Konečné automaty

Konečný automat je systém, který může nabývat konečně mnoho stavů. Daný stav se mění na základě vnějšího podnětu s tím, že pro daný stav a daný podnět je jednoznačně určeno jaký stav je následující.

Konečný automat (KA) tvoří množina stavů, vstupní abeceda, 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.

Konečný automat je uspořádaná pětice A (Q, Σ, δ, g_0, F), kde:

- Q je stavový prostor - konečná neprázdná **množina stavů**
- Σ je **vstupní abeceda** - konečná neprázdná množina vstupních symbolů (množina všech slov nad abecedou je značena jako Σ*)
- δ je **přechodová funkce** - zobrazení δ: Q × Σ → Q, které na základě stavu a symbolu abecedy vrátí další stav
- g_0 je **počáteční stav/y** - $g_0 \in Q$
- F je **množina koncových stavů** - $F \subseteq Q$

Slovo přijaté automatem je taková sekvence symbolů (ze vstupní abecedy), pro kterou automat skončí v koncovém stavu

Jazyk je množina slov, pro které automat skončí v koncovém stavu

Regulární jazyk – jazyk, který lze popsat konečným automatem

- Může být popsán regulárním výrazem
- Může být vygenerován regulární gramatikou
- Jazyk je regulární právě tehdy, když je množina kvocientů podle slov tohoto jazyka konečná.

Nedeterministický KA (NKA) – rozdíl proti běžnému deterministickému KA:

- V daném stavu máme při čtení vstupního znaku možnost přechodu do více různých
- stavů, nebo také nemusíme mít žádnou možnost přechodu. Dokonce můžeme přecházet po hranách značených symbolem ε bez čtení ze vstupního slova – tzv. ε-přechody.
- Je povolen více než jeden počáteční stav.

Pro každý NKA A existuje ekvivalentní (deterministický) KA A', tj. rozpoznávající stejný jazyk L(A) = L(A').
NKA rozpoznávají právě regulární jazyky a jsou v tomto smyslu ekvivalentní DKA.

Regulární výrazy

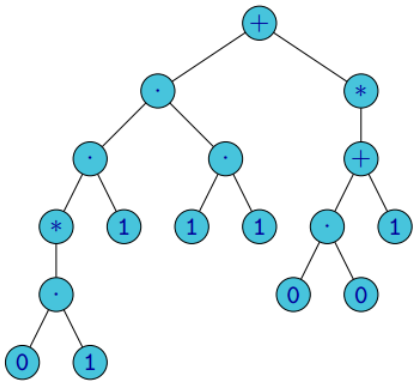
Regulární výraz je textový řetězec, který umožňuje popsat nějakou množinu slov (jazyk nad abecedou). Regulární výrazy také můžeme chápat jako jednoduchý způsob, jak popsat konečný automat umožňující generovat všechna možná slova patřící do daného jazyka. Pomocí regulárních výrazů můžeme definovat regulární jazyky. Regulární jazyk je takový jazyk, který je přijímán nějakým konečným automatem. Každý regulární jazyk je možné popsat nějakým regulárním výrazem.

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 Ø.

Jestliže A, B jsou regularni vyrazy, pak i (A + B), (A · B), (A∗) jsou regularni vyrazy:

- (A + B) . . . označuje sjednoceni jazyků
- (A· B) . . . označuje zřetězeni jazyků
- (A∗) . . . označuje iteraci jazyka

Strukturu regulárního výrazu si můžeme znázornit abstraktním syntaktickým stromem:



Uzávěrové vlastnosti třídy regulárních jazyků

Třídu regulárních jazyků značíme REG a je to množina všech regulárních jazyků.

Uzavřenost množiny nad operací znamená, že výsledek operace s libovolnými prvky z množiny bude opět spadat do dané množiny

Množina regulárních jazyků je uzavřena vůči:

- Sjednocení (spustí se najednou, konečné stavy jsou sjednocením z obou automatů)
- Zřetězení (přidat epsilon přechod)
- iteraci
- průniku (spustí se najednou, konečné stavy jsou průnikem z obou automatů)
- doplňku (nepřijímající stavy se stanou přijímacími)
- množinovému rozdílu
- zrcadlovému obrazu (přehodíme orientaci, z počátečních uděláme koncové stavy a naopak)
- levému i pravému kvocientu množiny.

Bezkontextové gramatiky a jazyky

Bezkontextová gramatika definuje **bezkontextový jazyk** - jazyk jehož slova se tvoří nezávisle na okolí (na předchozích krocích). Jedná se o obecnější třídu jazyků (tj. máme v ní k dispozici silnější popisné prostředky - odvozovací pravidla) než byly regulární jazyky. S bezkontextovými jazyky a gramatikami se hojně setkáváme při syntaktické analýze textu, třeba zdrojových kódů programovacích jazyků. Bezkontextovou gramatiku tvoří **neterminály** (proměnné), **terminály** (konstanty) a **pravidla**, které každému neterminálu definují, nač je lze přepsat. Jeden neterminál označíme jako startovní, tam začínáme a podle pravidel je dál přepisujeme na výrazy složené z terminálu a neterminálu. Jakmile už není co přepisovat, výraz obsahuje už jen terminály, získali jsme slovo.

Jazyk L je bezkontextový, jestliže existuje bezkontextová gramatika G taková, že L(G) = L (L(G) je jazyk generovaný gramatikou G)

Definice

Bezkontextová gramatika je čtveřice G = (Π,Σ, S, P), kde

- Π je konečná množina neterminálních symbolů (neterminálů)
- Σ je konečná množina terminálních symbolů (terminálů), přičemž Π ∩ Σ = Ø
- S ∈ Π je počáteční (startovací) neterminál
- P je konečná množina pravidel typu A → β, kde
    - – A je neterminál, tedy A ∈ Π
    - - β je řetězec složený z terminálů a neterminálů, tedy β ∈ (Π∪Σ)*.

Příklad:

S* -> O | (O)

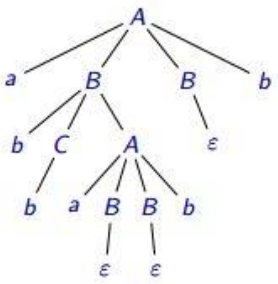
O -> O+O | O*O | S | 1 | 2 | 3 | 4 | 5

Bezkontextová gramatika generující početní příklady pro sčítání a násobení do pěti.

Derivace slova je odvození slova pomocí gramatiky, tedy záznam postupných přepisů od startovního neterminálu po konečné slovo. Pokud přepisujeme nejprve levé neterminály, jde o **levou derivaci**. Pokud jedeme zprava jedná se o **derivaci pravou**.

Derivační strom je 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.

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



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

Řekneme, že bezkontextová gramatika G je **jednoznačná**, jestliže každé slovo z L(G) má právě jedno levé odvození (tj. právě jeden derivační strom). V opačném případě je G nejednoznačná (či víceznačná).

Bezkontextová gramatika je v **Chomského normální formě**, jestliže její pravidla jsou výhradně ve tvarech X –→ YZ a X → a.

Zásobníkové automaty a vztah k bezkontextovým gramatikám

Zásobníkový automat slouží k rozpoznání bezkontextových jazyků. Konečný automat si pamatuje akorát aktuální stav. Neví kolik znaku přečetl a které to byly. A to právě potřebujeme k rozpoznání bezkontextového jazyku. Zásobníkový automat je v podstatě konečný automat rozšířený o zásobník.

Zásobníkový automat M je definován jako šestice M = (Q,Σ, Γ, δ, q0, Z0), kde

- Q je konečná neprázdná množina stavů,
- Σ je konečná neprázdná množina vstupních symbolů (vstupní abeceda),
- Γ je konečná neprázdná množina zásobníkových symbolů (zásobníková abeceda),
- q0 ∈ Q je počáteční stav,
- Z0 ∈ Γ je počáteční zásobníkový symbol a
- δ je zobrazení množiny Q×(Σ∪{ε})×Γ do množiny všech konečných podmnožin množiny Q × Γ∗.

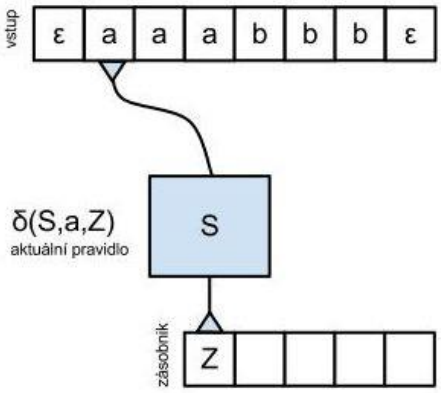
Zásobníkový automat na základě aktuálního znaku na vstupu, prvního (posledně zapsaného) znaku v zásobníku a aktuálního stavu, změní svůj stav a přepíše znak v zásobníku, a to podle daného pravidla δ. Zásobník na začátku výpočtu obsahuje právě (jeden) počáteční zásobníkový symbol.

Výpočet je tedy posloupnost konfigurací, kde každá jednotlivá konfigurace zachycuje

- (aktuální) stav řídicí jednotky,
- obsah vstupní pásky, který zbývá přečíst,
- obsah zásobníku (zapsaný jako řetězec symbolů; nejlevější symbol odpovídá
- vrcholu zásobníku).

Výpočet zásobníkového automatu je přijímajícím, jestliže skončí v konfiguraci (q, ε, ε), tedy když se přečte celé vstupní slovo a vyprázdní se zásobník (přičemž na dosaženém stavu nezáleží).

Nedeterministické zásobníkové automaty rozpoznávají právě bezkontextové jazyky (a jsou takto ekvivalentní bezkontextovým gramatikám).



pravidlo	vysvětlení
δ(S,a,Z)={(S,AZ)}	**přidání** prvku do zásobníku Do zásobníku se přidá A.
δ(S,a,Z)={(S,A)}	**přepsání** prvku v zásobníku První prvek zásobníku se přepíše na A.
δ(S,a,Z)={(S, ε)}	**smazání** prvku v zásobníku První prvek zásobníku se smaže (nahradí prázdným slovem ε)
δ(S,a,Z)={(T, Z)}	**změna stavu** Stav S se změní na stav T. Pro stav T budou platit pravidla typu δ(T,...
δ(S,a,Z)= ø	**pád automatu** Ukončení výpočtu, slovo nebylo přijato.

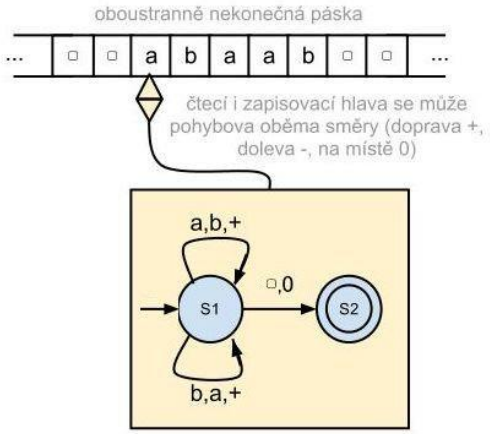
Turingův stroj

Historicky prvním „univerzálním programovacím jazykem“ jsou Turingovy stroje. Alan Turing se snažil návrhem toho, čemu říkáme Turingovy stroje, formalizovat práci „výpočtáře“, pracujícího s tužkou, gumou a papírem a svou omezenou pamětí. Je to teoretický model počítače a využívá se pro modelování algoritmů.

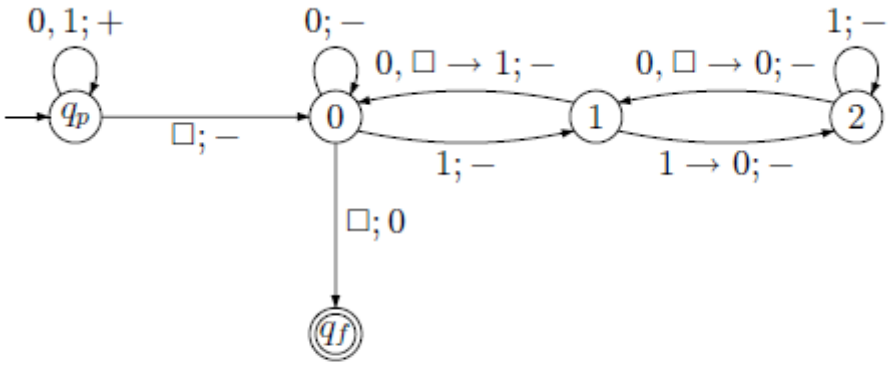
Turingův stroj je podobný konečnému automatu, významný rozdíl je ale popsán v následujícím:

- **páska je oboustranně nekonečná**; je na ní na začátku zapsáno vstupní slovo, na jehož první pozici stojí hlava, a ostatní buňky jsou prázdné, tj. je v nich zapsán speciální prázdný znak □,

- hlava (spojená s konečnou řídicí jednotkou) se může pohybovat po pásce oběma směry a je nejen **čtecí, ale i zapisovací** – symboly v buňkách pásky je tedy možné přepisovat, a to obecně i jinými než vstupními symboly.



Turingův stroj lze zadat **seznamem instrukcí**: δ(stav, znak) = (novy_stav, novy_znak, posun).



Obrázek 6.2: Turingův stroj realizující násobení třemi

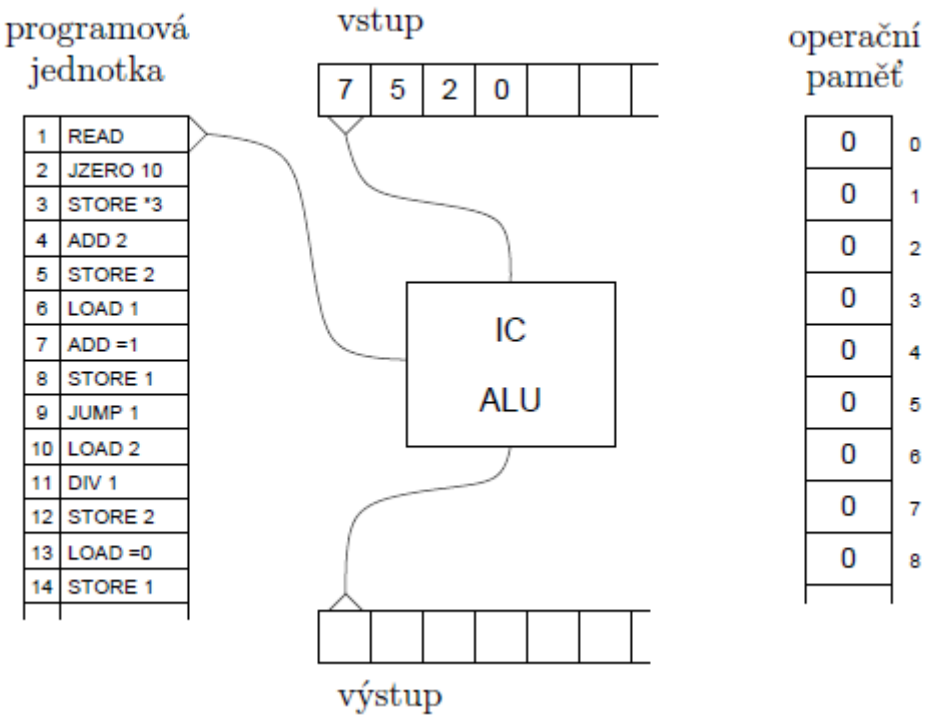
RAM stroj

- Random Access Machine model
- RAM stroje už vycházejí ze skutečných počítačů. RAM stroje mají paměť, pracují s přirozenými čísly (1,2,3,...) a instrukce se podobají klasickým příkazům
- Dá se říci, že se jedná o jednoduchou abstrakci reálného procesoru s jeho strojovým kódem, pracujícího s lineárně uspořádanou pamětí, tj. posloupností „buněk“ s adresami

RAM se skládá z těchto částí:

- **Programová jednotka**, ve které je uložen program tvořený posloupností instrukcí
- **Neomezená pracovní paměť (operační),** každá buňka může obsahovat libovolné číslo a buňky jsou číslovány od 0. Z buněk lze číst i zapisovat; buňka 0 je *pracovní registr* a buňka 1 je *indexový registr*.
- **Vstupní páska –** z této pásky lze pouze číst, čtecí hlava přečte buňku a posune se doprava
- **Výstupní páska –** je možné pouze sekvenčně zapisovat

- **Centrální jednotka –** obsahuje programový registr ukazující, která instrukce má být v daném okamžiku prováděna



Tvary operandů

- =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

Instrukce vstupu a výstupu

- **READ:** do pracovního registru se uloží číslo, které je v aktuální buňce vstupní pásky
- **WRITE:** na výstupní pásku se zapíše obsah pracovního registru

Instrukce přesunu v paměti

- **LOAD op:** do pracovního registru se načte hodnota operandu
- **STORE op:** hodnota operandu se přepíše obsahem pracovního registru

Instrukce aritmetických operací

- **ADD op:** číslo v prac. Registru se zvýší o hodnotu operandu
- **SUB op:** od prac. Registru se odečte hodnota operandu
- **MUL op:** vynásobení
- **DIV op:** prac. Registr se celočíselně vydělí hodnotou operandu

Instrukce skoku

- **JUMP návěstí:** skok na instrukci
- **JZERO návěstí:** je-li obsahem prac. Registru 0, provede se skok
- **JGTZ návěstí:** je-li číslo prac. Registru kladné, provede se skok

- **HALT – výpočet je ukončen**

Složitost algoritmu, asymptotické odhady

Složitost algoritmu

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. Jde o tzv.:

- **časová složitost** - sleduje nároky algoritmu na čas. Jak dlouho trvá výpočet s tímto algoritmem? Dočkají se výsledku alespoň mé vnoučata?
- **prostorová složitost** - sleduje nároky algoritmu na paměť. Je schopen provést tento algoritmus můj počítač s omezenou pamětí?

Pro určení složitosti algoritmu využíváme RAM stroje. Časovou složitostí algoritmu A rozumíme časovou složitost RAMu implementujícího A; podobně pro paměťovou složitost. Víme, že (časová nebo paměťová) složitost algoritmu je funkcí velikosti vstupu a vztahuje se k nějakému referenčnímu modelu počítačů; v našem případě jsme použili RAMy. 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.

Asymptotické odhady

Asymptonický odhad složitosti je další zobecnění složitosti algoritmu. Výpočet přesné funkce složitosti je pro komplikovanější algoritmy příliš náročný a navíc zbytečný. Pro představu o náročnosti algoritmu nám stačí vědět, jak rychle funkce roste.

