\downarrow\uparrow}$ a $cl(A) = A^{\uparrow\downarrow}$. Vlastnosti uzávěrového operátoru:

1. **Idempotence** – $cl(cl(A)) = cl(A)$.
2. **Extensionalita** – $A \subseteq cl(A)$.
3. **Monotónost** – $A \subseteq B \Rightarrow cl(A) \subseteq cl(B)$.

Platí-li navíc $cl(\emptyset) = \emptyset$ a $cl(A \cup B) = cl(A) \cup cl(B)$, pak je uzávěr $A = cl(A)$. Jinými slovy se jedná o nejmenší komplement (doplněk) množin TP obsahující daný prvek.



10 Shlukování.

11 Náhodná veličina. Základní typy náhodných veličin. Funkce určující rozdělení náhodných veličin.

Náhodný pokus – Děj, jehož výsledek není předem jednoznačně určen podmínkami, za nichž probíhá.
Náhodný jev – Tvrzení o výsledku náhodného pokusu, přičemž o pravdivosti tohoto tvrzení lze po ukončení pokusu rozhodnout. Náhodná veličina „NV“ – **číselné** vyjádření výsledku náhodného pokusu.

- NV je funkce, která každému elementárnímu jevu $\omega \in \Omega$ přiřadí reálné číslo.
- NV obvykle značíme velkými písmeny.
- NV přiřadí každému elementárnímu jevu ω reálné číslo (převádí elementární jevy (abstraktní objekty) na čísla).
- Hodnota $X(\omega)$ NV X závisí na tom, který elementární jev ω nastal.
- Víme–li, který elementární jev ω nastal, známe hodnotu $X(\omega)$ NV X.
- Příklady:

X... číselný výsledek hodu kostkou

Náhodný pokus: Hod kostkou.
Náhodný jev: Padne sudé číslo. ($X \in \{2; 4; 6\}$)

X... rychlost připojení k internetu (Mb/s)

Náhodný pokus: Měření rychlosti připojení k internetu (download).
Náhodný jev: Rychlost připojení k internetu je vyšší než 20 Mb/s. ($X > 20$)

X... počet dívek mezi 1 000 náhodně vybranými dětmi

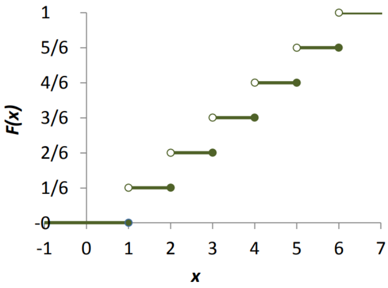
Náhodný pokus: Náhodný výběr 1 000 dětí a zjištění počtu dívek mezi nimi.
Náhodný jev: Mezi 1 000 náhodně vybranými dětmi bude více než 500 dívek. ($X > 500$)

11.1 Distribuční funkce F(x)

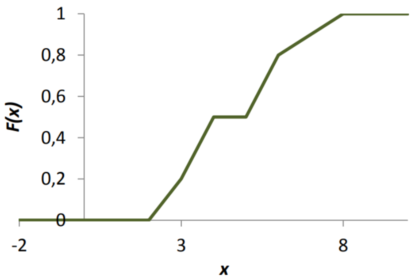
- Distribuční funkce F(t) udává pravděpodobnost, že náhodná veličina X bude **menší než** dané reálné číslo t.

$$F(t) = P(X < t)$$

- Distribuční funkce jednoznačně určuje rozdělení NV, tj. známe–li distribuční funkci, umíme určit pravděpodobnost $P(X \in M)$ pro libovolnou $M \subset \mathbb{R}$.



Ukázka distribuční funkce
diskrétní náhodné veličiny



Ukázka distribuční funkce
spojité náhodné veličiny

11.2 Základní typy náhodných veličin

- **Diskrétní NV**
- **Spojité NV**

11.2.1 Diskrétní náhodná veličina („DNV“)

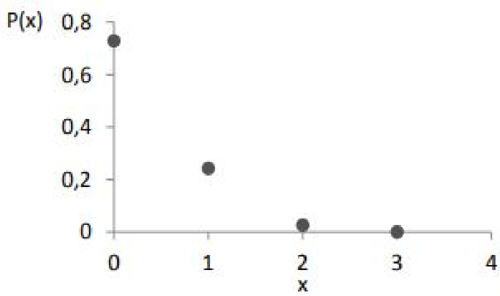
- mohou nabývat spočetně mnoha hodnot (počet dní hospitalizace, počet dní nemocenské, počet zákazníků v lékarně během jednoho dne..)
- DNV X s distribuční funkcí $F_x(t)$ je charakterizována **Pravděpodobnostní funkcí** $P(X = x_i)$, tj. funkcí pro níž platí:

$$F_x(t) = \sum_{x_i<t} P(X = x_i) = \sum_{x_i<t} P(x_i).$$

- **Pravděpodobnostní funkce**
  - $P(x_i) \geq 0$
  - $\sum_i P(X = x_i) = 1$
  - Lze zadat předpisem, tabulkou, grafem.

$$\forall x \in \{0; 1; 2; 3\}: P(X = x) = \binom{3}{x} \cdot 0{,}1^x \cdot 0{,}9^{3-x}$$

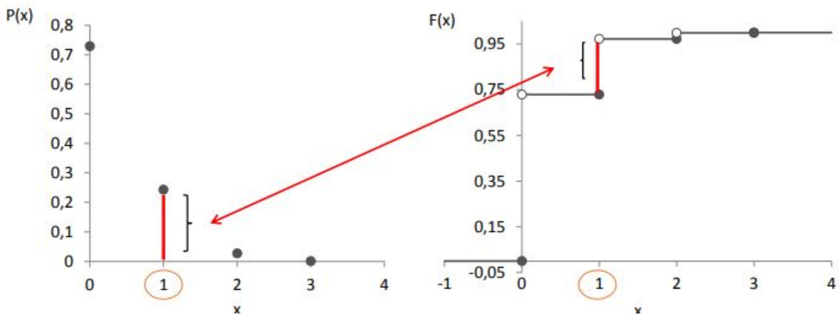
x	0	1	2	3
$P(x)$	0,729	0,243	0,027	0,001



⇒ *Příklad* – Při ověřování kvality výroby jsou náhodně vybrány dva výrobky a je testována jejich kvalita. Počet vadných výrobků mezi vybranými modelujeme náhodnou veličino X. Z dlouhodobého pozorování jsou známy údaje uvedené v následující tabulce.

Počet vadných výrobků	**Pravděpodobnost**
0	0,25
1	0,50
2	0,25

Určete pravděpodobnostní a distribuční funkci počtu vadných výrobků v testovaném vzorku.

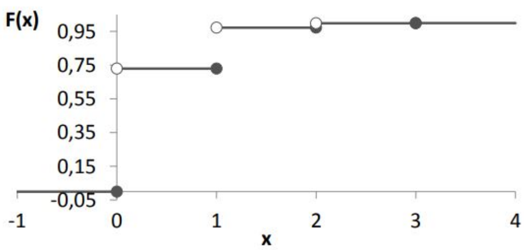


- Body nespojitosti distribuční f–ce jsou body, v nichž je pravděpodobnostní f–ce nenulová.
- $P(X = a) = \lim_{x \to a+} F(x) - F(a)$, tj. velikost „skoku“ distribuční funkce v bodech nespojitosti je rovna příslušným hodnotám pravděpodobnostní funkce.

- **Distribuční funkce**
  - $F(t) = \sum_{x_i < t} P(x_i)$
  - Lze zadat předpisem, tabulkou, grafem.

$$F(x) = \begin{cases} 0 & x \leq 0 \\ 0{,}729 & 0 < x \leq 1 \\ 0{,}972 & 1 < x \leq 2 \\ 0{,}999 & 2 < x \leq 3 \\ 1 & 3 < x \leq \infty \end{cases}$$

x	$F(x)$
$(-\infty; 0\rangle$	0
$(0; 1\rangle$	0,729
$(1; 2\rangle$	0,972
$(2; 3\rangle$	0,999
$(3; \infty)$	1

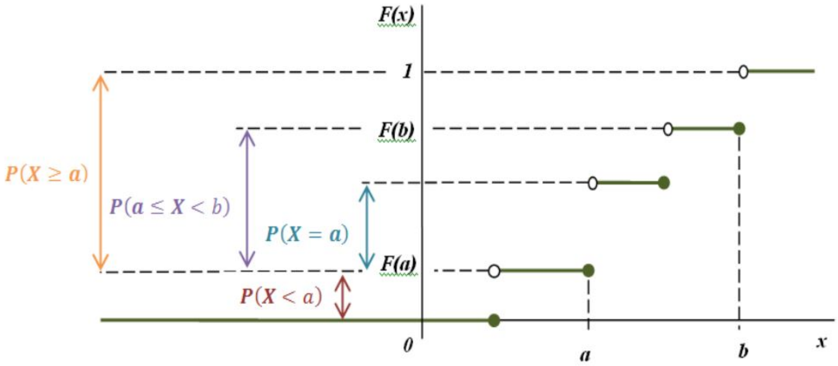


– Distribuční funkce $F(t)$ udává pravděpodobnost, že náhodná veličina X bude **menší než** dané reálné číslo t.

$$F(t) = P(X < t)$$

– Vlastnosti distribuční funkce:

  ◦ $0 \leq F(t) \leq 1$,
  ◦ je neklesající,
  ◦ je zleva spojitá,
  ◦ má nejvýše spočetně mnoho bodů nespojitosti,
  ◦ $F(t) \to 0$ pro $t \to -\infty$ („začíná“ v 0),
  ◦ $F(t) \to 1$ pro $t \to \infty$ („končí“ v 1).



$$P(X = a) \lim_{x \to a+} F(x) - F(a)$$

11.2.2 Spojitá náhodná veličina („SNV“)

- Náhodná veličina X má spojité rozdělení pravděpodobnosti (zkráceně „je spojitá“) právě když má spojitou distribuční funkci.
- Mohou nabývat všech hodnot na nějakém intervalu (mají spojitou distribuční funkci) (doba do remise onemocnění, výška, váha, BMO, IQ, vitální kapacita plic, chyba měření...).
- SNV X s distribuční funkcí $F_x(t)$ je charakterizována **hustotou pravděpodobností** $f(x)$, tj. funkcí pro níž platí:

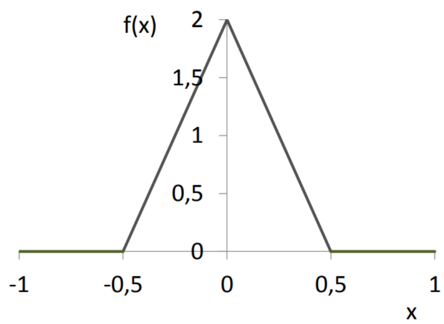
$$F_x(t) = \int_{-\infty}^{t} f(x)dx.$$

- **Hustota pravděpodobnosti f(x)**

  – $F(x) = \int_{-\infty}^{t} f(x)dx \Rightarrow f(x) = \frac{dF}{dx}$
  – Vlastnosti hustoty pravděpodobnosti:

    $f(x)$ je reálná nezáporná funkce,

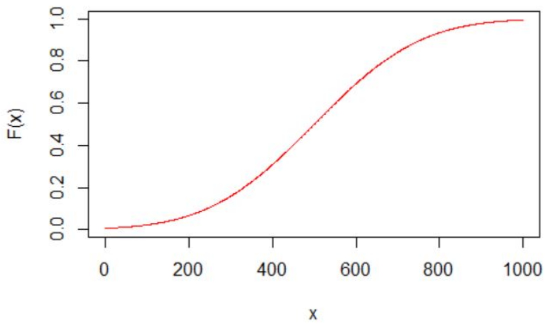
$\int_{-\infty} \infty f(x) = 1$ (plocha pod křivkou hustoty je 1),
$\lim_{x\to-\infty} f(x) = 0$ („začíná v 0“),
$\lim_{x\to\infty} f(x) = 0$ („končí v 0“).

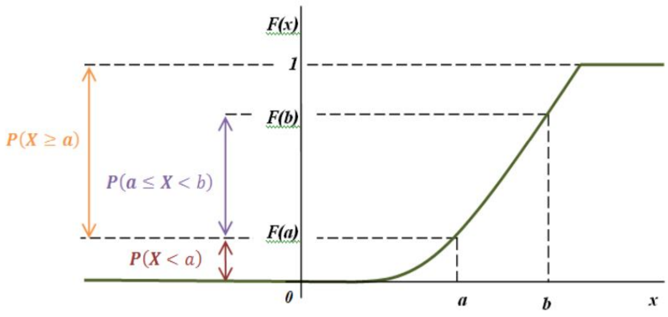


Obrázek 1: $f(x)$ může nabývat hodnot vyšších než 1

- **Distribuční funkce**

$$F_x(t) = \int_{-\infty}^{t} f(x)dx$$





Obrázek 2: Vztah mezi pravděpodobností a distribuční funkcí

$$P(X = a) = 0$$

11.3 Číselné charakteristiky NV

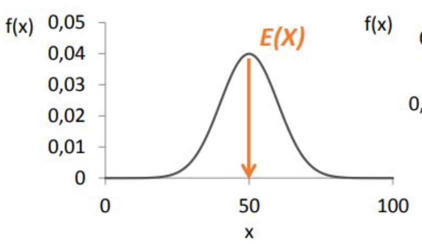
- Distribuční funkce (pravděpodobnostní funkce, hustota pravděpodobnosti) popisují rozdělení NV jednoznačně, do všech podrobností.
- Někdy nás zajímá pouze některý aspekt NV, který se dá popsat jedním číslem:
  - očekávaná hodnota NV,
  - variabilita možných hodnot,
  - hodnota, pod níž leží pouze malé množství hodnot NV,
  - šikmost rozdělení,
  - koncentrace hodnot NV kolem očekávané hodnoty (špičatost rozdělení).

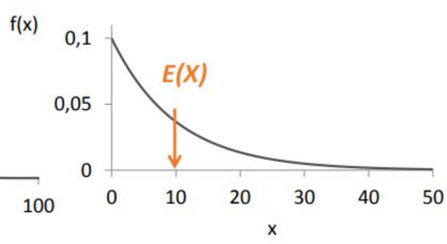
  - **Obecný moment r–tého řádu** (značí se μ_r nebo $E(X^r)$, pro r=1,2...)
    pro diskrétní NV: $\mu_r = \sum_{(i)} x_i^r \cdot P(x_i)$
    pro spojitou NV: $\mu_r = \int_{-\infty}^{\infty} x^r \cdot f(x)dx$
  - **Střední hodnota** (expected value, mean, značí se jako **E**(**X**) nebo μ)
    pro diskrétní NV: $E(X) = \mu = \sum_{(i)} x_i \cdot P(x_i)$
    pro spojitou NV: $E(X) = \mu = \int_{-\infty}^{\infty} x \cdot f(x)dx$
  - **Centrální moment r–tého řádu** $\mu_{\mathbf{r}}$’ (značíme μ_r’ $= E[(X - EX)]^r)$
    pro diskrétní NV: $\mu_r' = \sum_{(i)} (x_i - EX)^r \cdot P(x_i)$
    pro spojitou NV: $\mu_r' = \int_{-\infty}^{\infty} (x - EX)^r \cdot f(x)dx$
  - **Rozptyl** (dispersion, variance; značí se μ_2' nebo DX nebo σ^2)
    pro diskrétní NV: $D(X) = \mu_2' = \sum_{(i)} (x_i - EX)^2 \cdot P(x_i)$
    pro spojitou NV: $D(X) = \mu_2' = \int_{-\infty}^{\infty} (x - EX)^2 \cdot f(x)dx$

11.3.1 Význam střední hodnoty

Střední hodnotu $E(X)$ náhodné veličiny X lze chápat jako:

- průměrnou (očekávanou) hodnotu NV X, kolem níž hodnoty NV kolísají,
- míru polohy, populační průměr,
- vážený průměr všech možných hodnot ($E(X) = \sum_{(i)} x_i \cdot P(x_i)$),
- „těžiště“ možných hodnot.





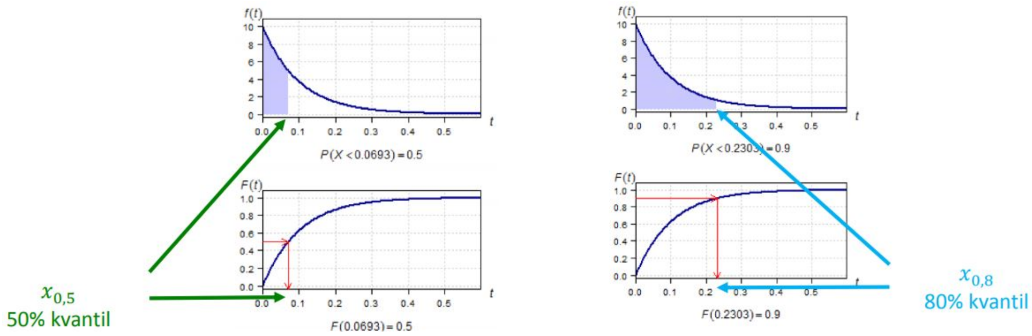
11.3.2 Kvantily

p–kvantil $\mathbf{x_p}$ (také 100_p%–ní kvantil je číslo, pro které platí):

$$P(X < x_p) = p.$$

$$\Rightarrow F(x_p) = p \Rightarrow x_p = F^{-1}(p)$$

(tj. kvantilová funkce $F^{-1}(p)$ je funkcí inverzní k distribuční funkci $F(x_p)$)



- Kvantily obvykle určujeme pouze pro SNV.
- Význačné kvantily
  - **Kvartily**

    Dolní kvartil $x_{0,25}$
    Medián $x_{0,5}$
    Horní kvartil $x_{0,75}$
  - **Decily** – $x_{0,1}; x_{0,2}; ...; x_{0,9}$
  - **Percentily** – $x_{0,01}; x_{0,02}; ...; x_{0,99}$
  - **Minimum** x_{min} a **Maximum** x_{max}

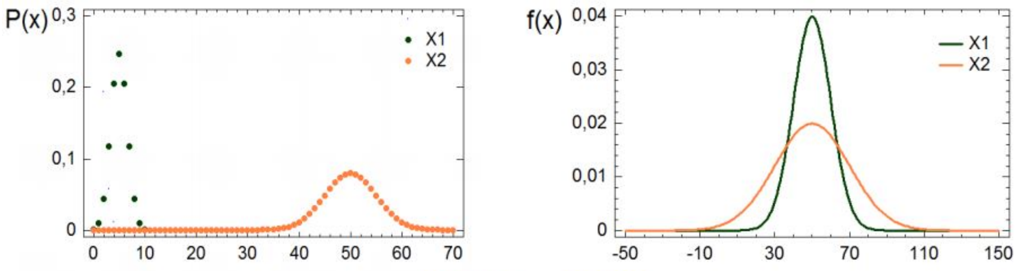
11.3.3 Modus

Modus $\hat{\mathbf{x}}$ – typická hodnota náhodné veličiny
pro diskrétní NV: $\forall x_i : P(X = \hat{x}) \geq P(X = x_i)$ (tzn. modus je taková hodnota DNV, v níž $P(x_i)$ nabývá svého maxima)
pro spojité NV: $\forall x_i : f(\hat{x}) \geq f(x)$ (tzn. modus je taková hodnota SNV, v níž $f(x)$ nabývá svého maxima)

- Modus není těmito podmínkami určen jednoznačně, tzn. náhodná veličina může mít několik modů (např. výsledek hodu kostkou). Má-li NV právě jeden modus, mluvíme o unimodálním rozdělení NV.
- Má-li NV unimodální symetrické rozdělení, pak $E(X) = x_{0,5} = \hat{x}$.