U předchozího příkladu nás tedy nezajímá, kolik přesně instrukci se provedlo vně či vně cyklu, důležité je že funkce roste lineárně. Jak moc je lineární funkce nakloněná či zvednutá, není podstatné, protože každá kvadratická funkce ji nakonec předběhne. Tedy kvadratická složitost se při dostatečně velkém vstupu, stane vždy horší.

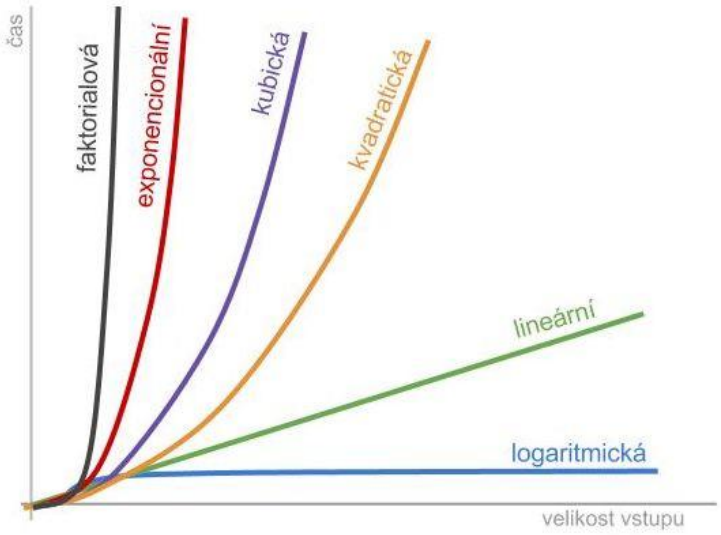
Funkce f(n) a konstanta krát funkce c * f(n) jsou ekvivalentní – stejně rychle rostoucí.

V souvislosti s asymptotickými odhady složitosti se používají tyto zápisy:

- $f \in O(g)$ . . . . f roste nejvýše tak rychle jako g = f je asymptoticky ohraničena funkcí g shora
- $f \in o(g)$ . . . . f roste (striktně) pomaleji než g = f je asymptoticky ohraničena funkcí g shora ostře
- $f \in \Theta(g)$ . . . . f roste stejně rychle jako g

Pokud takto "zaokrouhlíme" a zjednoduším funkci složitosti, mluvíme pak o složitosti podle typu funkce.

- $f(n) \in \Theta(\log n)$ . . . . logaritmická funkce (složitost)
- $f(n) \in \Theta(n)$ . . . . lineární funkce (složitost)
- $f(n) \in \Theta(n2)$ . . . . kvadratická funkce (složitost)
- $f(n) \in O(nc)$ pro nějaké $c > 0$ . . . . polynomiální
- $f(n) \in \Theta(cn)$ pro nějaké $c > 1$ . . . . exponenciální



Algoritmicky nerozhodnutelné problémy

Problém je algoritmicky řešitelný, pokud existuje algoritmus, který pro libovolný vstup svůj výpočet skončí a vydá výsledek. V případě ANO/NE problému se používá pojem **algoritmicky rozhodnutelný.**

Pokud tedy najdu takový vstup problému, na kterém si všechny dosavadní algoritmy řešící tento problém vylámou zuby, můžu tento problém nazvat neřešitelný. Důkaz neřešitelnosti lze provést také skrze jiné - už dokázané - problémy.

Je běžnou praxí, že když pro konkrétní problém P dokážeme neexistenci Turingova stroje, který by jej rozhodoval, řekneme, že jsme dokázali, že P je nerozhodnutelný

Příklady nerozhodnutelných problémů:

Diagonální problém zastavení: Vstupem je kód(M), kde M je Turingův stroj. Otázka je, zda-li se zastaví M na svůj kód?

Problém 1 je algoritmicky převeditelný na problém 2, pokud existuje převádějící algoritmus, který pro libovolný vstup problému 1 sestrojí vstup problému 2, přičemž platí, že odpověď na otázku problému 1 pro vstup je ANO právě tehdy, když odpověď na otázku problému 2 pro vstup je ANO. Nerozhodnutelnost dalších problémů se typicky ukazuje pomocí převeditelnosti z HP.

Název: **Eq-CFG (Ekvivalence bezkontextových gramatik)**

Vstup: Dvě bezkontextové gramatiky G1,G2.

Otázka: Platí L(G1) = L(G2)? Generují obě gramatiky stejný jazyk?

Název: **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)?

```
1 void Q(void) {
```

```
2   if (zastavi(Q))
3     while(1) { }
4 }
```

Riceova věta

Třídy složitosti problémů

Složitost algoritmů (viz otázka 3) VS složitost problémů

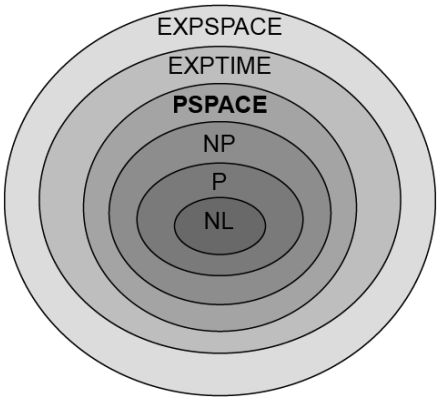
Intuitivně cítíme, že různé problémy mohou být různě „složité“; co to ale je ona složitost problémů? Víme-li např. o algoritmu (tedy RAMu), který daný problém řeší a má složitost Θ(n3), je toto Θ(n3) jen určitým horním odhadem „skutečné“ složitosti problému (můžeme říci „složitost problému je nejvýše kubická“). Je např. možné, že nalezneme jiný algoritmus, který řeší náš problém a který má složitost O(n2); tím jsme náš dosud známý odhad zlepšili (víme pak už, že „složitost problému je nejvýše kvadratická“).

Jistě nás teď napadne, že složitost problému bychom mohli definovat jako složitost „optimálního“ algoritmu, který daný problém řeší. Při exaktní definici onoho pojmu „optimální“ ovšem vzniknou jisté komplikace. Proto se pojem složitosti problému nedefinuje přímo, ale zavádějí se tzv. **třídy složitosti**.

Definice třídy složitosti:

Pro funkci f rozumíme *třídu časové složitosti* T (f), též značenou T(f(n)), množinu těch problémů, které jsou řešeny RAM stroji s časovou složitostí v O(f) (neboli problémy, pro které jsou algoritmy s časovou složitostí O(f)).

Příklad: Problém P patří do $T(n^2)$ (nebo časová složitost P je v $O(n^2)$) – existuje algoritmus s nejvýš kvadratickou časovou složitostí, který řeší problém P (existuje RAM s nejvýš kvadratickou časovou složitostí).



Třídy složitosti:

- **PTIME** – obsahuje všechny problémy řešitelné polynomiálními algoritmy (n^k, kde k je konstanta)

- **NPTIME** – třída všech ANO/NE problémů, které jsou rozhodovány nedeterministickými polynomiálními algoritmy
- **coNP**
- **NP-úplný** (NPC)
- **NP-těžké**
- **PSPACE** - problémy řešitelný s polynomiálním množstvím paměti
- **NPSPACE -** problémy řešitelné s polynomiálním množstvím paměti
- **EXPTIME** - problémy řešitelné v exponenciálním čase
- **EXPSPACE**

Třída PTIME a NPTIME

PTIME

Do této třídy složitosti spadají všechny problémy řešené s polynomiálními algoritmy. PTIME = $T(n^k)$. Třídu těchto problému považujeme za **zvládnutelnou**.

Tato třída je robustní. **Robustnost třídy** znamená nezávislost na zvoleném modelu počítačů. Je tedy jedno zda algoritmus budeme implementovat na Turingově či RAM stroji, složitost problému zůstane v PTIME třídě. Protože RAM a Turingove stroje jsou polynomiálně ekvivalentní.

Do této třídy spadá mnoho problémů:

- Třídění
- Vyhledávání a vkládání v lineární či stromové struktuře
- Aritmetické operace (násobení či dělení)
- Ekvivalence deterministických konečných automatů
- Problém nalezení nejkratší cesty v grafu
- Problém minimální kostry grafu
- Výběr aktivit (výstupem je množina obsahující největší možný počet vzájemně se nepřekrývajících aktivit)
- Problém nejdelší společné posloupnosti (xxxyyy, xxyyy -> nejdelší společná posloupnost je xx)

NPTIME

NPTIME je třídou všech ANO/NE problémů, které jsou rozhodovány nedeterministickými polynomiálními algoritmy. Do třídy NPTIME patří rozhodovací problémy, pro které existuje polynomiální algoritmus, který ověří správnost nalezeného řešení.

Přesněji řečeno, pokud je odpověď „Ano“, existuje svědek, který dosvědčuje, že odpověď je „Ano“, kterého je možné v polynomiálním čase otestovat, že tomu tak skutečně je. Pokud je odpověď „Ne“, žádný takový svědek neexistuje.

Ověření pravdivosti pro konkrétní ohodnocení lze provést rychle (tj. polynomiálně). Exponencionalita algoritmu spočívá v tom, že je exponencionálně mnoho případů, které algoritmus ověřuje. U problému z NPTIME třídy pro každý vstup existuje hodně potencionálních řešení, přičemž pro každé z nich se dá rychle zjistit, zda se jedná o skutečné řešení. V případě, že existuje alespoň jedno skutečné řešení, je odpověď na vstup problému ANO.

Nedeterministický algoritmus rozhoduje ano/ne problémy, ale ne vždy správně. 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.

Příklady NP problémů:

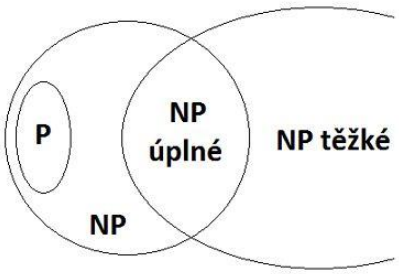
Název: IS (problém nezávislé množiny [Independent Set] (také nazývaný problém anti-kliky [anti-clique]))
Vstup: Neorientovaný graf G (o n vrcholech); číslo k (k ≤ n).
Otázka: Existuje v G nezávislá množina velikosti k (tj. množina k vrcholů, z nichž žádné dva nejsou spojeny hranou)?
Nedeterministický algoritmus pro problém IS (viz. obrázek a definice níže) 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.

Další NP problémy:

- Je přirozené číslo složené? (dělitelné beze zbytku)
- Jsou dva grafy izomorfní?
- Je možné obarvit graf K barvami tak, aby sousední vrcholy nebyly obarveny stejnou barvou?
- Je daná booleovská formule v konjunktivní normální formě splnitelná? (existuje pravdivostní ohodnocení, při kterém je formule pravdivá?)
- Existuje v grafu hamiltonovská kružnice? (uzavřená cesta procházející každým vrcholem právě jednou?)

NP-úplné problémy

Polynomiální převoditelnost – problém P1 je polynomiálně převeditelný na P2, jestliže existuje převádějící polynomiální algoritmus, který pro libovolný vstup problému P1 sestrojí vstup problému P2, přičemž platí, že odpověď na otázku P1 pro vstup *w* je ANO právě tehdy, když odpověď na P2 pro vstup *w* je ANO.



NP-těžký – Problém Q nazveme NP-těžkým, pokud každý problém ve třídě NP lze na problém Q polynomiálně převést.

NP-úplný – Problém Q nazveme NP-úplným, pokud je NP-těžký a zároveň náleží do třídy NP. To znamená, že třídu NP-úplných úloh tvoří v jistém smyslu ty nejtěžší úlohy z NP.

Pokud bychom nalezli polynomiální algoritmus rozhodující některý (kterýkoliv) NP-těžký problém, znamenalo by to, že existuje polynomiální algoritmus pro každý problém NP (P = NP).

Cook ukázal základní NP-těžký problém, který je také NP a je tedy NP-úplný: Problém splnitelnosti booleovských formulí (SAT).

Příklady NP-úplných problémů:

- SAT problém
- Barvení grafu k barvami
- Barvení grafu třemi barvami
- Problém hamiltonovské kružnice
- Subset-sum
- Problém obchodního cestujícího

Jazyk predikátové logiky prvního řádu

Výroková logika – umožňuje analyzovat věty pouze do úrovně elementárních výroků, jejichž strukturu již dále nezkoumá, př. „V Praze prší a v Brně je hezky“.

Predikátová logika 1. řádu – umožňuje navíc analyzovat elementární výroky do úrovně vlastností jednotlivých objektů zájmů (tzv. individuí – prvků univerza diskurzu) a jejich vztahů. Zatímco výroková logika se zabývá jednoduchými a deklarativními výroky, predikátová logika prvního řádu zavádí jako nadstavbu predikáty a kvantifikátory. Predikát se podobá funkci nabývající booleovské hodnoty (pravda - nepravda).

Zatímco výroková logika zkoumala pravdivost jednoduchých tvrzení, která byla maximálně spojena logickými spojkami, predikátová logika zkoumá vlastnosti a vztahy prvků daných množin. Kromě vlastností jednotlivých objektů tak vyřešíme potřebu formalizovat výroky typu "V této množině existuje prvek s danou vlastností" a "Všechny prvky této množiny mají danou vlastnost".

Predikátová logika vyšších řádů – umožňuje navíc analyzovat výroky do úrovně vlastnosti vlastností, vlastnosti funkcí atd.

Predikátová logika 1. řádu

Pouze jen malá část úsudků může být formalizována a dokázána v rámci výrokové logiky. PL1 formalizuje úsudky o vlastnostech předmětů a vztazích mezi předměty pevně dané předmětné oblastní (univerza). Na rozdíl od výrokové logiky si všímá i struktury vět samotných a obsahuje predikáty a kvantifikátory.

Jazyk PL1 se skládá z *abecedy* a *gramatiky*:

- **Abeceda** je tvořena následující skupinami symbolů:
  - **Logické symboly**
    - **Předmětové proměnné:** x, y, z
    - **Symboly pro spojky:** ¬, ∨, ∧, ⊃, ≡ (negace, disjunkce, konjunkce, implikace, ekvivalence)
    - **Symboly pro kvantifikátory:** ∀, ∃
  - **Speciální symboly**
    - **Predikátové symboly:** P, Q, R – reprezentují vlastnosti prvků univerza (unitární predikáty) nebo vztahy mezi dvojicemi prvků (binární predikáty – R(x,y)).
    - **Funkční symboly:** f, g, h

        - Ke každému funkčnímu symbolu a predikátovému symbolu je přiřazena **arita** – udává počet individuových proměnných, které jsou argumenty symbolu nebo predikátu
    - **Pomocné symboly** – závorky
- **Gramatika**, která udává, jak tvořit:
    - **Termy –** každá symbol proměnné TODO
    - **Atomické formule –** predikátový symbol s termy (P(t1…tn)) je atomická formule.
    - **Složené formule –** Každá atomická formule je formule; Je-li výraz A, B formule a x proměnná pak i výrazy: ¬A, ∀xA a ∃xA, (A ∨ B), (A ∧ B), (A ⊃ B), (A ≡ B) jsou formulemi

Volné a vázané proměnné

Volné proměnné jsou ty, které nejsou spjaty s kvantifikátorem, zatímco vázané jsou. **Formule s čistými proměnnými** – každá proměnná má buď všechny výskyty volné, nebo všechny výskyty vázané. **Uzavřená formule –** neobsahuje žádnou volnou proměnnou.

Převod z přirozeného jazyka do PL1

- Po všeobecném kvantifikátoru obvykle následuje formule ve tvaru implikace
- Po existenčním kvantifikátoru obvykle následuje formule ve tvaru konjunkce.
- Větu musíme často ekvivalentně přeformulovat, pozor: v češtině dvojí zápor !
- Výrazy „všichni“, „každý“, „nikdo“ překládáme **všeobecným kvantifikátorem** ∀
- Výrazy „někdo“, „někteří“ překládáme **existenčním kvantifikátorem** ∃

1) Nikdo, kdo není zapracován (P), nepracuje samostatně (S).
2) Ne každý talentovaný (T) spisovatel (Sp) je slavný (Sl).
3) Pouze zaměstnanci (Z) používají výtahu (V).
4) Všichni zaměstnanci (Z) používají výtahu (V).
5) Ne každý člověk (C), který hodně mluví (M), nemá co říci (R).
6) Někdo je spokojen (Sn) a někdo není spokojen.
7) Někteří chytří lidé (Ch) jsou líní (L).

1) $\forall x\,[\neg P(x) \supset \neg S(x)]$
2) $\neg\forall x\,\{[T(x) \wedge Sp(x)] \supset Sl(x)\}$
3) $\forall x\,[V(x) \supset Z(x)]$
4) $\forall x\,[Z(x) \supset V(x)]$
5) $\neg\forall x\,\{[C(x) \wedge M(x)] \supset \neg R(x)\}$
6) $\exists x\,Sn(x) \wedge \exists x\,\neg Sn(x)$
7) $\exists x\,[Ch(x) \wedge L(x)]$

Sémantika PL1 – interpretace formulí

Syntaxe – jak vypadají formule VS **Sémantika –** přiřazuje formulím a jednotlivým symbolům přesně definovaný význam.

Sémantika (význam formulí) predikátové logiky 1. řádu je dána jejich **intepretací -** Co daná formule znamená.

Interpretace – konkrétní soubor objektů, jejich vlastností a vztahů.

Nejprve musíme stanovit, o čem mluvíme – neprázdná množina **universum diskurzu** (soubor všech objektů v dané interpretaci), její prvky jsou **individua**. Predikátové symboly vyjadřují vztahy mezi těmito předměty (prvky univerza) – proto přiřadíme každému n-árnímu predikátovému symbolu jistou **n-ární relací nad univerzem** (podmnožina). **Funkční symboly** budou vyjadřovat n-ární **funkce** nad univerzem (zobrazení).

Teprve poté, co je daná formule interpretována, lze **vyhodnotit pravdivost v dané interpretaci**. Musíme ale nejdříve proměnným přiřadit valuaci individua tj. prvky univerza. Pravdivostní hodnoty formulí závisí na dané interpretaci a valuaci.

- **Tautologie** – pro každé pravdivostní ohodnocení je formule pravdivá

- **Kontradikce** – pro každé pravdivostní ohodnocení je formule nepravdivá
- **Splnitelná formule** – pokud má alespoň jedno ohodnocení, při kterém je pravdivá
- **Model formule** – Libovolná interpretace, ve které je daná formule pravdivá

Práce s kvantifikátory

Univerzální kvantifikátor ∀

- Pro každé x platí - $\forall x\, P(x)$
- „x je čtverec" – mluví se o jednom konkrétním prvku přiřazeném proměnné x – pravdivostní hodnota tvrzení závisí na konkrétním prvku (konkrétní valuaci)
- „každé x je čtverec" – mluví se o všech prvků univerza – pravdivostní hodnota tvrzení nezávisí na konkrétní valuaci.