11.3.4 Význam rozptylu

- Míra variability dat kolem střední hodnoty.
- Střední kvadratická odchylka od střední hodnoty ($D(X) = E(X - E(X))^2$).
- Malý rozptyl $\approx$ hodnoty NV se s vysokou pravděpodobností objevují blízko $E(X)$.
- Velký rozptyl $\approx$ hodnoty NV se často objevují ve velké vzdálenosti od $E(X)$.
- Jednotka rozptylu je kvadrátem jednotky náhodné veličiny.



$D(X1) < D(X2)$

11.3.5 Směrodatná odchylka

Směrodatná odychylka σ

$$\sigma(Y) = \sqrt{D(X)}$$

Směrodatná odchylka neumožňuje srovnávat variabilitu náhodných veličin měřených v různých jednotkách!

$$\forall k > 0 \colon P(\mu - k\sigma < X < \mu + k\sigma) > 1 - \tfrac{1}{k^2} \text{ (Čebyšova nerovnost)}$$

11.3.6 Variační koeficient

Variační koeficient γ je definován pouze pro nezáporné náhodné veličiny.

$$\gamma = \frac{\sigma}{\mu'}, resp. \frac{\sigma}{\mu'} \cdot 100[\%]$$

- Variační koeficient — směrodatná odchylka v procentech střední hodnoty
- Čím nižší var. koeficient, tím homogennější soubor.
- $V_X > 50\%$ značí silně rozptýlený soubor.

12 Vybraná rozdělení diskrétní a spojité náhodné veličiny - binomické, hypergeometrické, negativně binomické, Poissonovo, exponenciální, Weibullovo, normální rozdělení.

- Náhodná veličina – číselné vyjádření výsledku náhodného pokusu.
- Základní typy NV – diskrétní NV, spojitá NV.
- Rozdělení pravděpodobnosti – Předpis, který jednoznačně určuje všechny pravděpodobnosti typu $P(X \in M)$, kde $M \subset \mathbb{R}$
(tj. $P(X = a), P(X < a), P(X > a), P(a < X < b), ...,$ kde $a, b \in \mathbb{R}$).
- Rozdělení pravděpodobnosti DNV – distribuční funkcí $F(x) = P(X < x)$, resp. pravěpodobnostní funkcí $P(x_i)$.
- Rozdělení pravděpodobnosti SNV – distribuční funkcí $F(x) = P(X < x)$, resp. hustotou pravěpodobnostní $f(x)$.
- Číselné charakteristiky pro popis NV – střední hodnota $\mathbf{E(X)}$, rozptyl $\mathbf{D(X)}$, směrodatná odchylka $\sigma(\mathbf{X})$, p–kvantily $\mathbf{x_p}$

12.1 Diskrétní náhodná veličina

Příklady

- počet šroubů typu M10 mezi 10 vybranými (víme-li, že do dodávky 100 šroubů bylo omylem zařazeno 20 šroubů typu M50)
- počet pacientů (z 10 očkovaných) u nichž byla použita prošlá očkovací látka (víme-li, že v balení bylo 20 dávek očkovací látky, přičemž 5 z nich bylo prošlých)

  ▷ **Obecně:** – počet úspěchů v n (závislých) pokusech

- počet správně přenesených bitů předtím než dojde ke 4. chybě (víme-li, že pravděpodobnost chybného přenosu bitu je $0,12$)
- počet dobrovolníků, které budeme muset testovat dříve než najdeme 5 dárců s krevní skupinou AB (předpokládejme, že dobrovolníci neznají svou krevní skupinu, pravděpodobnost výskytu krevní skupiny (populační frekvence) krevní skupiny AB je $0,05$)

  ▷ **Obecně:** – počet (nezávislých) pokusů do k. úspěchu

- počet škrábanců na $1m^2$ lakovaného povrchu (víme-li, že průměrně lze očekávat 3 škrábance na $10m^2$)
- počet červených krvinek v $10ml$ krve ženy (víme-li, že u průměrně lze pozorovat $4,8 \cdot 10^{12}$ červených krvinek v $1l$ krve (u žen))

  ▷ **Obecně:** – počet událostí v časovém intervalu, na ploše, v objemu

12.1.1 Bernoulliho pokusy

- **Posloupnost nezávislých pokusů majících pouze dva možné výsledky** (takových pokusů, kdy úspěch v libovolné skupině pokusů neovlivňuje pravděpodobnost úspěchů v pokusu, který do této skupiny nepatří)
- **v nichž jev A** (úspěch) **nastává s pravděpodobností** π a neúspěch s pravděpodobností $1 - \pi$.

◦ *Příklad* Předpokládejme, že pravděpodobnost narození dívky je $0,49$. Jaká je pravděpodobnost toho, že mezi čtyřmi dětmi v rodině je právě jedna dívka?
X. . . počet dívek mezi 4 dětmi
D ... narodí se dívka, $P(D) = \pi$
$\bar{D}$ ... narodí se kluk, $P(D) = 1 - \pi$

$$(X = 1)...\{D\bar{D}\bar{D}\bar{D}, \bar{D}D\bar{D}\bar{D}, \bar{D}\bar{D}D\bar{D}, \bar{D}\bar{D}\bar{D}D\}$$
$$P(D\bar{D}\bar{D}\bar{D}) = P(\bar{D}D\bar{D}\bar{D}) = P(\bar{D}\bar{D}D\bar{D}) = P(\bar{D}\bar{D}\bar{D}D) = \pi \cdot (1 - \pi)^3$$
$$P(X = 1) = P(D\bar{D}\bar{D}\bar{D} \cup \bar{D}D\bar{D}\bar{D} \cup \bar{D}\bar{D}D\bar{D} \cup \bar{D}\bar{D}\bar{D}D) = \binom{4}{1} \cdot \pi \cdot (1 - \pi)^3$$
$$P(X = 1) = 0,260$$

12.1.2 Binomické rozdělení

- **X .. počet úspěchů v n Bernoulliho pokusech**

$$X \sim Bi(n; \pi)$$

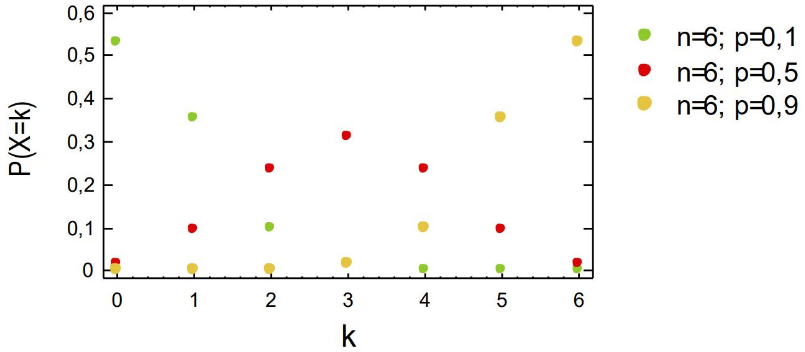
n – počet pokusů
π – pravděpodobnost úspěchu

Počet příznivých realizací posloupnosti Bernoulliho pokusů.

Pravděpodobnostní funkce: $P(X = k) = \binom{n}{k}\underbrace{\pi^k (1 - \pi)^{n-k}}$

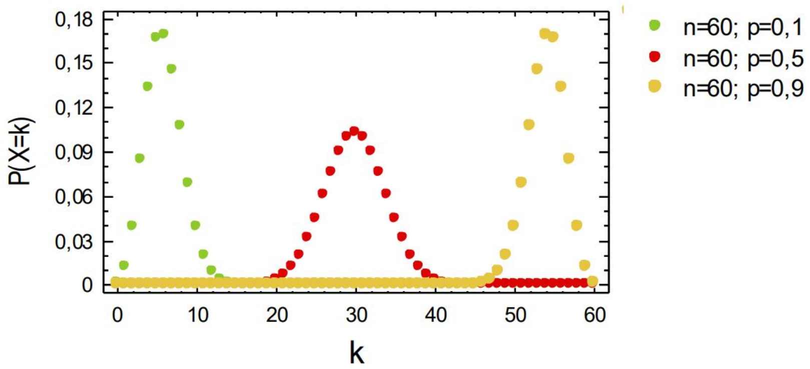
Pravděpodobnost výskytu jedné příznivé realizace posloupnosti Bernoulliho pokusů.

- **Střední hodnota**: $E(X) = n\pi$
- **Rozptyl**: $D(X) = n\pi(1 - \pi)$
- Vlastnosti



- $\pi < 0{,}5$, malé n $\Rightarrow$ poz. zešikmené r.
- $\pi = 0{,}5$ $\Rightarrow$ symetrické r.
- $\pi < 0{,}5$, malé n $\Rightarrow$ neg. zešikmené r.

Obrázek 3: $\pi \neq 0,5$, s rostoucím n se rozdělení stává více a více symetrickým



- *Příklad* Mezi 200 vajíčky určenými pro prodej v jisté maloobchodní prodejně je 50 vajíček prasklých. Jaká je pravděpodobnost, že vybereme–li si náhodně 20 vajec, bude 8 z nich prasklých?