Existenční kvantifikátor ∃

- Existuje x, pro které platí - $\exists x\, P(x)$

Ekvivalentní transformace formulí

Často je užitečné větu nejprve přeformulovat tak, aby měla stejné pravdivostní podmínky, tedy aby byla ekvivalentní s původní větou, ale její formalizace bude snazší.

Formule f1 a f2 jsou logicky ekvivalentní, jestliže pro každé pravdivostní ohodnocení v platí, že pro dané pravdivostní ohodnocení v mají formule f1 a f2 stejnou pravdivostní hodnotu. Pro zdůvodnění toho, že formule nejsou ekvivalentní, stačí najít jedno ohodnocení, které dává různé výsledky.

Nebo definice jinak pro LP1:

Formule f1 a f2 jsou **logicky ekvivalentní**, jestliže mají stejné pravdivostní hodnoty v každé intepretaci a valuaci.

Každá formule se dá převést na ekvivalentní formuli, která obsahuje z logických spojek pouze negaci, konjunkci a disjunkci.

Všechny ekvivalence, které platí ve výrokové logice, platí i v predikátové logice.

Některé důležité ekvivalence (výroková logika)

- Dvojitá negace
- Asociativita, komutativita, idempotence (pro konjunkci i disjunkci)

$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$ *asociativita*
$p \wedge q \Leftrightarrow q \wedge p$ *komutativita*
$p \wedge p \Leftrightarrow p$ *idempotence*

- Distributivní zákony

$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

- De Morganovy zákony

$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$
$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

- Ekvivalence týkající se implikace

$$p \to q \;\Leftrightarrow\; \neg p \vee q$$
$$\neg(p \to q) \;\Leftrightarrow\; p \wedge \neg q$$

- Ekvivalence týkající se spojky <->

$$(p \leftrightarrow q) \leftrightarrow r \;\Leftrightarrow\; p \leftrightarrow (q \leftrightarrow r) \qquad \textit{asociativita}$$
$$p \leftrightarrow q \;\Leftrightarrow\; q \leftrightarrow p \qquad \textit{komutativita}$$
$$p \leftrightarrow q \;\Leftrightarrow\; (p \to q) \wedge (q \to p)$$
$$p \leftrightarrow q \;\Leftrightarrow\; (p \vee \neg q) \wedge (\neg p \vee q)$$
$$p \leftrightarrow q \;\Leftrightarrow\; (p \wedge q) \vee (\neg p \wedge \neg q)$$

- Jak odstranit spojku <-> nebo ->

- Spojku "$\leftrightarrow$" je možno odstranit pomocí libovolné z následujících tří ekvivalencí:
  - $p \leftrightarrow q \;\Leftrightarrow\; (p \to q) \wedge (q \to p)$
  - $p \leftrightarrow q \;\Leftrightarrow\; (p \vee \neg q) \wedge (\neg p \vee q)$
  - $p \leftrightarrow q \;\Leftrightarrow\; (p \wedge q) \vee (\neg p \wedge \neg q)$
- Spojku "$\to$" je možno odstranit pomocí následující ekvivalence:
  - $p \to q \;\Leftrightarrow\; \neg p \vee q$

Příklad

$$(p \wedge q) \to r \;\Leftrightarrow\; p \to (q \to r)$$

$$\begin{aligned}(p \wedge q) \to r &\Leftrightarrow \neg(p \wedge q) \vee r \\ &\Leftrightarrow (\neg p \vee \neg q) \vee r \\ &\Leftrightarrow \neg p \vee (\neg q \vee r) \\ &\Leftrightarrow \neg p \vee (q \to r) \\ &\Leftrightarrow p \to (q \to r)\end{aligned}$$

Některé důležité ekvivalence (predikátová logika)

$$(\forall x \varphi) \wedge (\forall x \psi) \;\Leftrightarrow\; \forall x(\varphi \wedge \psi)$$
$$(\exists x \varphi) \vee (\exists x \psi) \;\Leftrightarrow\; \exists x(\varphi \vee \psi)$$

$$\neg\forall x \varphi \;\Leftrightarrow\; \exists x \neg\varphi$$
$$\neg\exists x \varphi \;\Leftrightarrow\; \forall x \neg\varphi$$

$$\forall x \forall y \varphi \;\Leftrightarrow\; \forall y \forall x \varphi$$
$$\exists x \exists y \varphi \;\Leftrightarrow\; \exists y \exists x \varphi$$

Pokud $x \notin \mathit{free}(\psi)$:

$$(\forall x \varphi) \wedge \psi \;\Leftrightarrow\; \forall x(\varphi \wedge \psi)$$
$$(\forall x \varphi) \vee \psi \;\Leftrightarrow\; \forall x(\varphi \vee \psi)$$
$$(\exists x \varphi) \wedge \psi \;\Leftrightarrow\; \exists x(\varphi \wedge \psi)$$
$$(\exists x \varphi) \vee \psi \;\Leftrightarrow\; \exists x(\varphi \vee \psi)$$

De Morganovy zákony:

3. $\models \neg\forall x A(x) \equiv \exists x \neg A(x)$
4. $\models \neg\exists x A(x) \equiv \forall x \neg A(x)$

Zákony distribuce kvantifikátorů:

5. $\models \forall x\,[A(x) \supset B(x)] \supset [\forall x A(x) \supset \forall x B(x)]$
6. $\models \forall x\,[A(x) \supset B(x)] \supset [\exists x A(x) \supset \exists x B(x)]$
7. $\models \forall x\,[A(x) \wedge B(x)] \equiv [\forall x A(x) \wedge \forall x B(x)]$
8. $\models \exists x\,[A(x) \wedge B(x)] \supset [\exists x A(x) \wedge \exists x B(x)]$
9. $\models [\forall x A(x) \vee \forall x B(x)] \supset \forall x\,[A(x) \vee B(x)]$
10. $\models \exists x\,[A(x) \vee B(x)] \equiv [\exists x A(x) \vee \exists x B(x)]$

Relace

Relace na množinách A_1, A_2, … , A_n je libovolná podmnožina kartézského součinu A_1 x A_2 x … x A_n.

Kartézský součin – součin množin, kdy každý prvek množiny je zkombinován s každým prvkem druhé množiny. Kartézský součin množin A a B (A x B) je **množina všech uspořádaných dvojic**, kde první prvek z dvojice patří do množina A a druhý prvek je z množiny B.

Množina – soubor prvků, kdy prvkem může být i další množina. Množina nemusí mít žádný prvek a prvky množiny jsou neuspořádané – nezáleží na pořadí.

Relaci můžeme klasifikovat podle

- Počtu druhů nosičů
  - Homogenní – pokud jsou množiny kartézského součinu shodné
  - Heterogenní
- Podle arity
  - Unární (relace na 1 množině)
  - Binární (relace na 2 množinách)
  - Ternární (3)
  - N-ární

Příklady homogenní relace

- **Č**íslo x leží v intervalu <y,z>
- Druhá mocnina x rovná se y
- X je prvočíslo
- Být ostře menší (<0,1>, <0,2>, <0.3>…)

Dvojice <a,b> patří do relace R se zapisuje jako: R(a,b)

Vysvětlení u zkoušky: Relace ne nějakých množinách je definována jako podmnožina kartézského součinu těchto množin. Je to takový vztah mezi dvěma prvky množiny. Je-li např. relace „být ostře menší“, pak mluvíme o binární homogenní relaci, protože je to relace na dvou množinách, proto je binární a ty množiny jsou stejné, jsou to množiny např. celých čísel, proto je to homogenní relace. Příklad dvojic prvků, které patří do relace je pak např. 0 je menší než 1 nebo 1 je menší než 2.

Operace s relacemi

Protože relace jsou ve skutečnosti množiny, můžeme s nimi provádět klasické **množinové operace**. Podmínkou je, že dané relace musí mít stejnou aritu – musí být všechny n-ární pro nějaké n.

- **Sjednocení** – prvek je v relaci $R_{sjednoceni}$, pokud je v relaci R_1 nebo v relaci R_2. Ve výsledné relaci bude v případě, že byl alespoň v jedné ze sjednocovaných relací
- **Průnik** - prvek je v relaci $R_{sjednoceni}$, pokud je v relaci R_1 a zároveň v relaci R_2. Ve výsledné relaci bude v případě, že je v obou relacích zároveň
- **Rozdíl** – Relace, do které patří prvky, které jsou v R_1, ale nejsou v R_2 (R_1-R_2)
- **Doplněk** – Všechny prvky, které nepatří do původní relace.

Další operace s binárními relacemi

- **Inverzní relace** – (neplést s doplňkem) – relace opačná k současné relaci. Vezmeme všechny prvky [a,b] z relace a do nové relaci vložíme jejich inverzi. ((a,b) -> (b,a))
- **Skládání relaci** – relace R∘S obsahuje prvek [a,c] právě tehdy, když R(a,b) a zároveň S(b,c).

Vlastnosti relací

- Relace je **reflexivní**, jestliže pro všechna a ∈ A platí (a, a) ∈ R. Každý prvek je v relaci sám se sebou
- Relace je **ireflexivní**, jestliže žádný prvek není v relaci sám se sebou
- Relace je **symetrická**, jestliže pro všechna a, b ∈ A platí, že pokud (a, b) ∈ R, pak (b, a) ∈ R. Je-li první v relaci s druhým, pak druhý musí být v relaci s prvním
- Relace je **anti-symetrická**, jestliže pro všechna a, b ∈ A platí, že pokud (a, b) ∈ R a (b, a) ∈ R, pak a = b. Je-li první v relaci s druhým a druhý je v relaci s prvním, pak první je identický s druhým.
- Relace je **asymetrická**, jestliže pro všechna a, b ∈ A platí, že pokud (a, b) ∈ R, pak (b, a) ∉ R. Je-li první v relaci s druhým, pak druhý není v relaci s prvním.
- Relace je **transitivní**, jestliže pro všechna a, b, c ∈ A platí, že pokud (a, b) ∈ R a (b, c) ∈ R, pak (a, c) ∈ R. Je-li první v relaci s druhým a druhý v relaci s třetím, pak je první v relaci s třetím.
- Relace je **cyklická –** Je-li první v relaci s druhým a druhý se třetím, pak je třetí v relaci s prvním.
- Relace je **lineární (souvislá)** je-li první v relaci s druhým nebo je druhý v relaci s prvním nebo jsou identické.

Je-li relace asymetrická, pak také musí být ireflexivní. Je-li relace asymetrická, pak také musí být antisymetrická a je-li relace souvislá, pak také musí být antisymetrická.

Typy binárních relací

- Relace typu **ekvivalence** – relace, která je současně **reflexivní, symetrická** a **tranzitivní**
- Relace typu **uspořádání** (částečného, neostrého) – relace, která je současně **reflexivní**, **antisymetrické** a **tranzitivní**.
- **Kvaziuspořádání** – relace, které je **reflexivní** a **tranzitivní**

Relace ekvivalence

Relace R na množině X je ekvivalence, jestliže je současně **reflexivní**, **symetrická** a **tranzitivní**. Je to zjemnění relace rovnosti. Hodí se určit, zda jsou si dva prvky podobné, ne nutně stejné – zdali patří do skupiny se stejnou základní vlastností. Např. dvě knihy jsou ekvivalentní, pokud mají stejný žánr.

Průnik ekvivalence je opět ekvivalence. Sjednocení ekvivalencí není obecně ekvivalence. Tranzitivní uzávěr sjednocení ekvivalencí však ekvivalencí je.

Příklady:

- **Bydlet ve stejném městě:** Sám bydlím ve stejném městě jako já sám (reflexivita). Pokud já i on bydlí ve stejném městě, tak i on i já bydlím ve stejném městě (symetrie). Pokud *a* bydlí ve městě s *b* a *b* bydlí ve městě s *c*, pak i *a* bydlí ve městě s *c* (tranzitivita)
- **Písně se stejným autorem**
- **Rovnost –** nejmenší relace splňující reflexivitu a tedy je to i nejmenší ekvivalence.

Třídy ekvivalence

Každá ekvivalence rozdělí množinu M na systém disjunktních množin, ty se nazývají **třídy ekvivalence.** Jednotlivé množiny tedy nemají žádný společný prvek a všechny prvky v dané množině jsou navzájem ekvivalentní. Každý prvek jednoznačně identifikuje svou třídu ekvivalence – zapisuje se jako M[x].

Relace uspořádání

Relace je **uspořádání** (částečné a neostré), jestliže je současně **reflexivní**, **antisymetrické** a **tranzitivní**. Relace je **úplné uspořádání** (neostré), jestliže je současně **reflexivní**, **antisymetrická tranzitivní** a **souvislá**. Relace je **ostré uspořádání** (částečné), jestliže je současně **asymetrická** a tedy také **ireflexivní** a **antisymetrická**, a **tranzitivní**. Relace je **úplné ostré uspořádání**, jestliže je současně **asymetrická** (a tedy i **ireflexivní** a **antisymetrická**), **tranzitivní** a **souvislá**.

Relace uspořádání (větší, menší (ostře větší, ostře měnší) umožňuje porovnávat jednotlivé prvky mezi sebou.

Úplně uspořádaná množina - nebo také **řetězec**, je, pokud jsou každé dva prvky z množiny porovnatelné (číselné množiny).

Typy prvků:

- **Minimální** prvek – pokud neexistuje menší prvek
- **Maximální** prvek – Nejsme schopni nalézt prvek, který by byl větší od prvku x (max prvků může být více)
- **Nejmenší** prvek – nejmenší prvek a je menší nebo roven než všechny ostatní prvky z množiny M
- **Největší** prvek – největší prvek b je větší nebo roven než všechny ostatní prvky z množiny M. Pokud má množina největší prvek, pak tento prvek je zároveň i maximální.
- **Dolní závora -**
- Horní závora -
- **Infimum množiny B** – největší prvek, který je však pořád menší než prvky množiny B
- **Supremum množiny B** – nejmenší prvek větší než prvky množiny B

Uspořádání množiny se dá nakreslit pomocí **Hasseových diagramů** – graf, ve kterém vrcholy představují prvky množiny a hrana mezi vrcholy (a,b) říká, ze a < b. Musí platit, že v grafu je vrchol a níže než vrchol b.

Svaz – jestliže spolu s každými dvěma prvky z množiny X existuje v X jejich infimum a supremum.

Úplný svaz – jestliže pro každou neprázdnou podmnožinu z množiny X existuje v X infimum a supremum.

TODO: Dát obrázek (ručně) diagramu a určit min, max, sup, inf…:

Pojem operace a obecný pojem algebra

Operace

Pojem operace označuje postup, který na základě daných vstupů (argumenty) vyprodukuje jednu nebo více výstupních hodnot.

Formální definice

Operace je zobrazení z kartézského součinu nějakých množin do kartézského součinu jiných množin: $A_1 \times A_2 \times A_3 \ldots \rightarrow A_i \times A_{i+1}$. V algebře se pracuje zejména s n-ární operací, která zobrazuje kartézský součin množin do množiny. Podle n se pak operace dělí na:

- **Nulární** – operace bez vstupu – konstanta
- **Unární** – na vstupu je jeden prvek, který převedeme na jiný (např. operace absolutní hodnoty)
- **Binární** – nejčastější operace, pracuje se dvěma vstupními prvky (např. sčítání, odčítání). Binární operace přiřazuje každé dvojici prvků prvek nějaké množiny.
- **N-ární**

Algebra

Algebra je odvětví matematiky zabývající se abstraktními pojmy struktury (nějakého objektu) a vztahů (mezi nějakými objekty). **Algebraická struktura** je systém množin spolu se systémem operací definovaných na těchto množinách.

Formální definice:

Univerzální algebra je dvojice (A, F^A), kde:

- A je nosič algebry. Je to nějaká množina objektů, například celých čísel
- F^A je množina operací nad nosičem A

Příklad algebry: $(N, +^2, *^2)$ – množina přirozených čísel s operacemi sčítání a násobení

Algebry s jednou a dvěma binárními operacemi

Algebry s jednou binární operací (grupoidy)

- Grupoidy, pologrupy, monoidy a grupy

Základní algebraická struktura s jednou operací je grupoid. Je to množina, na které je definována jedna binární operace tak, aby byla na této množině proveditelná – výsledkem operace provedené na libovolných prvcích množiny byl prvek z této množiny (množina je vůči této operaci uzavřená).
Příklady grupoidu – sčítání nad množinou reálných čísel, násobení nad množinou reálných čísel.

G = (G, ●) je grupoid s operací ●, která je ve tvaru G x G → G. Grupoid G se nazývá

- **Komutativní**, platí-li: a ● b = b ● a (nezáleží na pořadí prvků v operaci)
- **Asociativní**, platí-li: (a ● b) ● c = a ● (b ● c) (nezáleží na závorkách)
- **S jednotkovým (neutrálním prvkem),** platí-li: a ● e = a (takový prvek, který nezmění vstupní prvek; 0 u sčítání a 1 u násobení)
- **S nulovým prvkem**, platí-li: a ● o = o (prvek, který změní výsledek na sebe – na nulový prvek, 0 u násobení)

- **S inverzními prvky**, platí-li: a ● a^{-1} = e (pomocí inverzního prvku lze každý prvek změnit na jednotkový prvek)

Podle vlastností rozlišujeme několik typů grupoidů:

- **Pologrupa** – asociativní grupoid (musí platit asociativita)
- **Monoid** – pologrupa s jednotkovým prvkem (musí platit asociativita a musí mít jednotkový prvek)
- **Grupa** – monoid s inverzním prvkem
- **Abelova grupa** – komutativní grupa

Příklady:

- **Odečítání na množině celých čísel** – grupoid (není to pologrupa, protože odečítání není asociativní)
- **Sčítání na množině přirozených čísel (bez nuly)** – pologrupa, ale není to monoid, protože nemá jednotkový prvek
- **Sčítání na množině přirozených čísel (s nulou)** – monoid, ale není to grupa, protože nemá inverzní prvek (má jednotkový prvek – 0)
- **Sčítání na množině celých čísel** – abelova grupa, protože sčítání je asociativní a komutativní, jednotkový prvek je 0 a inverzní prvek je –číslo
- **Násobení na množině přirozených čísel** – pouze komutativní monoid, jednotkový prvek je 1, chybí ale inverzní prvek
- **Násobení na množině racionálních čísel** – abelova grupa: násobení je asociativní i komutativní, jednotkový prvek je 1, inverzní prvek je 1/m

Algebry se dvěma binárními operacemi (okruh, svaz?)