  – X ... počet prasklých vajec z 20 vybraných (výběr bez vracení)

celkem vajec
200
/ \
prasklých vajec nepprasklých vajec
50 150

$$P(X = 8) = \frac{\binom{50}{8}\binom{150}{12}}{\binom{200}{20}}$$

počet příznivých možností – vybíráme 8 prasklých z 50 prasklých a zároveň 12 ze 150 neprasklých
počet všech možností – vybíráme 20 vajec z 200

12.1.3 Hypergeometrické rozdělení

- **X ... počet prvků se sledovanou vlastností ve výběru n prvků**
- V souboru **N** prvků je **M** s danou vlastností a zbylých (**N − M**) prvků tuto vlastnost nemá. Postupně vybereme ze souboru **n** prvků, z nichž žádný **nevracíme zpět**.

$$X \sim H(N; M; n)$$

N – rozsah základního souboru
M – počet prvků s danou vlastností
n – rozsah výběru

- **Pravděpodobnostní funkce**: $P(X = k) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$
  - Počet příznívých možností, tj. počet možností jak vybrat k prvků s danou vlastností z M a zároveň $(n - k)$ prvků, které uvedenou vlastnost nemají z $(N - M)$ prvků.
  - Počet všech možností, jak vybrat n prvků z N (nezáleží na pořadí).
- **Střední hodnota**: $E(X) = n \cdot \frac{M}{N}$
- **Rozptyl**: $D(X) = n \cdot \frac{M}{N}(1 - \frac{M}{N})(\frac{N-n}{N-1})$
- **Možnost aproximace** – je–li n/N, tzv. **výběrový poměr**, menší než $0,05$, lze hypergeometrické rozdělení nahradit binomickým s parametry n a M/N.

$$(\frac{n}{N} < 0,05) \Rightarrow [H(N; M; n) \approx Bi(n; \frac{M}{N})]$$

12.1.4 Negativně binomické (Pascalovo) rozdělení

- X **... počet Bernoulliho pokusů do k. výskytu události (úspěchu), včetně k. výskytu**

$$X \sim NB(k; \pi)$$

k – požadovaný počet úspěchů (výskytu události)
π – pravděpodobnost úspěchu

- **Pravděpodobnostní funkce:**

$$P(X = n) = \underbrace{\binom{n-1}{k-1}\pi^{k-1}(1-\pi)^{((n-1)-(k-1))}}_{\text{pravděpodobnost } k-1 \text{ úspěchů v } n-1 \text{ Bernoulliho pokusech}} \cdot \underbrace{\pi}_{\text{pravděpodobnost } k. \text{ úspěchu}}$$

- **Střední hodnota**: $E(X) = \frac{k}{\pi}$

- **Rozptyl**: $D(X) = \frac{k(1-\pi)}{\pi^2}$
- V případě negativně binomické náhodné veličiny není definice ustálená. Někteří statistici (popř. statistický software) ji definují jako **počet neúspěchů před** k. **úspěchem**.

12.1.5 Poissonův proces

Bodový proces

- Sledujeme chod procesu, v němž čas od času dochází k nějaké význačné události

- Registrujeme počet událostí $N(s;t)$ v časovém intervalu $< s; s+t >$
- **Rychlost výskytu události** λ je parametrem Poissonova procesu
- Poissonův proces lze obdobně jako v časovém intervalu definovat na libovolné uzavřené prostorové oblasti (na ploše, v objemu).

Jako **Poissonův proces** označujeme proces, který je:

- **ordinální** – pravděpodobnost výskytu více než jedné události v limitně krátkém časovém intervalu ($t \to 0$) je nulová. (tzv. **řídké jevy**),
- **stacionální** – rychlost výskytu událostí λ je konstantní v průběhu sledovaného intervalu,
- **beznásledný** – pravděpodobnost výskytu událostí není závislá na čase, který uplynul od minulé události.

12.1.6 Poissonovo rozdělení

- **X ... počet výskytu události v uzavřené oblasti (v čase, na ploše, v objemu)**
- náhodný pokus jako Poissonův proces (nezávislé události probíhající v čase t, s rychlostí výskytu λ; popř. nezávislé události objevující se na ploše t, resp. v objemu t s hustotou výskytu λ).

$$X \sim Po(\lambda t)$$

- **Pravděpodobnostní funkce**: $P(X = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$
- **Střední hodnota**: $E(X) = \frac{k}{\pi}$
- **Rozptyl**: $D(X) = \frac{k(1-\pi)}{\pi^2}$

12.2 Spojitá náhodná veličina

👉 Hustota pravděpodobnosti f(x) – funkce $f(x)$ je **hustotou pravděpodobnosti** (na intervalu $a \leq x \leq b$), jestliže splňuje následující podmínky:

- $f(x) \geq 0; x \in \mathbb{R}$
- plocha pod křivkou hustoty je rovna 1 ($\int_{-\infty}^{\infty} f(x)dx = 1$).

12.2.1 Exponenciální rozdělení

- **X ... délka časových intervalů mezi událostmi v Poissonově procesu**

$$X \sim Exp(\lambda)$$

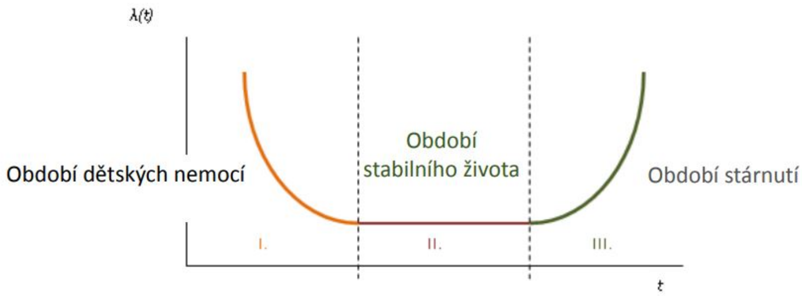
- Předpokládá konstantní intenzitu události $\lambda(t)$- rozdělení „bez paměti“.
- **Hustota pravděpodobnosti**: $f(t) = \begin{cases} \lambda \cdot e^{-\lambda t} & t > 0 \\ 0, & t \leq 0. \end{cases}$
- **Distribuční funkce**: $F(t) = \begin{cases} 1 - e^{-\lambda t} & t > 0 \\ 0, & t \leq 0. \end{cases}$
- **Střední hodnota**: $E(X) = \frac{1}{\lambda}$
- **Rozptyl**: $D(X) = (E(X))^2 = \frac{1}{\lambda^2}$

- **Riziková funkce** (intenzita poruch) $\lambda(t)$
  - Pro nezápornou náhodnou veličinu X se spojitým rozdělením popsaným distribuční funkcí $F(t)$ definujeme pro $F(t) \neq 1$ rizikovou funkci jako

$$\lambda(t) = \frac{f(t)}{1 - F(t)}.$$

  - $\lambda(t)$ – Představuje-li náhodná veličina X dobu do poruchy nějakého zařízení, pak pravděpodobnost, že pokud do času t nedošlo k žádné poruše, tak k ní dojde v následujícím krátkém úseku délky Δt, je přibližně

$$P(t \geq X < t + \Delta t | X > t) = \lambda(t) \cdot \Delta t$$



- *Příklad* Střední doba čekání zákazníka na obsluhu v prodejně je 50 sekund. Doba čekání se řídí exponenciálním rozdělením (pravděpodobnost, že zákazník nebude obsloužen klesá s rostoucím časem exponenciálně). Jaká je pravděpodobnost, že náhodný zákazník bude obsloužen dříve než za 30 sekund?

  X – doba čekání na obsluhu
  $X \sim Exp(\lambda = \frac{1}{50}s^-1), E(X) = \frac{1}{\lambda} = 50s$

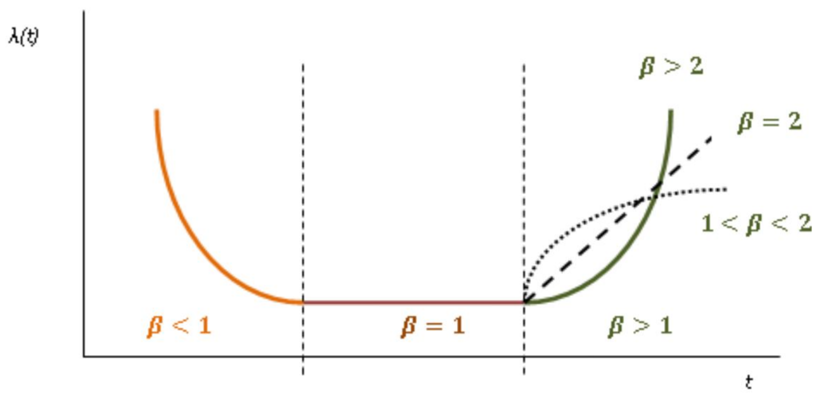
12.2.2 Weibullovo rozdělení

- **X ... délka časových intervalů mezi událostmi v Poissonově procesu**
- Zobrecnění exponenciálního rozdělení
- Tímto rozdělením lze modelovat i dobu do výskytu události u systémů (jedinců), které jsou v období dětských nemocí, resp. období stárnutí.

$$X \sim W(\theta; \beta)$$

θ – parametr měřítka (scale) $\theta = \frac{1}{\lambda}; \theta > 0$
β – parametr tvaru (shape); $\beta > 0$

- **Hustota pravděpodobnosti**: $f(t) = \begin{cases} \beta\lambda^\beta t^{\beta-1} e^{-(\lambda t)^\beta}, & t > 0 \\ 0, & t \leq 0. \end{cases}$
- **Distribuční funkce**: $F(t) = \begin{cases} 1 - e^{-(\lambda t)^\beta} & t > 0 \\ 0, & t \leq 0. \end{cases}$
- **Riziková funkce**: $\lambda(t) = \begin{cases} \beta\lambda^\beta t^{\beta-1} & t > 0 \\ 0, & t \leq 0. \end{cases}$



Obrázek 4: Riziková funkce

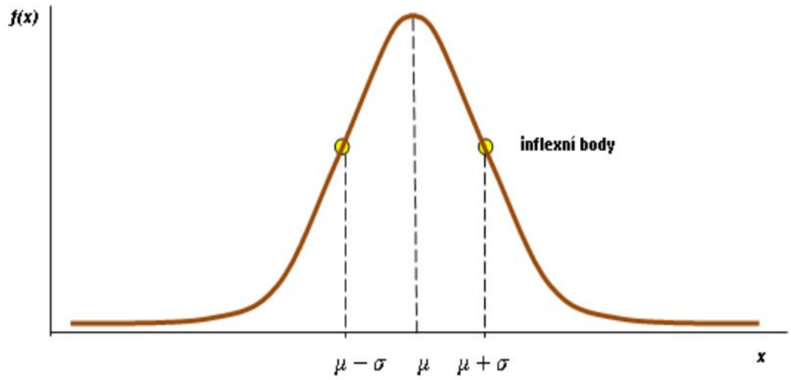
12.2.3 Normální rozdělení

- Bývá vhodné k popisu náhodných veličin, které lze interpretovat jako aditivní výsledek mnoha nepatrných a vzájemně nezávislých faktorů (výška člověka, IQ, délka končetiny).
- Popisuje náhodné veličiny, jejichž hodnoty se symetricky shlukují kolem střední hodnoty a vytvářejí tak charakteristický tvar hustoty pravděpodobnosti známý pod názvem Gaussova křivka.