Definice okruhu:

Okruh je algebraický systém (A, +, *) se dvěma základními binárními operacemi sčítání a násobeni s následujícími vlastnostmi:

- (A,+) je Abelova grupa – **aditivní grupa okruhu**
- (A,*) je grupoid – **multiplikativní grupoid okruhu**

Těleso – Speciální případ okruhu, který navíc přináší existenci inverzního prvku. (komutativní grupa).

Pro množinu A platí následující axiomy:

- Uzavřenost pro + a *
- Komutativita +
- Asociativita +
- Asociativita *
- Nulový prvek +
- Opačný prvek +
- Jednotkový prvek *
- Distributivita *

Definice svazu:

Svazy mají dvě binární operace s duálními vlastnostmi (okruhy mají binární operace, které nejsou navzájem duální – axiomy jsou stejné jak pro spojení tak i pro průsek). Svaz je algebra (L,∪,∩) s binárními operacemi **spojení** a **průsek**, které splňují následující vlastnosti:

- Univerzaliza a jednoznačnost
- Asociativita
- Komutativita
- Absorpce

Příkladem svazu je **booleova algebra**.

Booleova algebra

Distributivní komplementární svaz. Šestice (A, $\wedge$, $\vee$, $-$, 0, 1), kde A je množina prvků (0,1), - je operace doplněk, a binární operace průsek a spojení. BA splňuje následující axiomy:

- Komutativita
- Distributivita
- Neutralita 0 a 1: $x \wedge 1 = x$
- Komplementarita: $x \wedge -x = 0$

Homomorfismus:

- Zobrazení z jedné algebraické struktury do jiné stejného typu, které zachovává veškerou důležitou strukturu. Každý typ algebraické struktury má svůj typ homomorfismu:
    - Monomorfismus – zobrazení je prosté
    - Epimorfismus- zobrazení je subjektivní (na)
    - Izomorfismus – zobrazení je bijektivní (prosté i na)

TODO: Podrobně prozkoumat svazy

Formální konceptuální analýza

Uspořádání dat do tabulky – *Objekty* mají *atributy* (vlastnosti) a můžeme formulovat různá tvrzení o tom, zda daný objekt má nějakou vlastnost nebo ji nemá. Vztah „mít“ mezi objekty a atributy je nejčastěji reprezentován tabulkou, kde řádky jsou objekty a sloupce atributy. Položka tabulky ukazuje, zda daný objekt má danou vlastnost, popř. informaci o tom, jakou hodnotu má objekt.

Formální konceptuální analýza nebo také **metoda konceptuálních svazů** je metoda (průzkumové) analýzy tabulkových dat. Vstupem pro FKA jsou tabulková data. Nabízí netriviální informace o vstupních datech, např. nové poznatky o vstupních datech, které nejsou při pouhém pohledu na vstupní data zřejmé, popř. výstup může být využitelný při dalším zpracování dat. FKA poskytuje 2 výstupy:

- **Konceptuální svaz** – hierarchicky uspořádá množina shluků (formálních konceptů), které jsou přítomny ve vstupní tabulce
- **Atributové implikace** – popisují závislosti mezi atributy

Formální kontext

Formální definice:

Formální kontext je trojice (X,Y,I), kde I je binární relace mezi množinami X a Y. Prvky množiny X jsou objekty a prvky množiny Y jsou atributy. Relace I určuje, zda má objekt x atribut y – I(x, y). Formální kontext tedy reprezentuje tabulku s objekty a atributy.

Formální koncept

Pojem – je tvořen svým rozsahem (extent) a obsahem (intent). Rozsah pojmu je seskupení všech objektů, které pod pojem patří. Obsah pojmu je seskupení všech atributů, které podpojem patří.

Formální koncept – dvojice (A,B), kde A je množina objektů a B je množina atributů, které pod pojem patří a musí platit, že všechny objekty v A sdílejí všechny atributy z B. V tabulkových datech koncepty odpovídají maximálním obdélníkům, které jsou vyplněné jedničkami.

Jednotlivé formální koncepty lze mezi sebou uspořádat ve smyslu obecnosti, kdy jeden koncept je podkonceptem jiného konceptu. Pokud je každý objekt z A1 patří do A2 a pokud každý atribut z B1 je i v B2, pak (A1,B1) je podkonceptem (A2,B2). Tento vztah umožňuje množinu všech konceptů uspořádat podle jejich obecnosti. Takto uspořádaná množina se nazývá **konceptuální svaz**.

Formální definice:

Formální koncept v kontextu (X, Y, I) je dvojice (A,B), kde A je podmnožinou X a B je podmnožinou Y takové, že B jsou právě všechny atributy společné objektům z A a A jsou právě všechny objekty sdílející atributy z B.

Množinu A nazýváme extent a množinu B intent konceptu.

Konceptuální svazy

Svaz obecně: Množina M, na které jsou definovány binární operace průsek a spojení takové, že tato struktura splňuje:

- Asociativitu (pro průsek a spojení)
- Komutativitu (pro průsek a spojení)
- Booleovy vlastnosti (pro průsek a spojení)

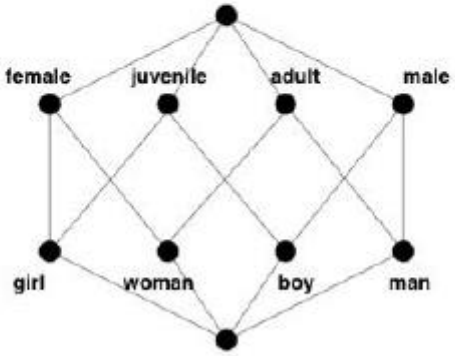
Úplný svaz: Uspořádána množina, v níž pro libovolnou podmnožinu existuje infimum a supremum.

Konceptuální svaz je množina všech formálních konceptů spolu s relací „podpojem-nadpojem“ definovanou na těchto formálních konceptech, kde koncept 1 je podpojmem konceptu 2, jestliže objekty z konceptu 1 jsou podmnožinou objektů z konceptu 2.

Konceptuální svaz může být vyjádřen tzv. **Hasseovým diagramem**.

Příklad:

	female	juvenile	adult	male
girl	x	x		
woman	x		x	
boy		x		x
man			x	x



TODO: Udělat na papír nějaký příklad podle zadání dráždilové

Asociační pravidla, hledání často se opakujících množin položek

Asociační pravidla – takové pravidlo, kde přítomnost jedné nebo více položek implikuje přítomnost jiných položek v téže transakci. Příklad: nákupní košíky – co si lidé nejčastěji kupují s určitým druhem zboží. Asociační pravidlo ve tvaru A => B říká, že pokud je A v relaci, tak je tam také B.

Hledání často se opakujících množin položek – jde o hledání asociací (vztahy) mezi jednotlivými položky pomocí apriori algoritmu.

Základními charakteristikami asociačních pravidel jsou **podpora (support)** a **spolehlivost (confidence)**

Asociační pravidlo: *předpoklad => závěr (Ant => Suc)*

Support – relativní počet objektů, které splňují předpoklad i závěr.

$$P(Ant \wedge Suc) = \frac{počet\ společných\ výskytů\ Ant\ a\ Suc}{počet\ řádků\ tabulky}$$

Confidence – nazývána též platnost nebo správnost je podmíněná pravděpodobnost závěru pokud platí předpoklad:

$$P(Suc|Ant) = \frac{počet\ společných\ výskytů\ Ant\ a\ Suc}{počet\ výskytů\ Suc}$$

Pokrytí (coverage) – podmíněná pravděpodobnost předpokladu pokud platí závěr:

$$P(Ant|Suc) = \frac{počet\ společných\ výskytů\ Ant\ a\ Suc}{počet\ výskytů\ Ant}$$

Na základě confidence a coverage lze dělit implikace do několika skupin:

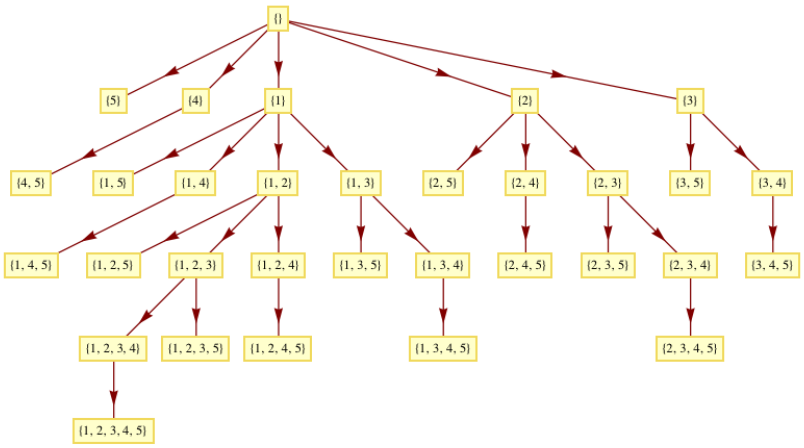
- Konzistentní pravidla – s confidence = 1, levá strana implikace je postačující podmínkou pro splnění pravé strany
- Úplná pravidla – coverage = 1, levá strana implikace je nutnou podmínkou pro splnění pravé strany
- Deterministická pravidla – pravidla s confidence i coverage = 1, levá strana je nutnou a postačující podmínkou pro splnění pravé strany.

Postup generování pravidel

1. Nejdříve vstupní tabulku převedeme do binární formy
2. Vygenerování kombinací, které splňují požadavek na minimální četnost (support)
3. Vytváření asociačních pravidel z kombinací z kroku 3, tak, že pokud máme kombinaci ABC, tak jsou z toho asociační pravidla AB=>C, AC=>B, BC=>A, C=>AB, B=>AC, A=>BC
4. Získání confidence asociačních pravidel

Základem všech algoritmů pro hledání asociačních pravidel je generování kombinací hodnot atributů. Generování kombinací je výpočetně náročný proces, proto se používá **apriori algoritmus:**

- Jedná se o hledání často se opakujících množin položek (frequent itemsets), což jsou kombinace kategorií, které dosahují předem zadané četnosti v datech.
- Předem zadaná četnost je nazývána **minsup**
- Při hledání kombinací délky *k*, které mají vysokou četnost se využívá toho, že již známe kombinace délky *k-1*. Při vytváření kombinací délky *k* spojujeme kombinace délky *k-1*.
- Pro vytvoření jedné kombinace délky *k* požadujeme, aby všechny její podkombinace délky *k-1* splňovaly požadavek na četnost.
- Toto lze vyjádřit tzv. **Rymon tree (ukázka na obrázku):**



TODO – Jednoduchý příklad na papír pro generování pravidel

Metrické prostory – metriky

Kromě Euklidovské vzdálenosti, která reprezentuje např. vzdušnou vzdálenost na mapě, můžeme zavádět i různé jiné vzdálenosti. Abstrakcí vzdálenosti je **metrika.** Musíme však mít množinu objektů, jejichž vzdálenost chceme měřit. **Metrický prostor** je pak tato množina spolu se vzdáleností pro libovolné dva prvky této množiny.

Formální definice:

Metrický prostor je dvojice (X, d), kde X je množina objektů a *d* je reálná funkce, která pro každou dvojici prvků (a, b) z množiny X přiřazuje reálné nezáporné číslo, tj. $d: X \times X \to R$. Tato funkce musí splňovat následující vlastnosti:

1. Funkce d je symetrická: $d(a,b) = d(b,a)$
2. Pro $a, b \in X$ je vzdálenost $d(a,b) = 0$ právě tehdy, když $a = b$, neboli vzdálenost dvou totožných bodů je nulová
3. Funkce d splňuje **trojúhelníkovou nerovnost:** $d(a,b) \leq d(a,c) + d(c,b)$, neboli součet vzdáleností od a do b přes jiný vrchol c musí být nutně větší nebo roven přímé vzdáleností od a do b.

Takovou funkcí d nazýváme **metrikou** nebo jen vzdáleností.

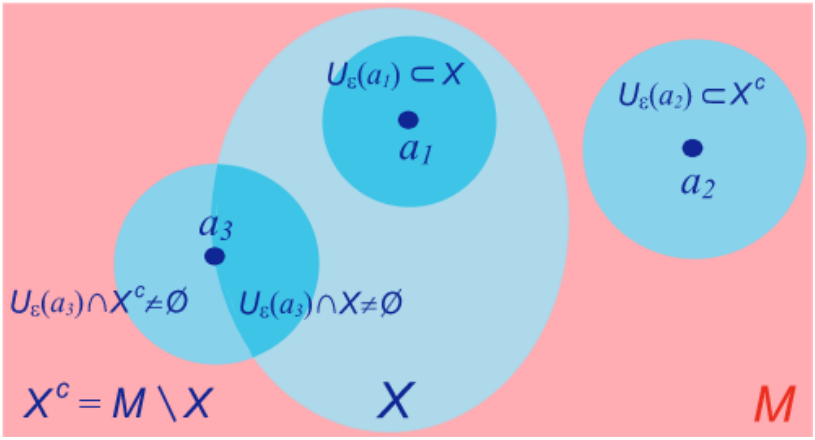
Množiny v metrických prostorech

Okolí bodu x_0: množina všech bodů z X, které mají vzdálenost od bodu x_0 menší než nějaká vzdálenost ε

Prstencové okolí bodu x_0: množina všech bodů z X, pro která platí, že $< 0 < d(x, x_0) < \varepsilon$, tedy okolí bodu bez samotného bodu x_0.

Nechť M je podmnožinou metrického prostoru X a x je z množiny X. Říkáme, že x je:

- **Vnitřním bodem** množiny M, pokud existuje okolí, které je podmnožinou M
- **Vnějším bodem** množiny M, pokud existuje okolí, které má s M prázdný průnik – leží mimo vlastní množinu M
- **Hraničním bodem** množiny M, pokud okolí bodu x má neprázdný průnik jak s množinou M, tak s jejím doplňkem – množina leží jak v M, tak i vně.



Pro množinu M pak definujeme

- **Vnitřek** jako množinu jejích vnitřních bodů
- **Hranici** jako množinu jejich hraničních bodu
- **Uzávěr** jako množinu vnitřních a hraničních bodu

Hromadný bod množiny M: Máme metrický prostor X. Množina M je podmnožinou množiny X. Bod z celkové množiny X nazýváme hromadný bod množiny M právě tehdy, když každé prstencové okolí bodu x obsahuje alespoň jeden bodu z množiny M. Nebo-li (asi) jsou to body na hraně množiny M, protože jedině na hraně mají pro každou vzdálenost v prstencovém okolí nějaký bod z M. Kdyby byl

bod v někde v X a množina M by byla dál, tak pro malou vzdálenost by neexistovalo okolí, které obsahuje bod z množiny M. Ale chuj ví k čemu to je…

Uzávěr množiny je M je samotná množina M + množina hromadných bodů množiny M.

Podmnožina M metrického prostoru X se nazývá **uzavřená** právě tehdy, když obsahuje všechny své hromadné body, tj. právě tehdy když M = uzávěr množiny M. **Otevřená** je, když každý bod z M je vnitřním bodem množiny M.

TODO – udělat si v těch pojmech pořádek + k čemu to je

Příklady metrických prostorů

- **Euklidovská metrika –** definována na množině R^n. Vyjadřuje délku úsečky mezi dvěma body. Tento metrický prostor se nazývá euklidovský prostor dimenze n. Euklidovská metrika je definována následujícím vztahem (pro 2D):

$$d_E(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

- **(Hammingova) Manhattanská vzdálenost –** metrika na množině R^n, při kterém je možný posun jen svisle nebo vodorovně. Definována následovně (pro 2D):

$$d_M = |x_1 - y_1| + |x_2 - y_2|$$

- **Maximová (Čebyševová) metrika –** bere se maximální hodnota z absolutních hodnot rozdílů jednotlivých složek bodu. Vzorec:

$$d_{max} = max\{|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|\}$$

- **Levenštejnova (editační) vzdálenost –** pro měření editační vzdálenosti v prostoru textových řetězců. Připouští tři jednoduché editační operace: přidání libovolného znaku, vypuštění libovolného znaku nebo záměnu libovolného znaku. Nejmenší počet těchto operací nutný k převedení jednoho řetězce na druhý udává vzdálenost mezi těmito řetězci.

Metriky pro určení podobnosti

- **Skalární součin** pro dva vektory (2D) - $a \cdot b = a_1 b_1 + a_2 b_2$
- **Kosinová podobnost –** oba vektory musí být normovány (mít jednotkovou délku); míra podobnosti dvou vektorů, která se získá výpočtem kosinu úhlu těchto vektorů. Použití: pro zjištění podobnosti dvou dokumentů. Skalární součin vektorů vydělený součinem jejich velikostí:

$$s_{cos} = \frac{A \cdot B}{||A||\,||B||}$$

- **Pearsonův korelační koeficient-** hodnoty z intervalu <-1,1>, měří sílu lineární závislosti mezi dvěma veličinami.

Topologické prostory

Metrický prostor je velmi obecná struktura umožňující pracovat jednotně s mnoha druhy množin. Přesto je možno mnohé pojmy z metrických prostorů definovat ještě obecněji v pojmu **topologický prostor**. Každý metrický prostor je zároveň topologickým prostorem, ovšem nikoli opačně.

Topologické prostory umožňují studovat vlastnosti ještě širší skupiny množin, než metrické prostory. Cílem topologie je studium vlastností prostorů. Nezajímáme se ale o vzdálenosti mezi body prostoru.

Formální definice:

Topologický prostor je dvojice (X, τ), kde X je množina a τ je kolekce podmnožin C, splňující následující axiomy:

1. $\emptyset \in \tau$, $X \in \tau$; neboli součástí kolekce podmnožin je i prázdná množina a celá množina X
2. Sjednocení libovolného počtu množin z τ leží v τ.
3. Průnik konečného počtu množin z τ leží v τ

Kolekci τ říkáme **topologie** na X. Množiny v τ nazveme **otevřené množiny**, jejich doplňky v X jsou uzavřené množiny.

Dva topologické prostory jsou **homeomorfní**, pokud mezi nimi existuje homeomorfimus, tj. zobrazení, které je

- Prosté
- Na
- Spojité
- Jeho inverze je spojitá

Topologie zkoumá tvar objektů bez přihlédnutí k vzdálenostem. Něco lze vytvarovat v zas něco jiného.

Absolutně nechápu, co těmi topologickými prostory chtěl básník říct ani k čemu to je.

Shlukování

Shluková analýza je postup formulovaný jako procedura, pomocí níž objektivně seskupujeme jedince do skupin na základě jejich podobnosti a odlišnosti. Shluková analýza patří mezi metody učení bez učitele.

Cílem shlukové analýzy je nalézt skupiny objektů tak, aby dva objekty z téže skupiny si byly podobnější, než dva objekty z různých skupin. Cílem je tedy najít shluky objektů.

Formalizace úlohy

Nechť X značí množinu n objektů. Rozklad $\Omega = \{C_1, C_2, \dots, C_m\}$ množiny X je množina disjunktních, neprázdných podmnožin X, které dohromady tvoří X. Průnik dvou podmnožin musí být nulový – prvek může patřit jen do jedné podmnožiny a sjednocení všech podmnožin dává celou množinu X. Každá množina C_i se nazývá **komponentou rozkladu**.

Shlukování je rozklad množiny X. Komponenty tohoto rozkladu se nazývají **shluky.**

Způsoby shlukování

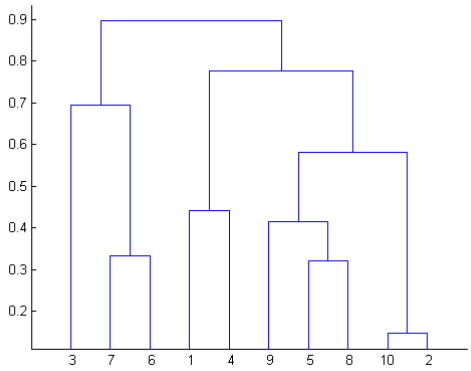
Podle způsobu shlukování se postupy dělí na **hierarchické** a **nehierarchické**. Hierarchické se dělí dále na **aglomerační** a **divizní**.

1. **Hierarchické shlukovací postupy**

Jsou založeny na hierarchickém uspořádání objektů a jejich shluků. Jedná se o posloupnost rozkladů, kde na jedné straně máme shluk obsahující všechny objekty a na straně druhé jednoprvkové shluky.

Podle směru postupu při shlukování dělíme metody hierarchického slukování na aglomerativní a divizní.

Graficky se hierarchicky uspořádané shluky zobrazují formou **dendrogramu.** Dendrogram je binární strom znázorňující hierarchické shlukování. Každý uzel stromu představuje shluk. Horizontální řezy jsou rozklady ze shlukovací sekvence. Vertikální směr představuje vzdálenost mezi shluky.

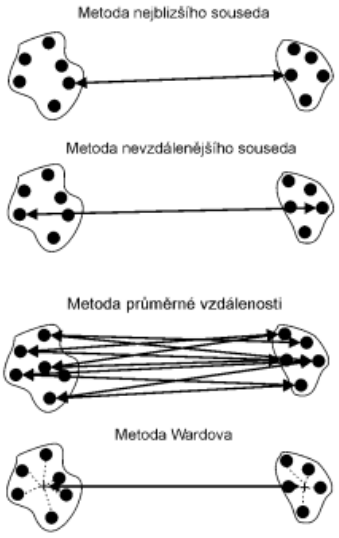


Aglomerativní hierarchické shlukování

U **aglomerativního hierarchického shlukování** máme na začátku n jednoprvkových objektů. Tyto jednoprvkové shluky dále slučujeme tak, že vybereme dle nějakého předem daného kritéria dva nejpodobnější jednoprvkové shluky a sloučíme je. Tím vznikne nový shluk. V každém dalším kroku vytváříme z nejpodobnějších shluků nižšího stupně rozkladu nový shluk. Shlukování může být ukončeno po dosažení požadovaného počtu shluků nebo až nám zůstane jen jeden shluk se všemi objekty.

Dělení aglomerativní metod je na základě toho, jakým způsobem se měří podobnost shluků:

- **Metoda nejbližšího souseda (simple linkage)** – Vzdálenost mezi shluky se počítá tak, že se vezme nejmenší ze vzdáleností každých dvou objektů z dvou různých shluků. Nevýhodou této metody je, že pokud existují objekty se stejnou vzdáleností od již existujících shluků, může dojít k zřetězení.
- **Metoda nejvzdálenějšího souseda (complete linkage)** – Tato metoda slučuje do jednoho shluku objekty nebo shluky, které jsou v rámci tříděné množiny dat nejdále od sebe. To znamená, že vzdálenost dvou shluků bere největší možnou vzdálenost ze vzdálenosti každých dvou objektů z dvou různých shluků. Tato metoda vytváří těsné shluky přibližně stejné velikosti. Zabraňuje vzniku zřetězených shluků.
- **Metoda průměrné vzdálenosti** – vzdálenost shluků od sebe je určena jako průměrná vzdálenost všech objektů v jednom shluku ke všem objektům ve druhém shluku.
- **Centroidní metoda** – Využívá se euklidovská metrika, ve které se změří vzdálenosti center shluků nebo objektů. Následně dojde ke sloučení shluků, které mají nejmenší vzdálenosti mezi centry.
- **Wardova metoda** – Slučuje takové shluky, kde je minimální součet čtverců. Má tendenci vytvářet poměrně malé shluky. V každém kroku se pro všechny dvojice odchylek spočítá přírůstek součtů čtverců odchylek, vzniklý jejich sloučením a pak se spojí ty shluky, kterým odpovídá minimální hodnota tohoto přírůstku.



Obr. 3 Nejčastěji užívané metriky shlukování.

Divizní hierarchické shlukování

Na začátku se bere množina, která se má zpracovat jako jeden shluk, ten se dále dělí na menší shluky a tím vytváří hierarchický systém. Každý shluk je rozdělen na dva nové, tak, aby byl rozklad optimální vůči nějakému kritériu. Na konci procesu jsou všechny shluky jednoprvkové. Exponenciální časová složitost – jen pro malý počet objektů.

2. Nehierarchické shlukování

Vytvářejí nehierarchickou strukturu, rozkládají množinu do podmnožin dle předem daného kritéria. První rozklad na množiny se již dále nedělí. Dělíme na

- **Optimalizační metody**, které hledají nejlepší rozklad přeřazováním objektů ze shluku do shluku, čímž minimalizují nebo maximalizují nějaké kritérium rozkladu.
- **Analýzy modů**, jež berou shluky jako místa s větší koncentrací objektů v n-rozměrném prostoru proměnných.

Před započetím samotné analýzy je nutné najít optimální počet shluků. Dle toho dělíme nehierarchické metody na **metody s pevně daným počtem shluků** a na **metody měnící počet shluků za běhu**. K určení optimálního počtu shluků, před samotnou analýzou se využívá různých indexů vypočtených na základě proměnlivého faktoru K – např. Calinski-Harabascův index, C index, Goodman-Kruskal.

K-means shlukování

Nehierarchická metoda shlukování s pevně daným počtem shluků. Shluky jsou definovány svými centroidy. Objekty se zařazují do toho shluku, jehož centroidu je nejblíže. Algoritmus postupuje iterativně tak, že se vyjde z nějakých počátečních (obvykle náhodně zvolených) centroidů, přiřadí do nich body, přepočítá centroidy tak, aby šlo o těžiště shluku bodů a pak opět přiřadí body k nově stanoveným centroidům a tak dál, dokud se poloha centroidů neustálý. Výhodou je jednoduchost a rychlost, dá se použít na velké množství dat. Nevýhodou je, že výsledky jsou ovlivňovány počátečních výběrem vrcholů. Není vhodné pro překrývající se shluky a pro přítomnost izolovaných objektů ležících mimo ostatní.

Fuzzy C-means

Další metoda s pevně daným počtem shluků. Ve fuzzy shlukování se pro každý bod počítá, s jakou pravděpodobností patří do jakého shluku. Takže body na okraji shluku mají menší stupeň příslušnosti než body v centru shluku. Objekt tak může patřit do více shluků zároveň. Dají se lépe identifikovat objekty, které se nedají přiřadit do žádného shluku. V případě, že každý objekt má pravděpodobnost příslušnosti k nějakému shluku rovnou jedné a k ostatním nulovou, pak je výsledkem pevné shlukování.

Metoda PAM (Partition around medoid)

Tato metoda se používá v případech, kdy je nutné z analyzovaných dat získat reprezentativní objekty přestavující jednotlivé shluky. To je užitečné při následné prezentaci výsledků shlukování. Reprezentativní objekty jsou nazvané medoidy.

Základem PAM metody je identifikace jednoho reprezentativního objektu, medoidu pro každý z k shluků. Počet shluků k bývá opět zvolen na začátku. Medoidy jsou vybrány tak, aby měl každý objekt k medoidu uvnitř svého shluku minimální vzdálenost. Objekty jsou přiřazeny do toho shluku, jehož medoidu jsou nejblíže.

Algoritmus má dvě základní fáze. V prvním je vybráno k reprezentativních bodů z matice dat. První bod má nejblíže ke všem bodů, je to tedy střed všech dat, další se vybírají tak, aby vzdálenost k ostatním objektům byla minimální. V druhé fázi se snažíme vylepšit rozklad, zkoumá se alternativní rozložení k bodů, zařazují se dosud nezařazené body.

V porovnání s metodou K-means je tato metoda robustnější, nevadí výstřední hodnoty a šum v datech. Ale pracuje efektivně pouze pro malé množiny dat

MacQueenova metoda se dvěma parametry

Nehierarchická metoda s proměnným počtem shluků. Oproti konstantním metodám není výsledek ovlivněn ojedinělými izolovanými objekty. Protože v případě, že izolovanému objektu připadne celý shluk, by mohl být nedostatek volných shluků pro ostatní objekty. Kromě hledání ideálního rozkladu, také hledají optimální počet shluků, na který tříděná data rozdělí. Tyto metody proto umožňují rozdělování a slučování skupin objektů během shlukování.

Na začátku tohoto algoritmu zadáváme počáteční počet k typických bodů, rozdělovací parametr R0 a slučovací parametr C. Prvních k bodů dosadíme za typické body. Typické body budou i v tomto případě těžiště shluků.

Dále algoritmus spočítá vzdálenost mezi typickými body. Pokud je nejmenší vzdálenost mezi nimi menší než C, tak se dané shluky sloučí a typický bod bude těžiště nově vzniklého shluku. Pokračujeme dalším měřením vzdáleností mezi typickými body a slučováním shluků, dokud nejsou všechny vzdálenosti mezi shluky větší nebo rovny C. K typickým bodům se přiřadí zbylé, nezařazené objekty. Vzdálenost mezi objektem a nejbližším typickým bodem by měla být nejvýše R0. Po přiřazení objektu, se přepočítá těžiště změněného shluku a dosadí se za typický bod. Opět se provede kontrola typických bodů a přeměření jejich vzdáleností. Pokud je vzdálenost přiřazovaného objektu k nejbližšímu typickému bodu vetší než R0, stává se sám typickým bodem. Nakonec se všechny objekty přesunou k nejbližšímu typickému bodu.

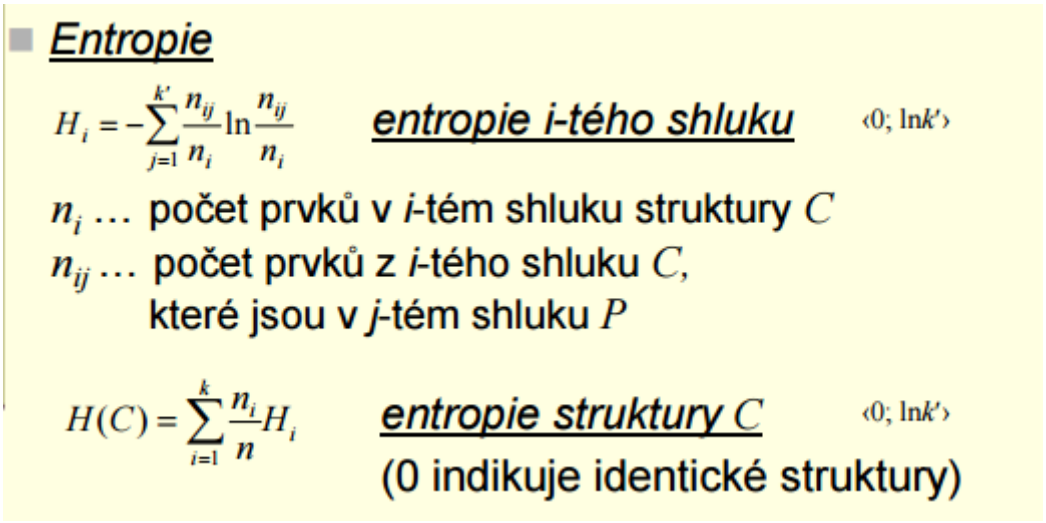
Tato metoda není náročná na čas, ale nemusí vždy dávat optimální výsledek. Někdy vznikne zbytečně mnoho shluků, které se musí dodatečně sloučit.

Měření podobnosti objektů

- Metriky (viz předešlá otázka)

Ověření kvality shlukování

- Pomocí entropie



TODO – nějaký jednoduchý příklad na hierarchické shlukování nastudovat + entropie?

Náhodná veličina

Základní pojmy

- **Teorie pravděpodobnosti** – matematická disciplína popisující zákonitosti týkající se náhodných jevů; používá se k modelování náhodnosti a neurčitosti.
- **Náhodný pokus** – děj, jehož výsledek není předem jednoznačně určen podmínkami, za nichž probíhá
- **Náhodný jev** – tvrzení o výsledku náhodného pokusu (Padne šestka) o kterém lze po ukončení pokusu rozhodnout
- **Elementární jev** – jednotlivý výsledek náhodného pokusu
- **Základní prostor** – množina všech elementárních jevů
- **Pravděpodobnost** – číselné vyjádření šance, že při náhodném pokusu daný jev nastane

Náhodná veličina

Představte si, že provádíte náhodný pokus, jehož výsledek jste schopni ohodnotit nějakým číslem. Před provedením pokusu jeho výsledek a tedy ani sledovanou hodnotu neznáte. Proto je proměnná, která připisuje výsledku náhodnému pokusu vámi sledovanou hodnotu označována jako **náhodná veličina**. Náhodnou veličinu značíme velkým písmenem, např. X. Množinu možných hodnot náhodné veličiny nazýváme obor hodnot náhodné veličiny X. Poté co je pokus proveden, je naměřená hodnota náhodné veličiny značena malým písmenem, např. x = 21mm.

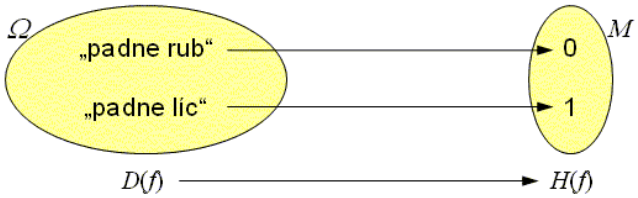
Příklady náhodné veličiny:

- Podíl vadných výrobků mezi tisíci
- Počet chybně přenesených bitů
- Proud v el. Obvodu

Výsledky některých pokusů (elementární jevy) jsou přímo vyjádřeny číselně (padne 1), u jiných tomu tak není (padne líc). Také u těchto pokusů je účelné přiřadit elementárním jevům čísla. Převádí se tedy abstraktní pojmy na čísla.

Formální definice:

Náhodná veličina X je reálná funkce definovaná na množině všech elementárních jevů, která každému jevu přiřadí reálné číslo. Čísla přiřazená elementárním jevům tvoří **obor hodnot M (H(f))**. Příklad s hodem mincí:



Další příklad:

- **Náhodná veličina X –** doba do remise onemocnění
- **Náhodný pokus –** Pozorování doby do remise onemocnění
- **Náhodný jev –** Doba do remise nepřekročí 60 měsíců ($X \leq 60$)

Náhodný vektor – vektor **X** = $(X_1, X_2, \ldots, X_n)^T$, jehož složky $X_1..X_n$ jsou náhodné veličiny na stejném základním prostoru. Např.: délka a šířka vyrobené součástky.

Číselné charakteristiky náhodných veličin

Distribuční funkce (viz dále) popisují rozdělení náhodné veličiny jednoznačně, do všech podrobností. Někdy nás zajímá pouze některý aspekt náhodné veličiny, který se dá popsat jedním číslem:

- **Obecný moment r-tého řádu $E(X^r)$**
- **Střední hodnota E(X)**
  - Lze chápat jako průměrnou (očekávanou) hodnotu náhodné veličiny, kolem niž hodnoty NV kolísají nebo jako míru polohy, populační průměr nebo jako vážený průměr možných hodnot.
- **P-kvantil x_p**
  - $P(X < x_p) = p$
  - Určujeme pouze pro spojitou náhodou veličinu
  - Kvartily
    - Dolní kvartil $x_{0,25}$
    - Medián $x_{0,5}$
    - Horní kvartil $x_{0,75}$
- **Modus $\hat{x}$** – hodnota náhodné veličiny, v niž funkce nabývá svého maxima
- **Rozptyl DX** – míra variability dat kolem střední hodnoty; malý rozptyl znamená, že se hodnoty NV s vysokovou pravděpodobností objevují blízko E(X)
- **Směrodatná odchylka σ**
- **Šikmost, špičatost**

Více viz otázka 13.

Základní typy náhodných veličin

Podle oboru hodnot M rozdělujeme náhodné veličiny na

- **Diskrétní** – obor hodnot M je konečná nebo nekonečná posloupnost; mohou nabývat spočetně mnoha hodnot
  - počet dní hospitalizace, počet zákazníků během 1 dne

- **Spojité** – obor hodnot M je otevřený nebo uzavřený interval; mohou nabývat všech hodnot na nějakém intervalu
    - Doba do remise onemocnění, výška, váha, IQ

Funkce určující rozdělení náhodných veličin

Pro úplný popis náhodné veličiny je nutné znát nejen množinu hodnot M, ale i pravděpodobnosti výskytu těchto hodnot. Popis náhodné veličiny provádíme nejčastěji pomocí **funkcí** a pomocí charakteristik (polohy, variability, koncentrace).

Funkce určující rozdělení náhodných veličin

- **Distribuční funkce F(x)**
- **Pravděpodobností funkce p(x)**
- **Funkce hustoty pravděpodobnosti f(x)**
- Diskrétní náhodnou veličinu lze charakterizovat pravděpodobností a distribuční funkcí
- Spojitou náhodnou veličinu lze charakterizovat hustotou pravděpodobnosti a distribuční funkcí

Pravděpodobností funkce DNV

Lze ji zadat

- Předpisem
- Tabulkou
- Grafem

Nechť X je diskrétní náhodná veličina s oborem možných hodnot $\{x_1, x_2, ..., x_n\}$, která tyto hodnoty nabývá s pravděpodobností $\{p_1, p_2, ..., p_n\}$. Údaje sestavíme do tabulky:

x_i	x_1	x_2	...	x_n
p_i	p_1	p_2	...	p_n

Každé hodnotě x_i je přiřazena právě jedna hodnota p_i. Lze to tedy chápat jako tabulkové určení pravděpodobnostní funkce.