$$X \sim N(\mu; \sigma^2)$$

μ – střední hodnota
σ^2 – rozptyl

- **Hustota pravděpodobnosti**: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$



Obrázek 5: Riziková funkce

- **Distribuční funkce**: $F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2} dt$
(integrál nelze řešit analyticky)

13 Popisná statistika. Číselné charakteristiky a vizualizace kategoriálních a kvantitativních proměnných.

13.1 Kvantitativní – Numerická proměnná

- **Míry polohy** – určující typické rozložení hodnot proměnné (jejich rozmístění na číselné ose).
  - Průměr – $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$
  - Modus – střed shorthu
  - Kvantily – dolní kvartil, médián, horní kvartil,..
- **Míry variability** – určující variabilitu (rozptyl) hodnot kolem své typické polohy.
  - Variační rozpětí $x_{max} - x_{min}$
  - Inverkvartilové rozpětí – $IQR = x_{0,75} - x_{0,25}$
  - Výběrový rozptyl
  - Výběrová směrodatná odchylka
  - Variační koeficient
- **Míra šikmosti a špičatosti**
  - Výběrová šikmost
  - Výběrová špičatost
- **Identifikace odlehlých pozorování**
  - Vnitřní hradby dolní mez: $h_D = x_{0,25} - 1,5IQR$, horní mez: $h_H = x_{0,75} + 1,5IQR$
  - Vnější hradby dolní mez: $h_D = x_{0,25} - 3IQR$, horní mez: $h_H = x_{0,75} + 3IQR$
  - Z–souřadnice
  - Mediánová souřadnice

Grafické zobrazení numerické proměnné:

- Empirická distribuční funkce
- Krabicový graf (box plot)
- Číslicový histogram (lodyha s listy, steam and leaf)

13.2 Kvalitativní –Kategoriální proměnná

a) Nominální proměnná – nemá smysl uspořádaní

- **Základní statistiky pro popis nominální proměnné:**

- četnost
- relativní četnost
- modus

- **Grafické zobrazení nominální proměnné:**
  - histogram
  - výsečový graf

b) Ordinální proměnná – má smysl uspořádání

- **Základní statistiky pro popis ordinální proměnné:**
  - četnost
  - relativní četnost
  - kumulativní četnost
  - relativní kumulativní četnost
  - modus
- **Grafické zobrazení ordinální proměnné:**
  - histogram
  - výsečový graf
  - Lorenzova křivka
  - Paretův graf

Paretův princip – 80% následků pramení z 20% příčin.
Paretova analýza – postup vedoucí k nalezení „životně důležité menšiny“ (spektra příčin ovlivňujících rozhodujícím způsobem následky).

13.3 Míry polohy a variability

Snad nejpoužívanějšími mírami polohy jsou průměry proměnných. Průměry představují průměrnou nebo typickou hodnotu výběrového souboru. Zřejmě nejznámějším průměrem pro kvantitativní proměnnou je **aritmetický průměr**

- **Aritmetický průměr $\bar{\mathbf{x}}$** (mean)
- **Aritmetický vážený průměr**
- **Harmonický průměr** – Pro výpočet průměru v případech, kdy proměnná má charakter části z celku (úlohy o společné práci, ...).
- **Harmonický vážený průměr** – Pokud máme údaje setříděné do tabulky četností.
- **Geometrický průměr** – Pracujeme-li s kladnou proměnnou představující relativní změny (růstové indexy,cenové indexy...). $\bar{x}_G = \sqrt[n]{x_1 \cdot x_2 \cdot ... \cdot x_n}$
- **Geometrický vážený průměr** – Pracujeme-li s kladnou proměnnou představující relativní změny (růstové indexy,cenové indexy...). $\bar{x}_G = \sqrt[n]{x_1^{n_1} \cdot x_2^{n_2} \cdot ... \cdot x_n^{n_k}}$ kde

$n = \sum_{i=1}^{k} n_i.$

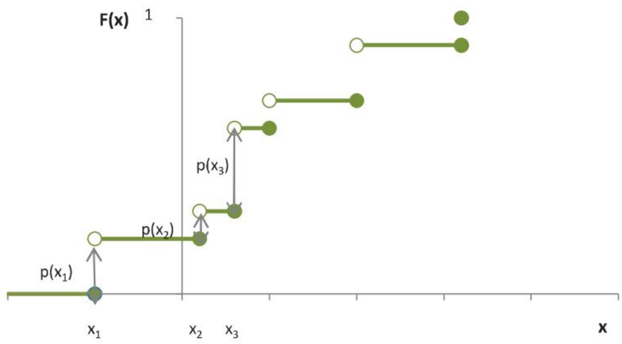
- **Modus** – Pro **diskrétní proměnnou** definujeme modus jako hodnotu nejčetnější varianty proměnné (podobně jako u kvalitativní proměnné), u spojité proměnné považujeme za modus $\hat{\mathbf{x}}$ hodnotu kolem níž je největší koncentrace hodnot proměnné. Mnohdy mluvíme o typické hodnotě proměnné. Pro určení této hodnoty využijeme tzv. **short**, což je nejkratší interval, v němž leží alespoň 50% hodnot proměnné (v případě výběru o rozsahu $n = 2k(k \in N)$ (sudý počet hodnot), leží v shorthu k hodnotě - což je 50% ($n/2$) hodnot proměnné, v případě výběru o rozsahu $n = 2k + 1(k \in N)$ (lichý počet hodnot), leží v shorthu $k + 1$ hodnot - což je o 1 více než je 50% hodnot proměnné). **Modus pak definujeme jako střed shorthu**.

- **Výběrové kvantily** (quantile, resp. percentile) – jsou to statistiky, které charakterizují polohu jednotlivých hodnot v rámci proměnné. Podobně jako modus, jsou i výběrové kvantily **rezistentní** (odolné) vůči odlehlým pozorováním. Obecně je výběrový kvantil chápán jako hodnota, která rozděluje výběrový soubor na dvě části – hodnoty, které jsou menší než daný kvantil, druhá část obsahuje hodnoty, které jsou větší nebo rovno danému kvantilu. Pro určení kvantilu je nutné **výběr uspořádat** od nejmenší hodnoty k největší.

- **Kvartily**
  - **Dolní kvartil** $x_{0,25}$ – 25%–ní kvartil (rozděluje datový soubor tak, že 25% hodnot je menších než tento kvartil a zbytek, tj. 75% větších (nebo rovných))
  - **Medián** $x_{0,5}$ – 50%–ní kvartil
  - **Horní kvartil** $x_{0,75}$ – 75%–ní kvartil
    Kvartily dělí výběrový soubor na 4 přibližně stejně velké části.
  - **Decily** – $x_{0,1}; x_{0,2}; ...; x_{0,9}$ – Decily dělí výběrový soubor na 10 přibližně stejně četných částí.
  - **Percentily** – $x_{0,01}; x_{0,02}; ...; x_{0,99}$ – Percentily dělí výběrový soubor na 100 přibližně stejně četných částí.

- **Empirická distribuční funkce F(x) pro kvantitativní proměnnou**
  - Empirická distribuční funkce je monotónně rostoucí, zleva spojitou funkcí, která „skáče“ podle relativních četností příslušných jednotlivým hodnotám proměnné.
  - Označme si $p(x_i)$ relativní četnost hodnoty xi seřazeného výběrového souboru $x_1 < x_2 < ... < x_n$. Pro empirickou distribuční funkci $F(x)$ pak platí:

$$F(x) = \begin{cases} 0 & \text{pro } x \leqq x_i \\ \sum\limits_{i=1}^{j} F(x) & \text{pro } x_j < x \leqq x_{j+1}, 1 \leqq j \leqq n-1 \\ 1, & \text{pro } x_n < x \end{cases}$$



- **Interkvartilové rozpětí IQR** – Tato statistika je mírou variability souboru a je definována jako vzdálenost mezi horním a dolním kvartilem.
- **MAD** (median absolute deviation from the median) – medián absolutních odchylek od mediánu

  1. Výběrový soubor uspořádáme podle velikosti.
  2. Určíme medián souboru.
  3. Pro každou hodnotu souboru určíme absolutní hodnotu její odchylky od mediánu.
  4. Absolutní odchylky od mediánu uspořádáme podle velikosti.
  5. Určíme medián absolutních odchylek od mediánu, tj. MAD.

- **Výběrový rozptyl $\mathbf{s^2}$** („s kvadrát“, sample variance) –

  – je nejrozšířenější mírou variability výběrového souboru

  $$s^2 = \frac{\sum\limits_{i=1}^{n} (x_1 - \bar{x})^2}{n-1}$$

  – Výběrový rozptyl je dán podílem součtu kvadrátu odchylek jednotlivých hodnot od průměru a rozsahu souboru sníženého o jedničku.

  – Nevýhodou použití výběrového rozptylu jakožto míry variability je to, že jednotka této charakteristiky je druhou mocninou jednotky proměnné. Např. je–li proměnnou denní tržba uvedena v Kč, bude výběrový rozptyl této proměnné vyjádřen v Kč^2.

  – Následující míra variability tuto vlastnost nemá.

- **Výběrová směrodatná odchylka s** – je definována jako kladná odmocnina výběrového rozptylu

$$s = \sqrt{s^2} = \sqrt{\frac{\sum\limits_{i=1}^{n}(x_1 - \bar{x})^2}{n-1}}.$$

Nevýhodou výběrového rozptylu i výběrové směrodatné odchylky je skutečnost, že neumožňují porovnávat varibilitu proměnných vyjádřených v různých jednotkách. Která proměnná má větší variabilitu – výška nebo hmotnost dospělého člověka? Na tuto otázku nám dá odpověď tzv. variační koeficient.

- **Variační koeficient $\mathbf{V_x}$** – vyjadřuje relativní míru variability proměnné x. Podle níže uvedeného vztahu jej lze stanovit pouze pro proměnné, které nabývají výhradně kladných hodnot. Variační koeficient je bezrozměrný. Uvádíme-li jej v [%], hodnotu získanou z definičního vzorce vynásobíme 100%.

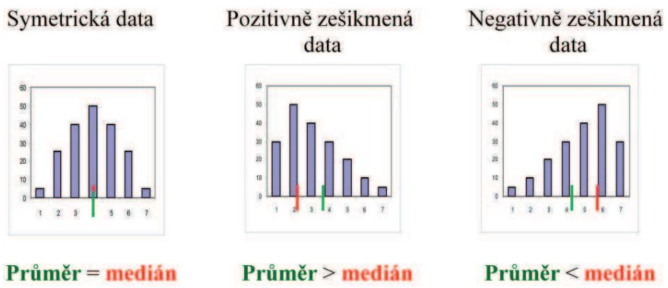
$$V_x = \frac{V}{\bar{x}}, popr. V_x = \frac{V}{\bar{x}} \cdot 100[\%].$$

13.4 Identifikace odlehlých pozorování

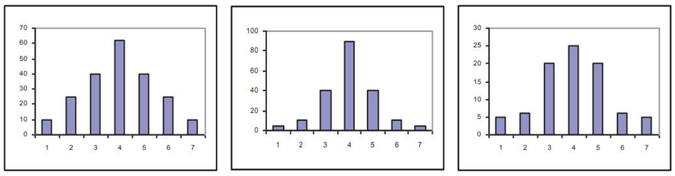
- **Vnitřní hradby** – Za odlehlé pozorování lze považovat takovou hodnotu x_i , která je od dolního, resp. horního kvartilu vzdálená více než 1,5 násobek interkvartilového rozpětí. Tedy: $[(x_i < x_{0,25} - 1,5 \cdot IQR) \vee (x_i > x_{0,75} + 1,5 \cdot IQR)] \Rightarrow x_i$ je odlehlým pozorováním.
- **z–souřadnice (z–skóre)** Za odlehlé pozorování lze považovat takovou hodnotu x_i, jejíž absolutní hodnota z-souřadnice je větší než 3, tj. hodnota, která je od průměru vzdálenější než 3s. Tedy: $z - skore_i = \frac{x_i - \bar{x}}{s}$
$|z - skore_i| > 3 \Rightarrow |\frac{x_i - \bar{x}}{s}| > 3s \Rightarrow x_i$ je odlehlým pozorováním.
- **$\mathbf{x_{0,5}}$–souřadnice ($\mathbf{x_{0,5}}$–skóre)** – : Za odlehlé pozorování lze považovat takovou hodnotu x_i, jejíž absolutní hodnota mediánové souřadnice je větší než 3, tj. hodnota, která je od mediánu vzdálenější než $3 \cdot 1,483 \cdot MAD$. Tedy: $x_{0,5} - skore_i = \frac{x_i - x_{0,5}}{1,483 MAD}$
$|x_{0,5} - skore_i| > 3 \Rightarrow |\frac{x_i - x_{0,5}}{1,483 MAD}| > 3 \Rightarrow |x_i - x_{0,5}| > 3 \cdot 1,483 MAD \Rightarrow x_i$ je odlehlým pozorováním.

13.5 Míra šikmosti a špičatosti

- **Výběrová šikmost a** (skewness) – vyjadřuje asymetrii rozložení hodnot proměnné kolem jejího průměru.
  - $a = 0$ – hodnoty proměnné jsou kolem jejího průměru rozloženy symetricky
  - $a > 0$ – u proměnné převažují hodnoty menší než průměr
  - $a < 0$ – u proměnné převažují hodnoty větší než průměr



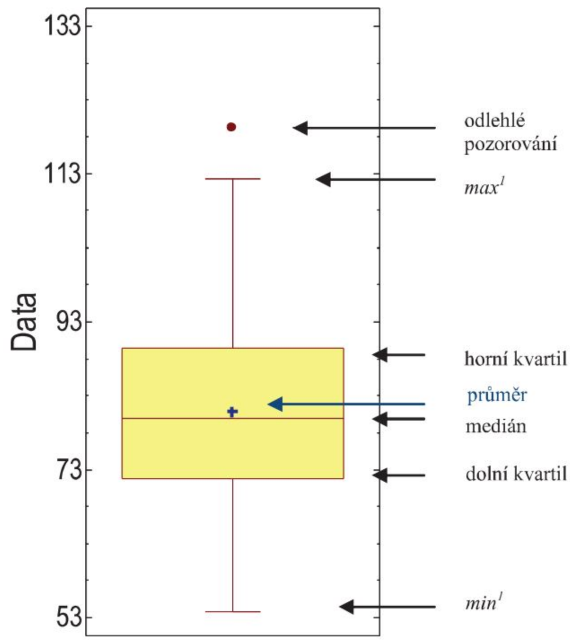
- **Souvislost mezi šikmostí a charakteristikami polohy**
  - Symetrické rozdělení: $\bar{x} = x_{0,5}$
  - Pozitivně zešikmené rozdělení: $\bar{x} > x_{0,5}$
  - Negativně zešikmené rozdělení: $\bar{x} < x_{0,5}$
- **Výběrová špičatost b** (kurtosis) – vyjadřuje koncentraci hodnot proměnné kolem jejího průměru.
  - $b = 0$ – špičatost odpovídá normálnímu rozdělení (bude definováno později)
  - $b > 0$ – špičaté rozdělení proměnné
  - $b < 0$ – ploché rozdělení proměnné



b=0 b>0 b<0

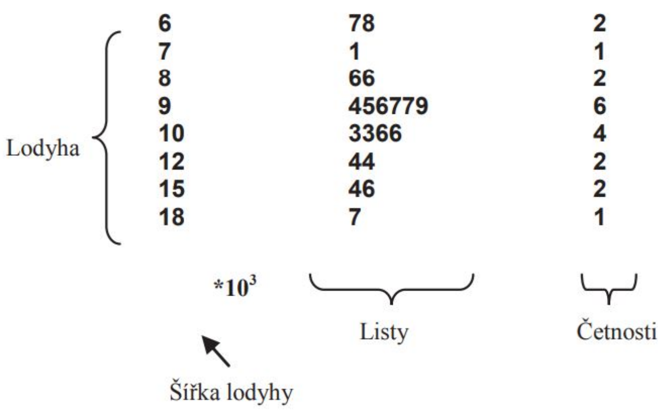
13.6 Grafické znázornění kvalitativní proměnné

- **Krabicový graf** (Box plot)
  - Odlehlá pozorování jsou znázorněna jako izolované body, konec horního (popř. konec dolního) vousu představují maximum (popř. minimum) proměnné po vyloučení odlehlých pozorování,„víko“ krabice udává horní kvartil, „dno“ dolní kvartil, vodorovná úsečka uvnitř krabice označuje medián.
  - Z polohy mediánu vzhledem ke „krabici“ lze dobře usuzovat na symetrii vnitřních 50% dat a my tak získáváme dobrý přehled o středu a rozptýlenosti proměnné.



- **Číslicový histogram** (Lodyha s listy, angl. Stem and leaf plot)
  - Jak jsme si ukázali, výhodou krabicového grafu je jeho jednoduchost, někdy nám však chybí informace o konkrétních hodnotách proměnné.
  - Chtěli bychom proto nějak přehledně zapsat číselné hodnoty výběru a k tomu nám slouží právě číslicový histogram.
  - Navíc nám tento graf dává dobrou představu o šikmosti proměnné.
  - Příklad: *Představme si proměnnou představující průměrné měsíční platy zaměstnanců ve státní správě.*

Průměrný měsíční plat [Kč]
10 654, 9 765, 8 675, 12 435, 9 675, 10 343, 18 786, 15 420, 8 675, 7 132, 6 732,6 878, 15 657, 9 754, 9 543, 9 435, 10 647, 12 453, 9 987, 10 342.



* Pro naší informaci nejsou tak důležité koruny ani desetikoruny rozdílu. V tomto případě se nám jedná přinejmenším o stokoruny.
* Co kdybychom tedy informaci o „nedůležitých“ řádech zanedbali a znázornili setříděná data pouze na základě vyšších řádů? My jsme se rozhodli, že důležitý řád jsou pro nás stokoruny.
* Hodnoty stojící o řád výš (v našem případě tisíce) zapíšeme setříděné pod sebe, tak, že tvoří jakýsi stonek (**lodyhu**), přičemž pod graf uvedeme tzv. **šířku lodyhy**, která udává koeficient, jímž se hodnoty uvedené v grafu násobí.
* Druhý sloupec grafu, **listy**, budou tvořit číslice, reprezentujíci zvolený „důležitý“ řád, zapisované do příslušných řádků (opět seřazené podle velikosti).
* Třetí sloupec udává absolutní četnosti příslušné daným řádkům.

13.7 Statistické charakteristiky kvalitativních proměnných

13.7.1 Nominální proměnná

Nominální proměnná nabývá v rámci souboru různých, avšak rovnocenných kategorií. Počet těchto kategorií nebývá příliš vysoký, a proto první statistickou charakteristikou, kterou k popisu proměnné použijeme je četnost.

- **Četnost $\mathbf{n_i}$** (absolutní četnost, „frequency“) – je definována jako počet výskytu dané varianty kvalitativní proměnné. V případě, že kvalitativní proměnná ve statistickém souboru o rozsahu n hodnot nabývá k různých variant, jejichž četnosti označíme $n_1, n_2, ..., n_k$, musí zřejmě platit $n_1 + n_2 + ... + n_k = \sum_{i=1}^{k} n_i = n$.
  Chceme-li vyjádřit, jakou část souboru tvoří proměnné s některou variantou, použijeme pro popis proměnné relativní četnost.
- **Relativní četnost $\mathbf{p_i}$** („relative frequency“) je definována jako $p_i = \frac{n_i}{n}$, popř. $p_i = \frac{n_i}{n} \cdot 100[\%]$. Pro relativní četnosti musí platit $p_1 + p_2 + ... + p_k = \sum_{i=1}^{k} p_i = p$.
  Při zpracování kvalitativní proměnné je vhodné četnosti i relativní četnosti uspořádat do tzv. **tabulky rozdělení četnosti** („frequency table“)

TABULKA ROZDĚLENÍ ČETNOSTI		
Hodnoty x_i	**Absolutní četnosti**	**Relativní četnosti**
	n_i	**p_i**
x_1	n_1	p_1
x_2	n_2	p_2
x_k	n_k	p_k
Celkem	$\sum_{i=1}^{k} n_i = n$	$\sum_{i=1}^{k} p_i = 1$

- **Modus** – s definujeme jako název varianty proměnné vykazující nejvyšší četnost.