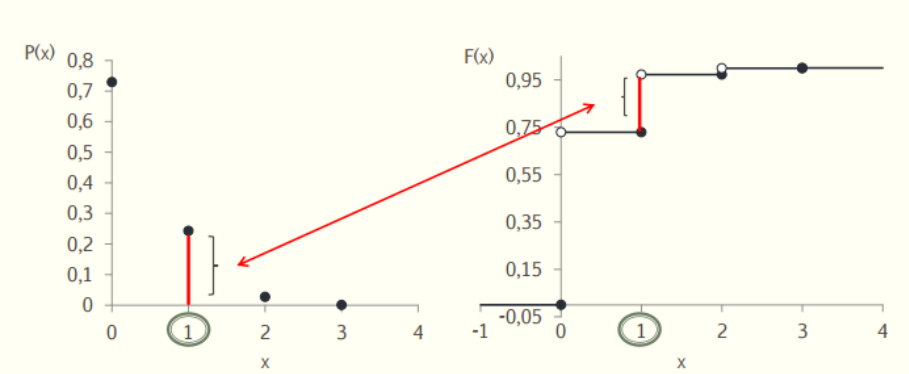
Vlastnosti pravděpodobnostní funkce

- Každá pravděpodobnost musí být větší nebo rovna 0
- Součet pravděpodobností se rovná jedné.

Distribuční funkce DNV

Distribuční funkce F(t) je pravděpodobnost, že náhodná veličina X bude menší než dané reálné číslo t:

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



Na obrázku je ukázána pravděpodobností funkce zadaná grafem a převod na distribuční funkci DNV. Body nespojitosti distribuční funkce jsou body, v nichž pravděpodobností funkce je nenulová. Velikost skoku distribuční funkce v bodech nespojitosti je rovna příslušným hodnotám pravděpodobností funkce.

Distribuční funkci lze zadat

- Předpisem
- Tabulkou
- Grafem (viz výše)

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

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

Vlastnosti distribuční fce pro DNV

- Hodnoty jsou v rozmezí 0 až 1
- Je neklesající
- Je zleva spojitá
- Má nejvýše spočetně mnoho bodů nespojitosti
- Začíná v 0 a končí v 1

Hustota pravděpodobnosti SNV

Pro spojitou náhodou veličinu se místo pravděpodobností funkce pro popis používá hustota pravděpodobnosti, protože náhodná veličina má spojité rozdělení pravděpodobnosti, tudíž má i spojitou distribuční funkci. Nemá tedy smysl jednotlivým realizacím náhodné veličiny přiřazovat hodnotu pravděpodobnosti, protože pravděpodobnostní funkce je nulová.

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

Vlastnosti hustoty pravděpodobnosti

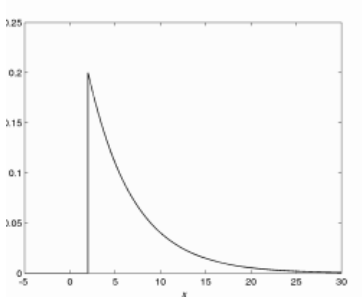
- Reálná nezáporná funkce
- Plocha pod křivkou hustoty je rovna jedné
- Začíná v nule a končí také v nule.

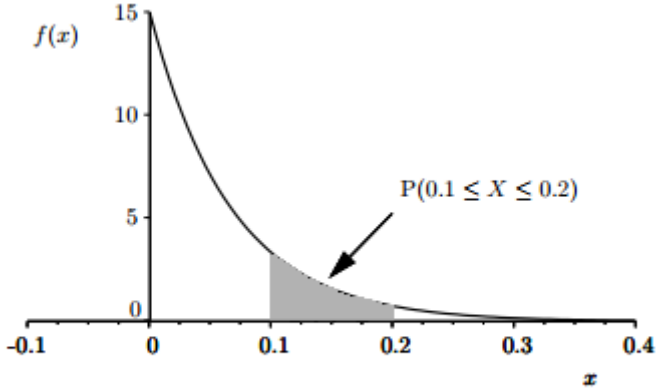
- Funkce může nabývat hodnot vyšších než 1, pravděpodobnost určuje plocha od mínus nekonečna do hodnoty t.

Funkce lze vyjádřit

$$f(x) = \begin{cases} \frac{1}{5}e^{-\frac{x-2}{5}} & \text{pro } x > 2, \\ 0 & \text{pro } x \leq 2. \end{cases}$$

- Vzorcem
- grafem





(a) Hustota náhodné veličiny X

Na obrázku je příklad určení pravděpodobnosti spojité náhodné veličiny, která má hodnotu v rozsahu 0,1 až 0,2. Pravděpodobnost je určená jako plocha pod křivkou v daném intervalu.

Distribuční funkce SNV

- definována stejně jako u diskrétní náhodné veličiny
- **Převod mezi distribuční funkcí a funkci hustoty pravděpodobnosti:** zderivuje se distribuční funkce:

Řešené úlohy

Příklad 3.3.1. Náhodná veličina X je dána distribuční funkcí:

$$F(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x^2}{4} & 0 < x \leq 2 \\ 1 & x > 2 \end{cases}$$

Určete $f(x)$, znázorněte graficky $F(x)$, $f(x)$, vypočtěte $P(0.4 \leq X < 1.6)$

Řešení: Hustotu pravděpodobnosti získáme zderivováním distribuční funkce:

$$f(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x}{2} & 0 < x \leq 2 \\ 0 & x > 2 \end{cases}$$

Vybraná rozdělení diskrétní a spojité náhodné veličiny

Rozdělení pravděpodobnosti náhodné veličiny – předpis, kterým charakterizujeme pravděpodobnost jevů, jež lze touto náhodnou veličinou popsat. Každému jevu popisovanému náhodnou veličinou

přiřazujeme určitou pravděpodobnost. U diskrétní náhodné veličiny je tímto předpisem (rozdělením) většinou pravděpodobnostní funkce, rozdělení spojité náhodné veličiny, je dáno distribuční funkcí, popř. hustotou pravděpodobnosti.

Vybraná rozdělení diskrétní náhodné veličiny

Binomické rozdělení

Bernoulliho pokusy – Posloupnost nezávislých pokusů (takových, kdy úspěch v libovolné skupině pokusů neovlivňuje pravděpodobnost úspěchu v pokusu, který do této skupiny nepatří). Pravděpodobnost úspěchu *p* v jednotlivých Bernoulliho pokusech je konstantní.

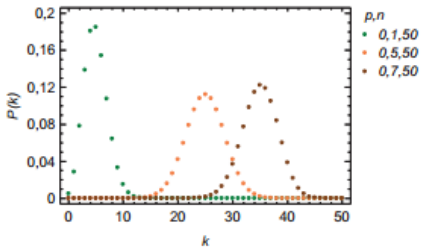
Binomické rozdělení – popisuje četnost výskytů náhodného jevu v *n* nezávislých pokusech (Bernoulliho pokusech). Neboli udává počet úspěchů v *n* Bernoulliho pokusech. Pokud *n=1*, jde o alternativní rozdělení.

Zápis: $X \to Bi(n,p)$, kde *n* je celkový počet Bernoulliho pokusů a *p* je pravděpodobnost úspěchu v každém z pokusů.

Pravděpodobností funkce: $P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$

Příklady:

- Počet chlapců mezi 10000 novorozenci
- Počet vadných výrobků mezi 30 testovanými
- Počet nevzrostlých rostlin ze 100 zasazených cibulek
- Na stůl vysypeme 15 mincí. Jaká je pravděpodobnost, že počet mincí ležících lícem nahoře je od 8 do 15? -> X = počet mincí ležících lícem nahoru z 15 mincí. $P(8 \leq x \leq 15) = P(x=8) + P(x=9) + \cdots + P(x=15)$



Obr. 5.1: Vliv parametru p na tvar pravděpodobnostní funkce binomické náhodné veličiny

Alternativní rozdělení

Rozdělení pravděpodobnosti náhodné proměnné, která s pravděpodobností p nabývá hodnoty 1 a s pravděpodobností 1 – p nabývá hodnoty 0. Jde o speciální případ binomického rozdělení. Náhodná veličina tedy udává **počet výskytů daného náhodného jevu (úspěchů) v jednom pokusu**.

Zápis: $X \to A(p)$

Pravděpodobností funkce: $P(X=1) = p$ a $P(X=0) = 1-p$ (stanovuje, jaká je pravděpodobnost, že dojde k úspěchu či neúspěchu.

Hypergeometrické rozdělení

Používá se podobně jako binomická náhodná veličina v situacích, kdy potřebujeme popsat počet úspěchů v *n* pokusech. Rozdíl je, že hypergeometrická náhodná veličina **popisuje počet úspěchů v *n***

závislých pokusech (Pravděpodobnost úspěchu v určitém pokusu závisí na výsledcích v předcházejících pokusech)**.**

Zápis: $X \rightarrow H(N, M, n)$, kde X … počet prvků se sledovanou vlastností ve výběru *n* z *N* prvků.

V základním souborů N prvků je M prvků s danou vlastností a zbylých N-M prvků tuto vlastnost nemá. Náhodně vybereme ze základního souboru *n* prvků, z nichž žádný nevracíme zpět. Hypergeometrické rozdělení je základním pravděpodobnostním rozdělením při výběru bez vracení.

Pravděpodobnostní funkce: $P(X = k) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}$

Příklady

- Počet vadných výrobků mezi 10 vybranými z dodávky 30 výrobků, mezi nimiž bylo 7 vadných
- Počet dívek v náhodně vybrané skupině 4 dětí ze třídy, v níž je 6 chlapců a 8 dívek
- Počet cibulí červených tulipánů v balíčku 10 cibulí vybraných ze směsi, která obsahuje 20 cibulí žlutých a 20 cibulí červených tulipánů
- Ve skladu je 200 součástek. 10% z nich je vadných. Jaká je pravděpodobnost, že vybereme-li ze skladu 30 součástek, tak nejméně 2 budou vadné? -> N = 200, M = 20, n = 30; 1 – P(x<2)
- Chci vybrat 4 piva (2 desítky, 2 dvanáctky) z basy, ve které je 10 desítek a 6 dvanáctek. -> X .. počet desítek ve výběru 4 piv z 16 -> N = 16, M = 10, n = 4; P(X = 2).

Je-li **výběrový poměr** (n/N) menší nez 0,05, lze hypergeometrické rozdělení nahradit binomickým s parametry *n* a *M/N*.

Negativně binomické rozdělení

Geometrické rozdělení

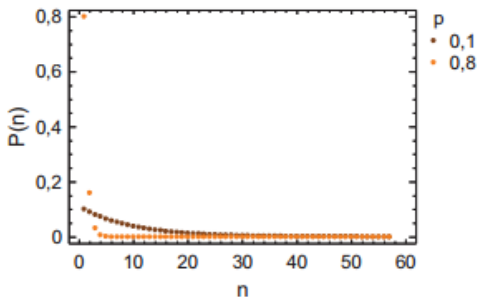
Geometrická náhodná veličina je definována jako: **počet Bernoulliho pokusů do prvního výskytu události (úspěchu)**, včetně něj.

Zápis: $X \rightarrow G(p)$, kde p je pravděpodobnost výskytu úspěchu v každém z pokusů

Pravděpodobnostní funkce: $P(X = n) = (1 - p)^{n-1} \cdot p$

Příklady

- Počet volání nutných k tomu, abychom se dovolali do televizní soutěže
- Počet řidičů, kteří podstoupí test na obsah alkoholu v krvi do doby, než bude nalezen první podnapilý řidič.
- Počet hodů poctivou kostkou nutných k padnutí 6



Obr. 5.2: Vliv parametru p na tvar pravděpodobnostní funkci geometrické náhodné veličiny

Negativně binomické (Pascalovo) rozdělení

Pascalova náhodná veličina je definována jako **počet Bernoulliho pokusů do k-tého výskytu události (úspěchu), včetně k-tého.** Jedná se o obecnější případ geometrické náhodné veličiny, kde geometrická náhodná veličina je speciálním případem negativně binomické náhodné veličin pro k = 1.

Zápis: $X \rightarrow NB(k, p)$, kde *p* je pravděpodobnost výskytu události a *k* udává požadovaný celkový počet výskytu události (úspěchů).

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

Příklady

- Počet dárců neznajících svou krevní skupinu, které musíte testovat proto, abyste našli 4 dárce s krevní skupinou 0
- počet cestujících, které musí revizor zkontrolovat do chvíle, než najde 10 černých pasažérů
- Pravděpodobnost, že se dovoláme do studia rozhlasové stanice je 0,08. Jaká je pravděpodobnost, že se dovoláme nejvýše na 4 pokus? -> X.. počet pokusů do prvního úspěchu -> NB(1; 0,08); $P(x \leq 4)$

Poissonovo rozdělení

Poissonův proces – Zobecnění Berhoulliho posloupnosti pokusů. Popisuje počet náhodných událostí v nějakém pevném časovém intervalu. Obecným názvem pro takové procesy je bodový proces (chod procesu, v němž čas od času dochází k nějaké události). Poissonův proces je speciálním případem bodového procesu. U Poissonova procesu musí být dodrženy následující tři předpoklady

- **Ordinarita** – pravděpodobnost výskytu více než jedné události v limitně krátkém časovém intervalu (t se blíží k nule) je nulová. Hovoříme o tzv. **řídkých jevech**.
- **Stacionarita** – pravděpodobnost výskytu jevu závisí pouze na délce intervalu, nikoli na jeho umístění na časové ose, neboli rychlost výskytu událostí je konstantní v průběhu celého intervalu
- **Nezávislé přírůstky** – počty událostí ve vzájemně disjunktních časových intervalech jsou nezávislé z toho vyplývá **beznáslednost**
    - Pravděpodobnost výskytu události není závislá na čase, který uplynul od minulé události

Poissonův proces lze obdobně jako v časovém intervalu definovat na libovolné uzavřené prostorové oblasti (plocha, objem).

Poissonovo rozdělení má náhodná veličina, která vyjadřuje počet výskytů jevů v určitém intervalu (času, délky, objemu), když nastávají nezávisle na sobě. Parametrem rozdělení je **rychlost výskytu událostí** (hustota výskytu události na ploše) – λ. Rychlost výskytu události je úměrná pravděpodobnosti výskytu jedné události za jednotku času.

Definujeme si náhodný pokus jako Poissonův proces (nezávislé události probíhající v čase *t* s hustotou výskytu λ). Pokud si náhodou veličinu X za těchto předpokladů definujeme jako **počet výskytů událostí v časovém intervalu délky t** nebo **počet výskytů událostí na ploše t (v objemu t)**, pak můžeme C považovat za náhodnou veličinu s Poissonovým rozdělením, což značíme:

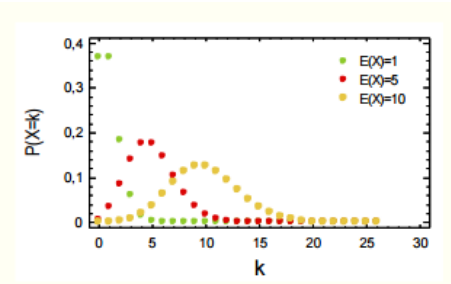
Značení: $X \rightarrow Po(\lambda t)$

Pravděpodobnostní funkce: $P(X = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}$

Příklady

- Počet pacientů ošetřených během dopoledních ordinačních hodin
- Počet mikro defektů na zadaném vzorku materiálu
- Počet mikroorganismů v 1dl vody
- Pokusy se zjistilo, že radioaktivní látka vyzařuje během 7,5s průměrně 3,87 alfa částice. Určete pravděpodobnost toho, že za 1 sekundu vyzáří tato látka alespoň 1 alfa částici.
    - X.. počet částic vyzářených za 1s
    - $\lambda t = \frac{3,87}{7,5} \cdot 1s$
    - $P(X \geq 1) = 1 - P(X > 1)$

Poissonovo rozdělení je charakteristické tím, že jeho střední hodnota a rozptyl se rovnají a to $E(X) = D(X) = \lambda t$



Poisonovým rozdělením lze velmi dobře aproximovat binomické rozdělení pro případ, že počet pokusů je velký (n > 30) a pravděpodobnost výskytu události je malá (p < 0,1) v takovém případě je $\lambda t = np$ -> $Bi(n,p) \sim Po(np)$.

Popis	Podmínky		Název náhodné veličiny - symbolický zápis
počet úspěchů v n pokusech	nezávislé pokusy	$k = 1$	**Alternativní** - $A(\pi)$
			Binomická - $Bi(n,\pi)$
	závislé pokusy		**Hypergeometrická** - $H(N,M,n)$
počet pokusů do k. úspěchu (včetně)	nezávislé pokusy	$k = 1$	**Geometrická** - $Ge(\pi)$
			Negativně binomická (Pascalova) - $NB(k,\pi)$
počet událostí v uzavřené oblasti (v čase, na ploše, v objemu)	ordinarita, stacionarita, beznáslednost procesu		**Poissonova** - $Po(\lambda t)$

Vybraná rozdělení spojité náhodné veličiny

Rovnoměrné rozdělení

Rozdělení, jehož hustota pravděpodobnosti je konstantní na nějakém intervalu a všude jinde je nulová.

Příklady

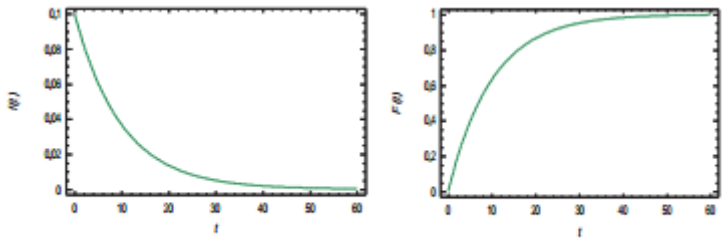
- Chyba při odečítání údajů z lineárních měřících přístrojů
- doba čekání na uskutečnění jevu opakujícího se v pravidelných intervalech.
- Na prohlídce výstavy je promítán film. Projekce začíná každých 20 minut. Určete pravděpodobnost, že pokud náhodně přijdete do promítaného sálu, nebude na začátek filmu čekat více než 5 minut.
    - X.. doba čekání na začátek filmu

- V každém časovém okamžiku je pravděpodobnost 1/20
- P(X < 5) = 5/20

Exponencionální rozdělení

Exponencionální rozdělení vyjadřuje rozdělení délky intervalu mezi náhodně se vyskytujícími událostmi, jejichž pravděpodobnost výskytu má Poissonovo rozdělení. Toto rozdělení úzce souvisí s rozdělením Poissonovým. Jestliže Poissonovo rozdělení popisuje počet výskytů událostí v časovém intervalu, exponenciální rozdělení se používá k **popisu doby do výskytu příslušné události**. Např. počet dopravních nehod na Martinovské křižovatce za určitý časový interval se popisuje Poissonovým rozdělením, zatímco dobu od jedné nehody do druhé lze popsat rozdělením exponenciálním.