- **Grafické znázornění nominální proměnné**
  - **Histogram** – je klasickým grafem, v němž na jednu osu vynášíme varianty proměnné a na druhou osu jejich četnosti. Jednotlivé hodnoty četností jsou pak zobrazeny jako výšky sloupců (obdélníků, popř. hranolů, kuželů...)
  - **Výsečový graf** – prezentuje relativní četnosti jednotlivých variant proměnné, přičemž jednotlivé relativní četnosti jsou úměrně reprezentovány plochami příslušných kruhových výsečí. (Změnou kruhu na elipsu dojde k trojrozměrnému efektu.)

13.7.2 Ordinální proměnná

Ordinální proměnná, stejně jako proměnná nominální, nabývá v rámci souboru různých slovních variant, avšak tyto varianty mají přirozené uspořádání, tj. můžeme určit, která je „menší“ a která „větší“.

- **Kumulativní četnost $\mathbf{m_i}$** i („cumulative frequency“) – definujeme jako počet hodnot proměnné, které nabývají varianty nižší nebo rovné i-té variantě.
  Uvažte např. proměnnou „známka ze statistiky“, která nabývá variant: „výborně“, „velmi dobře“, „prospěl“, „neprospěl“, pak např. kumulativní četnost pro variantu „prospěl“ bude rovna počtu studentů, kteří ze statistiky získali známku „prospěl“ nebo lepší.
  Jsou-li jednotlivé varianty uspořádány podle své „velikosti“(„$x_1 < x_2 < ... < x_k$“), platí $m_i = \sum_{i=1}^{i} n_j$, Je tedy zřejmé, že kumulativní četnost k–té („nejvyšší“) varianty je rovna rozsahu proměnné $\breve{}mk = n$.
- **Kumulativní relativní četnost $\mathbf{F_i}$** („cumulative relative frequency“) – vyjadřuje jakou část souboru tvoří hodnoty nabývající i-té a nižší varianty. $F_i = \sum_{j=1}^{i} p_j$, což není nic jiného než relativní vyjádření kumulativní četnosti: $F_i = \frac{m_i}{n}$.
  Obdobně jako pro nominální proměnné, můžeme i pro proměnné ordinální prezentovat statistické charakteristiky pomocí tabulky rozdělení četnosti. Ta obsahuje ve srovnání s tabulkou rozdělení četností pro nominální proměnnou navíc hodnoty kumulativních a kumulativních relativních četností.

TABULKA ROZDĚLENÍ ČETNOSTÍ				
Hodnoty x_i	Absolutní četnost n_i	Relativní četnost p_i	Kumulativní četnost m_i	Kumulativní relativní četnost F_i
x_1	n_1	p_1	$m_1 = n_1$	$F_1 = p_1$
x_2	n_2	p_2	$m_2 = n_1 + n_2 = m_1 + n_2$	$F_2 = p_1 + p_2 = F_1 + p_2$
x_k	n_k	p_k	$m_k = m_{k-1} + n_k = n$	$F_k = F_{k-1} + p_k = 1$
Celkem	$\sum_{i=1}^{k} n_i = n$	$\sum_{i=1}^{k} p_i = 1$	-----	-----

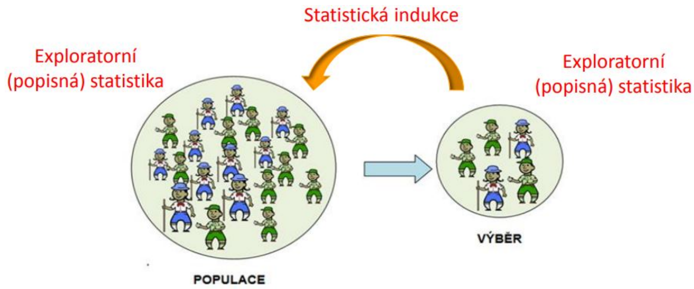
- **Grafické znázornění ordinální proměnné**
  - **Lorenzova křivka** ((polygon kumulativních četností, Galtonova ogiva, S křivka) – S křivka) je spojnicovým grafem, který získáme tak, že na vodorovnou osu vynášíme jednotlivé varianty proměnné v pořadí od „nejmenší“ do „největší“ a na svislou osu příslušné hodnoty kumulativních četností. Znázorněné body spojíme úsečkami.

  - **Paretova analýza** – lze formulovat tak, že 80% následků pramení z 20% příčin (20% lidí vlastní 80% celkového bohatství). V praxi pak bývá snahou nalézt toto malé spektrum příčin (životně důležitá menšina), které tak významně ovlivňuje výsledek.

14 Metody statistické indukce. Intervalové odhady. Princip testování hypotéz.

14.1 Statistická indukce

Statistická indukce je metoda, která dovoluje stanovit vlastnost celku (**základního souboru**) na základě pozorování jeho částí (**náhodného výběru**).



Základní soubor (populace)

- Je množina všech teoreticky možných objektů (např. jedinců) v uvažované situaci = statistický soubor, který je vymezen cílem výzkumu a pro který vyvozujeme závěry výzkumného šetření.
- Charakterizuje se **parametrem**, což je např. výška, váha, IQ, atp.
- Má konečný nebo nekonečný (hypotetický) **rozsah**, který je dán N (např.: N = 150 lidí, opic, rostlin,...).

Výběrový soubor (výběr)

- Je část populace vybrané na základě předem stanovených kritérii resp. pravidel (podmnožina základního souboru).
- **O náhodném výběru** uvažujeme, když splňuje dvě základní vlastnosti: pravděpodobnost zařazení do vzorku je pro všechny statistické jednotky populace nenulová a statistické jednotky jsou do vzorku vybrané nezávisle jedna od druhé.
- **O reprezentativním výběru** uvažujeme, když výběrový soubor dobře odráží strukturu celého zkoumaného souboru.

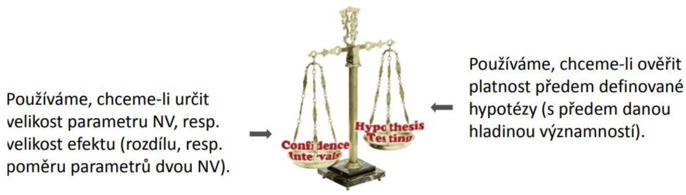
Principy statistického usuzování

1. Statistické usuzování znamená zobecňování z výběrových statistik na parametry rozdělení.
2. Abychom mohli provést statistické usuzování, musíme mít nějakou teorii, jež popisuje náhodné chování sledovaných proměnných.
3. Existují dva typy výběrových chyb: náhodné výběrové chyby a systematické chyby. Získáním náhodného výběru zmenšujeme systematickou chybu a získáváme podklad pro odhad náhodné výběrové chyby.

4. Výběrová rozdělení statistik jsou teoretická pravděpodobnostní rozdělení, která popisují vztah mezi výběrovou statistikou a populací.
5. Směrodatná odchylka výběrového rozdělení statistiky (odhad parametru) se nazývá směrodatná chyba. Odhaduje náhodnou výběrovou chybu vypočítané statistiky (odhadu parametru).
6. Jak roste velikost výběru, výběrová chyba a směrodatná chyba se zmenšují.
7. Směrodatná chyba se používá k získání intervalového odhadu parametrů i k testování hypotéz o parametrech rozdělení.

14.2 Základy metody statistické indukce

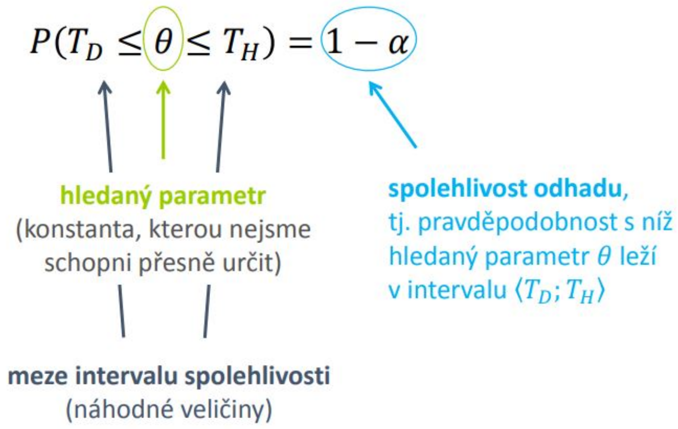
- **Intervalové odhady** (confidence intervals) – umožnují odhadnout nejistotu v odhadu parametru náhodné veličiny
- **Testování hypotéz**(hypothesis testing) – umožnuje posoudit, zda experimentálně získaná data nepopírají předpoklad, který jsem **před** provedením testování učinili.



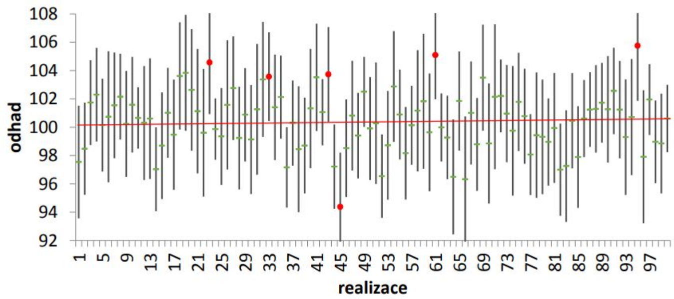
14.2.1 Intervalové odhady

- V praktických aplikacích často určujeme **odhad příslušného parametru** pomocí intervalového odhadu.
- Tento odhad je reprezentován intervalem $< t_D, t_H >$, v němž hledaný parametr leží s předem určenou pravděpodobností (spolehlivostí), kterou označujeme $(1 - \alpha)$.
- neboli parametr populace aproximujeme intervalem, v němž s velkou pravděpodobností příslušný populační parametr leží.

- **Interval spolehlivosti**(konfidenční interval) pro parametr θ se spolehlivostí $1-\alpha$,kde $\alpha \in< 0; 1 >$, je taková dvojice statistik (T_D, T_H), že $P(T_D \leq \theta \leq T_H) = 1 - \alpha$.



- **Intervalový odhad** t_D, t_H je jednou z realizací intervalu spolehlivosti.
- Požadavky na interval spolehlivosti:
  - Co **největší spolehlivost** odhadu.
  - Co **nejmenší šírka** intervalu spolehlivost. (S rostoucí šířkou intervalového odhadu klesá významnost získané informace.)
- S rostoucí spolehlivostí se zvětšuje šířka intervalového odhadu a tím klesá významnost takto získané informace.
- S rostoucím rozsahem výběru se šíčka intervalového odhadu snižuje.
- Typy intervalů spolehlivosti:
  - **oboustranné**
    $$P(\theta < T_D) = P(\theta > T_H) = \frac{\alpha}{2}$$
    Tyto dvě podmínky zaručují, že $P(T_D \leq \theta \leq T_H) = 1 - \alpha$
  - **jednostranné** (odhadujeme–li například délku života nějakého zařízení, je pro nás důležitá pouze dolní mez)
    * **levostranné** $P(\theta \geq T_D^*) = 1 - \alpha$
    * **pravostranné** $P(\theta \leq T_H^*) = 1 - \alpha$



Obrázek 6: Co to znamená, že spolehlivost odhadu je $1-\alpha$?
Simulace 100 intervalových odhadů střední hodnoty (spolehlivost $0,95$) získaných na základě opakovaných výběrů o rozsahu 30 z populace se střední hodnotou 100. 6 intervalů ze 100 neobsahuje skutečnou střední hodnou

14.3 Jak najít intervalový odhad parametru θ?

Obecně:

1. Zvolíme vhodnou výběrovou charakteristiku $T(\mathbf{X})$, jejíž rozdělení známe.
2.

$$P(\frac{x_\alpha}{2} \leq T(\mathbf{X}) \leq x_{1-\frac{alpha}{2}}) = 1-\alpha,$$
$$P(T(\mathbf{X}) \leq x_{1-\alpha}) = 1-\alpha,$$
$$P(T(\mathbf{X}) \geq x_{\alpha}) = 1-\alpha.$$

14.4 Testování hypotéz

- **Statistická hypotéza** – předpoklad (tvrzení) o rozdělení náhodné veličiny
- Zdrojem statistických hypotéz jsou například předchozí zkušenosti, teorie, kterou je třeba doložit, požadavky na kvalitu produktu, dohady založené na náhodném pozorování.
- Příklady statistických hypotéz:
  - Střední životnost žárovek Ed je nižší než výrobcem udávaných 5 let.
  - Mortalita je u laparoskopických operací nižší než u operací konvenčních.
  - Průměrné výsledky srovnávacích testů závisí na typu absolvované střední školy.
  - Pořízený datový soubor je výběrem z populace mající normální rozdělení.
- **Parametrická statistická hypotéza** – tvrzení ohledně efektu
  - Hypotézy o parametru jedné populace (o střední hodnotě, rozptylu, mediánu, parametru binomického rozdělení,...)
  - Hypotézy o parametrech dvou populací (srovnávací testy)

  - Hypotézy o parametrech více než dvou populací (ANOVA, Kruskalův–Wallisův test,...)

- **Neparametrická statistická hypotéza** – tvrzení o jiné vlastnosti rozdělení náhodné veličiny než o jejím parametru (např. hypotézy o typu rozdělení NV, hypotézy o závislosti NV,...)

Příklad, ověření, zda statistická hypotéza je pravdivá: Domníváme se, že střední hodnota obsahu cholesterolu v krvu je u české populace 4,7 mmol/l.

$$H_0 : \mu = 4.7$$
$$H_A : \mu \neq 4.7$$

Jak tento předpoklad ověřit?

- Zjistíme údaje o obsahu cholesterolu v krvi u 100 náhodně vybraných Čechů.
- Průměrný obsah cholesterolu v krvi probandů (tj. jedinců, kteří jsou předmětem zkoumání) byl 5,4 mmol/l.

Jsou tyto výsledky v souladu s naší hypotézou?

- ◦ I kdyby byla testovaná hypotéza pravdivá, nelze očekávat, že průměrná hodnota pozorovaná ve výběru bude přesně 4,7 mmol/l.
- ◦ **Nulovou hypotézu zamítneme, pokud získané uspořádání výberu bude za předpokladu platnosti nulové hypotézy velmi nepravděpodobné**.

- Rozhodovací proces, v němž proti sobě stojí nulová a alternativní hypotéza.
- **Nulová hypotéza $\mathbf{H_0}$** – tvrzení, že efekt je nulový, resp. že neexistuje závislost, že data mají určitý typ rozdělení,...
- **Alternativní hypotéza $\mathbf{H_A}$ ($\mathbf{H_1}$)** – tvrzení, popírající hypotézu nulovou (obvykle to, co chceme dokázat)

14.4.1 Klasický přístup při testování hypotéz

1. Formulujeme **nulovou a alternativní hypotézu**.
2. Zvolíme tzv. **testovací statistiku**, tj. výběrovou charakteristiku, jejíž rozdělení závisí na testovaném parametru θ. (Rozdělení testované statistiky za předpokladu platnosti nulové analýzy nazýváme **nulové rozdělení**.)
3. Ověříme předpoklady testu !
4. Určíme **kritický obor** W^*, tj. množinu, v níž se, za předpokladu platnosti H_0, hodnoty testované statistiky vyskytují s velmi malou pravděpodností.

    - Doplňkem k W^* je tzv. **obor přijetí** V^*.

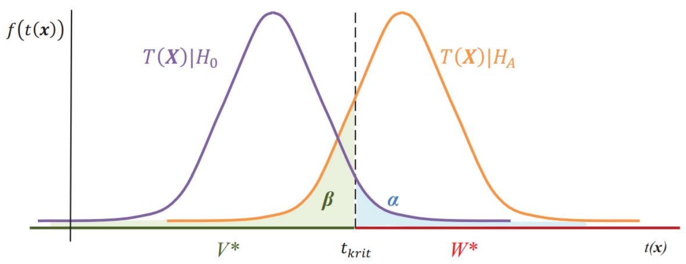
- Hranici mezi kritickým oborem a oborem přijetí označujeme jako **kritická hodnota testu** t_{krit}.

5. Na základě konkrétní realizace výběru určímě **pozorovanou hodnotu** X_{OBS} testované statistiky.
6. Na základě vztahu mezi X_{OBS} a t_{krit} rozhodneme o výsledku testu („Zamítáme H_0.“ nebo „Nezamítáme H_0.“)

14.4.2 Chyba I. a II. druhu

		Výsledek testu	
		Nezamítáme H_0	Zamítáme H_0
Skutečnost	Platí H_0	Správné rozhodnutí $1-\alpha$ (spolehlivost testu)	**Chyba I. druhu** α (hladina významnosti)
	Platí H_A	**Chyba II. druhu** β	Správné rozhodnutí $1-\beta$ (síla testu)

Jestliže nulová hypotéza je ve skutečnosti platná a my ji přesto zamítneme, dopouštíme se chyby, označované jako chyba **I. druhu**. Pravděpodobnost, že k takovémuto pochybení dojde, nazýváme **hladina významnosti** a označujeme ji α. Platí-li nulová hypotéza a my jsme ji nezamítli, rozhodli jsme správně. Pravděpodobnost tohoto rozhodnutí označujeme $1-\alpha$ a nazýváme ji **spolehlivost testu**. Správným rozhodnutím je rovněž zamítnutí nulové hypotézy v případě, že je platná hypotéza alternativní. Tohoto rozhodnutí se dopouštíme s pravděpodobností $1-\beta$, což bývá označováno jako síla testu. **Chybou II. druhu** je nezamítnutí nulové hypotézy v případě, že je platná hypotéza alternativní. Pravděpodobnost této chyby označujeme β.



Obrázek 7: Demonstrace pravděpodobností chyb I. a II. druhu

Při testování hypotéz se samozřejmě snažíme postupovat tak, abychom minimalizovali obě chyby, tj. dosáhnout vysoké síly testu (nízkého β) při co nejnižší hladině významnosti α. To však není možné, neboť snížením β se zvýší hladina významnosti α a naopak. Proto je třeba najít kompromis mezi požadavky na α a β.

Parametrická statistická hypotéza

Jednovýběrové testy

- Test o střední hodnotě (z–test, t–test).
- Test o rozptylu.
- Test o parametru binomického rozdělení.
- Test o mediánu (Wilcoxonův test, Mediánový test).

Dvouvýběrové testy

- Test o shodě dvou středních hodnot (t–test, Aspinové–Welchův test).
- Test o shodě rozptylů (F–test).
- Test o shodě parametrů dvou binomických rozdělení (test homogenity dvou binomických rozdělení).
- Test o shodě mediánů (Mannův–Whitneyův test).
- Párové testy (párový t–test, párový znaménkový test).

Vícevýběrové testy

- Testy shody rozptylů (Bartletův test, Hartleyův test, Cochranův test, Leveneův test).
- Analýza rozptylu (tzv. ANOVA, tj. shody středních hodnot) – post hoc analýza pro analýzu rozptylu
- Kruskalův–Wallisův test (test shody mediánů) – post hoc analýza Kruskal–Wallisův test

ANOVA

- Test umožnující srovnání průměrů více než dvou výběrových souborů
- Můžeme například zkoumat, zda
  - typ absolvované střední školy ovlivňuje počet bodů dosažených studenty u přijímací zkoušky z matematiky,
  - použitá medikace ovlivňuje krevní tlak pacientů,
  - typ použitého hnojiva ovlivňuje výnosy určité plodiny,
  - pracovní výkon dělníka závisí na umístění stroje, apod.