Zápis: $X \rightarrow Exp(\lambda)$



Obr. 6.2: Hustota a distribuční funkce exponenciální náhodné veličiny

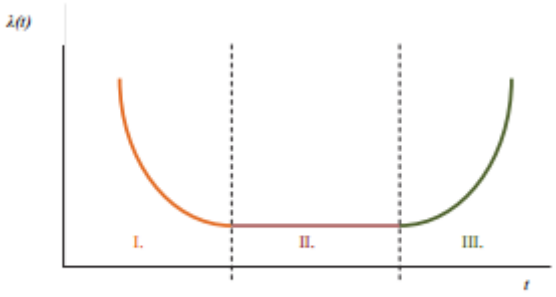
Příklady

- Doba do remise onemocnění
- Doba do poruchy zařízení
- Doba mezi 3. a 4. poruchou zařízení
- Do sítě se průměrně přihlašuje25 uživatelů za hodinu. Určete pravděpodobnost, že do dalšího pokusu přihlášení uběhnou 2-3 minuty
    - X.. doba mezi přihlášeními
    - $\lambda = \frac{25}{60}$
    - $P(2 < X < 3) = P(X < 3) - P(X > 2)$

Toto rozdělení hraje důležitou roli v teorii spolehlivosti. Časté aplikace jsou též v teorii hromadné obsluhy, kde se pomocí exponencionálnáho rozdělení modeluje doba čekání ve frontě.

Intenzita poruch

Modelujeme-li dobu do výskytu události (životnost, dobu do poruchy, dobu do relapsu apod), používáme kromě hustoty a distribuční funkce také funkci známou pod názvem **intenzita poruch** (hazardní funkce).



Obr. 6.4: Model intenzity poruch

Křivka se dělí na 3 úseky

1. Křivka intenzity poruch klesá. Interval se nazývá **období časných poruch**. Příčinou zvýšené intenzity poruch v tomto období jsou poruchy v důsledku výrobních vad, nesprávné montáže atd.
2. Ve druhém úseku dochází k běžnému využívání zaběhnutého výrobku, k poruchám dochází většinou z vnějších příčin, nedochází k opotřebení, které by změnilo funkční vlastnosti výrobku. Intenzita poruch je konstantní. Interval se nazývá **období stabilního života**.
3. Proces stárnutí a opotřebení mění funkční vlastnosti výrobku, intenzita poruch vzrůstá, interval se nazývá **období poruch v důsledku stárnutí a opotřebení.**

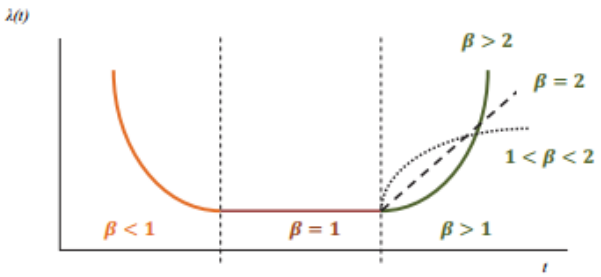
Intenzitu poruch modelujeme v jednotlivých úsecích pomocí různých rozdělení. Exponencionální rozdělení popisuje dobře rozdělení doby života systémů, u kterých dochází k poruše ze zcela náhodných příčin a nikoliv v důsledku opotřebení – v období stabilního života.

Weibullovo rozdělení

Weibullovo rozdělení stejně jako rozdělení exponencionální, slouží k modelování doby do výskytu události. Zatímco exponencionálním rozdělením lze modelovat pouze dobu do výskytu události u systémů, které se nacházejí v období stabilního života, Weibullovo rozdělení je mnohem flexibilnější a umožňuje tak modelovat dobu do výskytu události i u systémů, které jsou v **období časných poruch nebo v období stárnutí**.

Tab. 6.1: Vliv parametru β na tvar intenzity poruch

β=1	období stabilního života		$\lambda(t) = \lambda = \frac{1}{\theta} = konst.$ (exp. rozdělení)
$\beta > 1$	období stárnutí	$1 < \beta < 2$	λ(t) ... konkávní, rostoucí funkce
		$\beta = 2$	λ(t) ... lineárně rostoucí funkce
		$\beta > 2$	λ(t) ... konvexní, rostoucí funkce



Obr. 6.5: Vliv parametru β na tvar intenzity poruch

Značení: $X \to W(\Theta, \beta)$, kde Θ je parametr měřítka ($\Theta > 0;\ \Theta = \frac{1}{\lambda}$, závisí na materiálu, namáhání a podmínkách používání) a β je parametr tvaru ($\beta > 0$, ovlivňuje tvar intenzity poruch a tím i vhodnost použití pro určité období života).

Normální rozdělení

Nejpoužívanější pravděpodobnostní rozdělení modelující chování velkého množství náhodných jevů v technice, přírodních vědách i ekonomii je **rozdělení normální (gaussovo).** Za určitých podmínek lze pomocí něj aproximovat řadu jiných spojitých i nespojitých rozdělení.

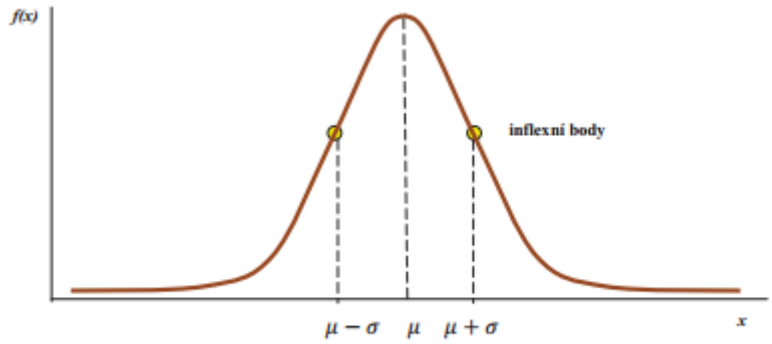
V souvislosti s normálním rozdělením jsou často zmiňovány náhodné chyby, např. chyby měření, způsobené velkým počtem neznámých a vzájemně nezávislých příčin. Proto bývá normální rozdělení také označováno jako **zákon chyb**. Podle tohoto zákona se také řídí rozdělení některých fyzikálních a technických veličin.

Příklady

- Výška v populaci chlapců. Určit jaké procento má výšku menší nebo rovno 93cm např.
- IQ populace
-

Zápis: $X \to N(\mu, \sigma^2)$

Normální rozdělení má dva parametry: **střední hodnotu** μ, charakterizující polohu tohoto rozdělení a σ^2 je **rozptyl**, charakterizující rozptýlení hodnot náhodné veličiny kolem střední hodnoty.



Obr. 6.9: Hustota pravděpodobnosti normálního rozdělení

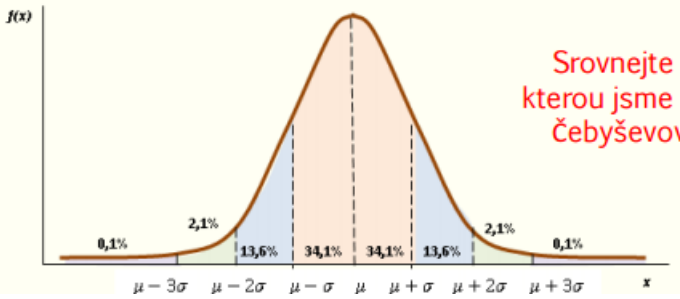
Grafem hustoty pravděpodobnosti náhodné veličiny s normálním rozdělením je Gaussova křivka (zvonovitá křivka). Maximum křivky je v její střední hodnotě. Parametr σ udává vzdálenost inflexních bodů (34,1%) od střední hodnoty a tím určuje i šířku Gaussovy křivky.

Normované normální rozdělení N(0;1) – normální rozdělení, které má střední hodnotu rovnu nule a rozptyl se rovná jedné. Pomocí distribuční funkce normované normální rozdělení lze spočítat distribuční funkce náhodných veličin s normálním rozdělením o libovolných parametrech.

Pravidlo 3σ

Máme-li data pocházející z normálního rozdělení, pak téměř všechna (99,8%) leží v intervalu $\mu \pm 3\sigma$. Pro náhodnou veličinu s normovaným rozdělením lze vyčíslit pravděpodobnost, že náhodná veličina se bude vyskytovat v intervalu $(\mu - k\sigma;\ \mu + k\sigma)$:

k	$P(\mu - k\sigma < X < \mu + k\sigma)$
1	0,682
2	0,954
3	0,998



Srovnejte s představou, kterou jsme měli na základě Čebyševovy nerovnosti!

Nástroje pro ověření normality

- **Graficky**
  - Q-Q graf – musí mít tvar přímky
  - Porovnání grafu hustoty s odhadem hustoty vypočítaného na základě dat

Popisná (Explorační) statistika

Explorační (popisná) statistika – Grafická prezentace a uspořádání dat do názornější formy a jejich popis několika málo hodnotami, které by obsahovaly co největší množství informací obsažených v původním souboru. Analýza výběru.

Statistika – zjišťování údajů o populaci na základě výběrového souboru. Populace (základní soubor) je množina všech prvků, které sledujeme při statistickém výzkumu. Populace může být: všechny dívky, které mají 15 let, všechny vyrobené součástky atd. Vzhledem k tomu, že rozsah populace je obvykle vysoký, získáváme informace o populaci prostřednictvím statistického výzkumu. Zkoumaná část populace se nazývá **výběr (výběrový soubor)**, ten je vytvořen několika vybranými prvky z populace (nejčastěji náhodným výběrem). Proč to dělat? Za prvé je to úspora času a financí. Dalším důvodem je minimalizace ztrát v důsledku destruktivního chování, kdy nějaké testování prvku má za následek její zničení (k čemu by pak bylo testování celé populace). A hlavním důvodem je nedostupnost celé populace – např. otestování všech lidí na zemi.

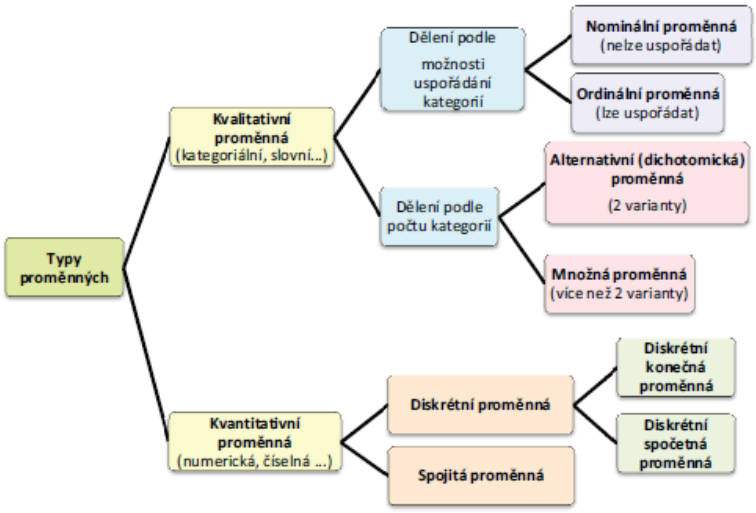
Pak se závěry ze zkoumání výběru přenesou na celou populaci – **statistická indukce**. Mezi metody využívající statistickou indukci patří teorie odhadů a testování hypotéz.

Standartní datový formát – formát zkoumaných dat, kde každý řádek matice obsahuje údaje o jedné statistické jednotce a v prvním sloupci je obvykle identifikační číslo statistické jednotky.

Proměnné

- Údaje, které u výběrového souboru sledujeme
- Obsahují co největší množství informací obsažených v původním souboru
- Dělí se na:
  - **Kvalitativní proměnná** (kategoriální ,slovní) – proměnné, kterou nemůžeme měřit, můžeme ji pouze zařadit do tříd. Varianty kvalitativní proměnné nazýváme kategoriemi, jsou vyjádřeny slovně a podle vztahu mezi jednotlivými kategoriemi se dělí na dvě základní podskupiny:
    - **Nominální proměnná** – nabývá rovnocenných variant, nelze je porovnávat ani seřadit (např. pohlaví, národnost, značka)

        - **Ordinální proměnná** – přechod mezi kvalitativními a kvantitativními proměnnými; jednotlivým variantám lze přiřadit pořadí a vzájemně je porovnávat a řadit (např. známka ve škole, velikost oděvů)
    - **Kvantitativní proměnná** – měřitelné proměnné. Jsou vyjádřeny číselně a dělí se na:
        - **Diskrétní proměnné** – nabývají konečného nebo spočetného množství variant (známka z matematiky nebo věk v letech)
        - **Spojitá proměnná** – nabývají libovolných hodnot z reálných čísel (výška, váha, vzdálenost)



Obr. 1.1: Demonstrace základních proměnných

Číselné charakteristiky

Pro kvalitativní proměnnou

Četnost

Absolutní četnost je definována jako počet výskytů nominální kvantitativní proměnné.

Relativní četnost

Definována jako $p_i = \frac{n_i}{n}$, kde n_i je absolutní četnost dané kategorie a n je celkový počet hodnot. Vyjadřuje, jakou část souboru tvoří proměnné s některou variantou. Součet relativních četností musí být roven 1.

Při zpracování kvantitativní proměnné je vhodné četnosti i relativní četnosti uspořádat do **tabulky rozdělení četností.**

TABULKA ROZDĚLENÍ ČETNOSTI		
Typ pasažéra	Absolutní četnosti	Relativní četnosti (%)
Muž	77	37,4
Žena	85	41,3
Dítě	44	21,3
Celkem:	206	100,0

Dopočet do 100%!

Pozor u zaokrouhlování relativní četnosti – poslední četnost zaokrouhlíme tak, aby byl součet rel. Četností roven 1 (100%).

Modus

Modus je název varianty proměnné (varianta = hodnota) vykazující nejvyšší četnost. Modus můžeme chápat jako typického reprezentanta souboru. V případě, že se nachází více variant se stejnou maximální četností, modus se neurčuje.

Kumulativní četnost (pro ordinální proměnné)

Ordinální proměnná nabývá v rámci souboru různých slovních variant, avšak tyto varianty mají přirozené uspořádání (můžeme je řadit). Pro ordinální proměnné se používají stejné statistické charakteristiky a grafy jako pro nominální proměnné + další dvě charakteristiky.

Kumulativní četnost je počet hodnot proměnné, které nabývají varianty nižší nebo rovné i-té variantě. Např. v případě známek kumulativní četnost pro variantu „prospěl“ bude rovna počtu studentů se známkou „prospěl“ nebo lepší („velmi dobře“, „výborně“). Kumulativní četnost nejvyšší varianty je rovná rozsahu proměnné.

Kumulativní relativní četnost (pro ordinální proměnné)

Vyjadřuje, jakou část souboru tvoří hodnoty nabývající i-té a nižší varianty. Je to relativní vyjádření kumulativní četnosti: $F_i = \frac{m_i}{n}$

Relativní četnosti lze také reprezentovat tabulkou.

Pro kvantitativní proměnnou

Pro popis numerické proměnné můžeme využít většinu charakteristik pro popis ordinální proměnné + další dvě skupiny charakteristik:

- **Míry polohy** – určují typické rozložení hodnot proměnné (jejich umístění na číselné ose). Odhadují skutečnou populační střední hodnotu na základě výběrového souboru. Patří mezi ně *výběrový aritmetický průměr, výběrový geometrický průměr, výběrový medián a modus a kvantily*
- **Míry variability** – určují rozptyl hodnot kolem své typické hodnoty. Patří mezi ně *variační rozpětí, mezikvartilové rozpětí, rozptyl, směrodatná odchylka a variační koeficient*
- **Míry šikmosti a špičatosti –** sledují tvary rozdělení dat. Patří sem *výběrová šikmost a výběrová špičatost*

Míry polohy

- **Aritmetický průměr**
  - Průměr představuje průměrnou nebo typickou hodnotu výběrového souboru.
  - $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$, kde x jsou jednotlivé hodnoty proměnné a n je počet hodnot proměnné
  - Součet hodnot vydělený počtem hodnot
  - **Vážený aritmetický průměr –** hodnoty jsou před sečtením vynásobeny svou četností
  - Průměr není rezistentní vůči odlehlým pozorováním, všeobecně je průměr (platí pro všechny průměry) velmi citlivý na odlehlá pozorování, které dokážou průměr vychýlit natolik, že přestává daný výběr reprezentovat.
- **Harmonický průměr**
  - Pro výpočet průměru v případech, kdy proměnná má charakter části z celku, používáme harmonický průměr, který je definován vztahem $\bar{x}_H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$
- **Geometrický průměr**

- Používá se tam, kde pracujeme s kladnou proměnnou představující relativní změny (růstové indexy, cenové indexy)
- N-tá odmocnina ze součinu hodnot proměnné: $\bar{x}_G = \sqrt{x_1^{n_1} \cdot x_2^{n_2} \cdot \dots \cdot x_n^{n_k}}$

- **Modus**
  - Pro diskrétní proměnnou je to hodnota nejčetnější varianty proměnné
  - Pro spojitou proměnnou je modus hodnota, kolem níž je největší koncentrace hodnot proměnné. Pro určení modusu potřebujeme **short –** nejkratší interval, v němž leží alespoň 50% hodnot proměnné. Modus pak definujeme jako střed shortu.
- **Výběrové kvantily**
  - Slouží pro podrobnější vyjádření rozložení hodnot proměnné v rámci souboru
  - Podobně jako modus, jsou i kvantily odolné vůči odlehlým pozorováním
  - Hodnota, která rozděluje výběrový soubor na dvě části – první z nich obsahuje hodnoty, které jsou menší než daný kvantil, druhá část obsahuje hodnoty, které jsou rovny nebo větší než daný kvantil.
  - Pro určení je nutné soubor uspořádat od nejmenší po největší
  - **V praxi se setkáváme s těmito kvantily**
    - **Kvartily**
      - **Dolní kvartil $x_{0,25}$:** Rozděluje datový soubor tak, že 25% hodnot je menších než tento kvartil a zbytek (75%) je větších
      - **Horní kvartil $x_{0,75}$**
      - **Medián $x_{0,5}$**: Rozděluje datový soubor na polovinu – polovina hodnot je menších a polovina větších (nebo rovných)
    - **Decily** – $x_{0,1}$, $x_{0,2}$, ..., $x_{0,9}$
    - **Percentily** - $x_{0,01}$, $x_{0,02}$, ..., $x_{0,99}$ – dělí soubor na 100 přibližně stejně četné části
  - **Interkvartilové rozpětí (IQR) –** Míra variability souboru; vzdálenost mezi horním a dolním kvartilem

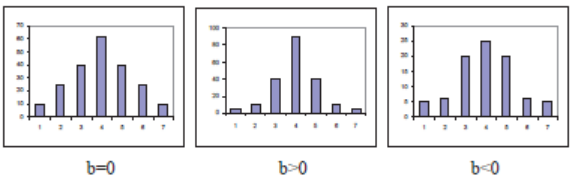
Míry variability

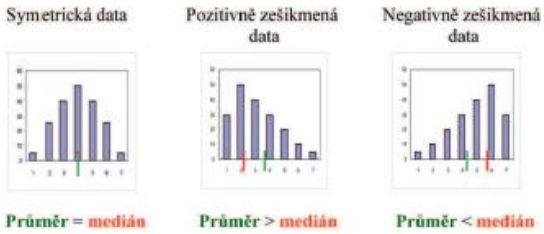
- Čím větší je rozptýlenost hodnot proměnné kolem jejího pomyslného středu, tím menší je schopnost tohoto středu reprezentovat proměnnou
- **Výběrový rozptyl**
  - $s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$
  - Je dán podílem součtu kvadrátů odchylek jednotlivých hodnot od průměru a rozsahu souboru sníženého o jedničku
  - Rozptyl je roven nule, pokud všechny hodnoty proměnné jsou stejné
  - Rozměr rozptylu je druhou mocninnou rozměru proměnné, proto se používá následující charakteristika
- **Výběrová směrodatná odchylka**
  - Odmocnina výběrového rozptylu: $s = \sqrt{s^2}$
  - Nevýhodou výběrového rozptylu a směrodatné odchylky je skutečnost, že neumožňují porovnávat variabilitu proměnných vyjádřených v různých jednotkách. Která proměnná má větší variabilitu – výška nebo hmotnost člověka? Na tuto otázku nám dá odpověď následující charakteristika
- **Variační koeficient**
  - Vyjadřuje relativní míru variability proměnné x

    - Lze stanovit pro proměnné, které nabývají výhradně kladných hodnot
    - Uvádí se v %
    - $V = \frac{s}{|\bar{x}|} \cdot 100$; podíl směrodatné odchylky a aritmetického průměru
    - Čím nižší hodnota, tím homogennější je soubor
- **Interkvartilové rozpětí**
    - $IQR = x_{0,75} - x_{0,25}$
    - Slouží pro identifikaci **odlehlých pozorování –** hodnoty, které se mimořádně liší od ostatních hodnot a tím ovlivňují hodnotu průměru
    - **Identifikace odlehlých pozorování**
        - $(x_i < x_{0,25} - 1{,}5 \cdot IQR) \wedge ((x_i > x_{0,75} + 1{,}5 \cdot IQR)$
        - Identifikace pomocí vnitřních hradeb

Míry šikmosti a špičatosti

- **Výběrová šikmost**
    - 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
- **Výběrová špičatost**
    - Vyjadřuje koncentraci hodnot proměnné kolem jejího průměru
    - b = 0 – špičatost odpovídá normálnímu rozdělení
    - b > 0 – špičaté rozdělení proměnné
    - b < 0 – ploché rozdělení proměnné





Přesnost číselných charakteristik

- směrodatnou odchylku jakožto míru nejistoty měření zaokrouhlujeme **nahoru** na jednu, max dvě platné cifry
- míry polohy zaokrouhlujeme tak, aby nejnižší zapsaný řád odpovídal nejnižšímu zapsanému řádu směrodatné odchylky

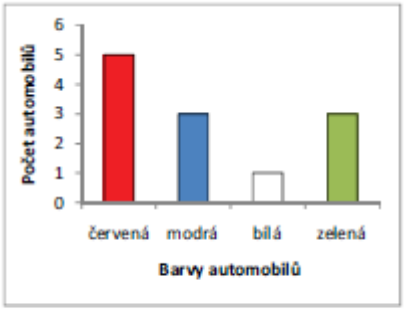
Vizualizace kategoriálních proměnných

Četnost a relativní četnost

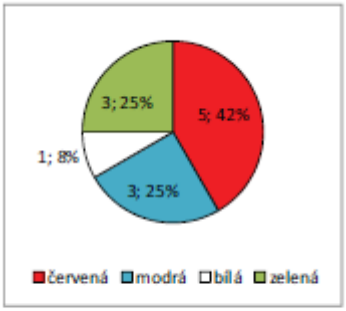
- **Sloupcový graf (histogram)**
    - Jedna osa je pro varianty proměnné a druhá osa je pro jejich četnosti
    - Jednotlivé hodnoty četností jsou zobrazeny jako výšky sloupců
- **Výsečový graf (koláčový)**
    - Znázorňuje poměr mezi hodnotami
    - musí být uvedeny absolutní četnosti v grafu

- Rozdělení četností kvalitativního znaku se znázorňuje kruhovým diagramem, kde různým hodnotám znaku odpovídají kruhové výseče, jejichž plošné obsahy jsou úměrné četnostem.

- **Obrázkové grafy** – např. v novinách, jsou různě zavádějící



Obr. 1.6: Pozorované barvy automobilů - histogram

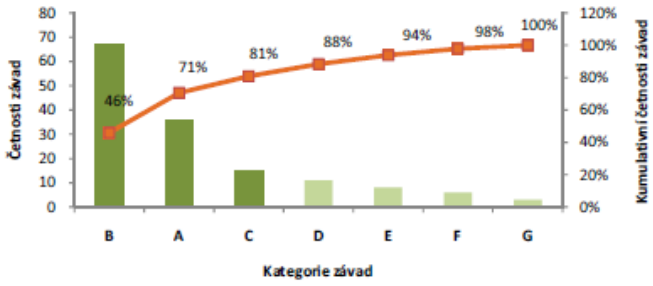


Obr. 1.7: Pozorované barvy automobilů - výsečový graf

Kumulativní a relativní kumulativní četnost

- **Lorenzova křivka**
  - Spojnicový graf, který má na vodorovné ose jednotlivé varianty seřazené od nejmenší do největší a na svislá osa obsahuje příslušně hodnoty kumulativních četností.
- **Paretův graf**
  - Sloučení histogramu proměnné seřazené podle četnosti výskytu (od největší četnosti po nejmenší) a Lorenzovy křivky

Obr. 1.8: Lorenzova křivka

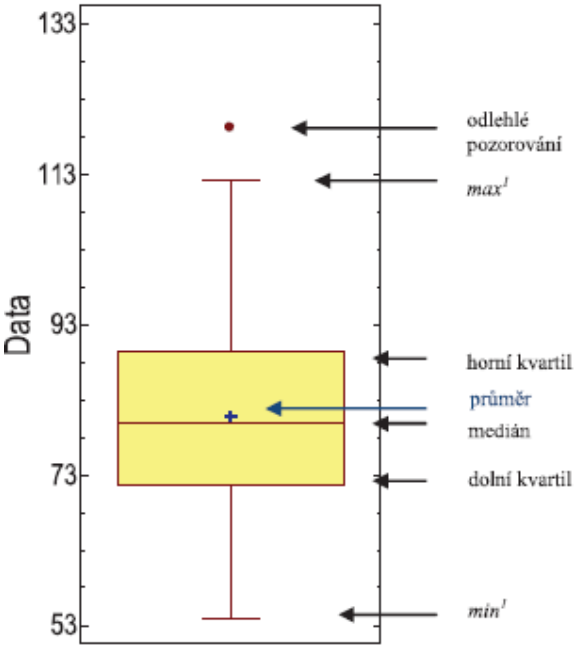


Obr. 1.9: Paretův graf závad

Vizualizace kvantitativních proměnných

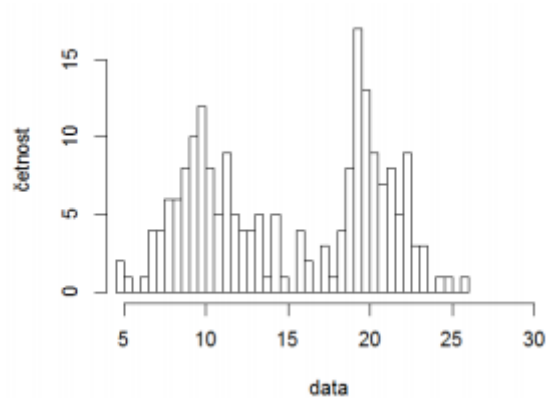
Krabicový graf

Odlehlá pozorování jsou znázorněna jako izolované body, konec úsečky představují maximum (nebo minimim) proměnné po vyloučení odlehlých pozorování. Hrany obdélníku představují horní a dolní kvartil, vodorovná úsečka uvnitř obdélníku představuje medián. Z polohy mediánu vzhledem ke obdélníku lze dobře usuzovat na symetrii vnitřních 50% dat.



Obr. 1.13: Krabicový graf

Histogram



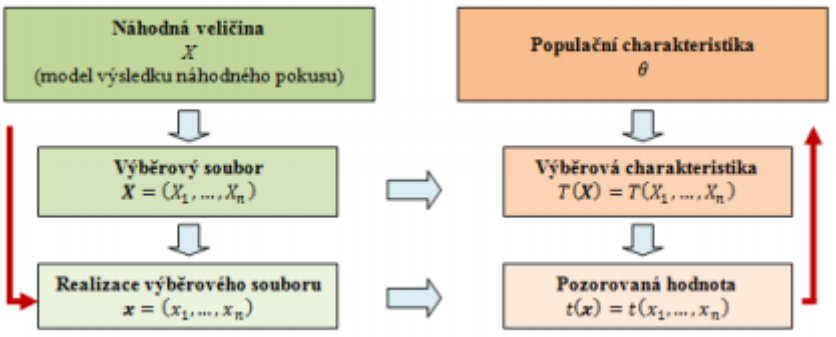
Metody statistické indukce

K modelování a zkoumání populace používáme výběrové soubory. Je-li výběr reprezentativní, dá se na jeho základě získat dobrá představa o vlastnostech populace.

Parametry populace – jsou to konstanty, parametry základního souboru (populace)

Charakteristiky výběru – jsou různé, v závislosti na pořízeném výběru – jsou to náhodné veličiny. Možných výběrů ze základního souboru (populace) může být mnoho a pro každou se vypočítá nějaká charakteristika – pro každý výběr může vyjít jinak. Proto mají výběrové charakteristiky charakter náhodných veličin a lze je popsat nějakým rozdělením (mají svojí střední hodnotu, rozptyl a další).

Princip statistické indukce je založen na tom, že chceme-li získat informace o určitém parametru populace, pak analyzujeme takovou výběrovou charakteristiku, která s velkou pravděpodobností nabývá hodnot blízkých našemu neznámému parametru populace.



Základní soubor (populace)	střední hodnota $E(X)$, resp. μ	medián $x_{0,5}$	rozptyl $D(X)$, resp. σ^2	směrodatná odchylka σ	pravděpodobnost π
Výběrový soubor (výběr)	(výběrový) průměr $\bar{X}$	výběrový medián $\tilde{X}_{0,5}$	výběrový rozptyl S^2	výběrová směrodatná odchylka S	relativní četnost p

Příklad: Průměrný plat 20 občanů ČR je náhodná veličina. Výpočtem průměrného platu konkrétních 20 občanů získáme jednu realizaci tohoto průměru, výpočtem průměrného platu jiného vzorku 20 občanů získáme jinou realizaci průměru.

Limitní věty – popis pravděpodobnostních modelů pro případ rostoucích počtu realizací náhodného pokusu. Přináší tvrzení o vlastnostech výběrového průměru pro případ dostatečně velkého rozsahu náhodného výběru.

- **Slabý zákon velkých čísel**
    - Výběrový průměr z náhodného výběru, který obsahuje celou populaci, je střední hodnota rozdělení, z něhož výběr pochází. Výběrový průměr z náhodného výběru o rozsahu menším než populace není přesně střední hodnota rozdělení, ale je to číslo, které je skutečné střední hodnotě blízko
    - Mějme nekonečný náhodný výběr X_1, X_2… z rozdělení se střední hodnotou a rozptylem, kde X_1, X_2.. jsou nekorelované náhodné veličiny. Potom platí, že výběrový průměr vypočítány z prvních n pozorování se pro n blížící se nekonečnu blíží ke střední hodnotě.
    - Průměr se s rostoucím rozsahem výběru blíží střední hodnotě
    - Relativní četnost se s rostoucím rozsahem výběru blíží pravděpodobnosti
- **Centrální limitní věta (CLV)**
    - Jsou-li X_i nezávislé náhodné veličiny s konečným rozptylem, pak výběrový průměr má při dostatečně velkém počtu pozorování přibližně normální rozdělení, ať už X_i pocházejí z libovolného rozdělení
    - Výběrový průměr má při dostatečně velkém počtu pozorování ($n > 30$) přibližně normální rozdělení, ať už X_i pocházejí z libovolného rozdělení.

Výběrová rozdělení

Výběrová rozdělení – rozdělení pravděpodobnosti výběrových charakteristik

Výběrová rozdělení nacházejí uplatnění při odhadech střední hodnoty a pravděpodobnosti, resp. jejich rozdílů nebo při testování hypotéz o těchto parametrech.

Na základě CLV byla popsána rozdělení:

- Výběrového průměru při dostatečném rozsahu výběru, resp. při výběru z normálního rozdělení
- Rozdělení relativní četnosti při dostatečném rozsahu výběru
- Rozdělení rozdílu průměrů dvou nezávislých výběrů z normálního rozdělení
- Rozdílu relativních četností dvou dostatečně velkých nezávislých výběrů

Při odhadech rozptylu, poměrů rozptylů, odhadech střední hodnoty v případě, že máme k dispozici pouze malý výběr, který nepochází z normálního rozdělení, a v dalších metodách statistické indukce nacházejí uplatnění tři důležitá spojitá rozdělení – chí kvadrát rozdělení, studentovo rozdělení a Fisherovo-Snedecorovo rozdělení.

Chí-kvadrát rozdělení (Pearsnovo)

- Parametrem je stupeň volnosti v (označuje počet sčítaných nezávislých náhodných veličin)
- **Vlastnosti:**
    - Střední hodnota je rovna *v*
    - Rozptyl je roven dvojnásobku *v*
    - Se vzrůstajícím počtem stupňů volnosti *v* se rozdělení blíží normálnímu rozdělení
    - Výběr je z normálního rozdělení
- **Použití:**
    - Testování toho, zda rozptyl základního souboru s normálním rozdělením je roven známému rozptylu, resp. odhadování směrodatné odchylky základního souboru s norm. rozdělením
    - Ověření nezávislosti kategoriálních proměnných

- Test dobré shody – zda náhodné veličiny pocházejí z určitého rozdělení

Studentovo rozdělení

- Parametrem je stupeň volnosti *v*
- **Použití**
  - Modelování založené na analýze malých výběrů (n < 30)
  - Testování hypotéz o střední hodnotě, pokud je rozptyl populace neznámý a výběr pochází z normálního rozdělení
  - Testování hypotéz o shodě středních hodnot, za předpokladu, že máme dispozici dva nezávislé výběry z normálních rozdělení, jejichž rozptyly jsou neznámé, ale shodné
  - Analýza výsledků regresní analýzy

Fisherovo-Snedecorovo rozdělení

- Parametrem jsou dva stupně volnosti *m* a *n*
- Použití:
  - K testu o shodě rozptylů dvou základních souborů
  - K testům o shodě středních hodnot více než dvou základních souborů – analýza rozptylu
  - K testům v regresní analýze

Něco navíc

- Pro modelování průměru výběru dostatečně velkého rozsahu je vhodné použít **normální** rozdělení
- Pro modelování průměru výběru malého rozsahu je vhodné použít **Studentovo** rozdělení
- Pro modelování relativní četnosti ve výběru o dostatečném rozsahu je vhodné použít **normální** rozdělení
- Pro modelování rozptylu výběru z normálního rozdělení je vhodné použít **Pearsonovo** rozdělení
- Pro modelování poměru rozptylů dvou výběrů z normálního rozdělení je vhodné použít **Fisherovo-Snedecorovo** rozdělení

Intervalové odhady

V praktických případech většinou nedokážeme přesně určit parametry základního souboru (populace). K jejich odhadu používáme charakteristiky příslušného výběrového souboru – výběrové charakteristiky.

Používáme dva typy odhadů parametrů populace z výběrových rozdělení charakteristik:

- **Bodový odhad** – parametr základního souboru (populace) aproximujeme jediným číslem
- **Intervalový odhad** – parametr aproximujeme intervalem, v němž s velkou pravděpodobností příslušný populační parametr leží

Intervalový odhad je reprezentován intervalem (t_P, t_H) v němž hledaný parametr leží s předem určenou pravděpodobností (spolehlivostí), kterou označujeme 1 – α. Intervalový odhad je jednou z realizací intervalu spolehlivosti.

Interval spolehlivosti pro parametr θ se spolehlivostí 1 – α je taková dvojice statistik (T_P, T_H), že:

$$P(T_P \leq \theta \leq T_H) = 1 - \alpha$$

Spolehlivost odhadu 1 – α udává, že při opakovaných výběrech s konstantním rozsahem *n* z dané populace přibližně $100 \cdot (1 - \alpha)\%$ intervalových odhadů obsahuje skutečnou hodnotu odhadovaného parametru a naopak 100α% intervalových odhadů skutečnou hodnotu odhadovaného parametru neobsahuje.

Jde nám o

- Co největší spolehlivost odhadu
- Co nejmenší šířka intervalu spolehlivosti (S rostoucí šířkou intervalového odhadu klesá významnost získané informace)

Rostoucí šířka intervalového odhadu ubírá na jeho vypovídající schopnosti, jeho významnost klesá. Čím vyšší je spolehlivost odhadu, tím širší intervalový odhad získáme. Je nutné najít kompromis mezi spolehlivostí a **významností** odhadu. **Hladina významnosti** je α, s rostoucí spolehlivostí odhadu klesá hladina významnosti. V praxi se hladina významnosti dává jako 5%.

Typy intervalů spolehlivosti

- Oboustranné
- Jednostranné – levo nebo pravostranné
    - Pokud např. odhadujeme délku života nějakého zařízení, je pro nás důležitá jen dolní mez

Určení intervalu spolehlivosti

Parametry, pro které chceme určit intervaly spolehlivosti musí být z normálního rozdělení. V případě, že základní soubor nemá normální rozdělení, musíme přistoupit k tzv. **neparametrickým metodám odhadu,** které se použijí v případě, že**:**

- Výběrový soubor obsahuje odlehlá pozorování, která nemohou být opravena a není možné je vyloučit
- Výběrový soubor nepochází z normálního rozdělení
- Výběrový souhlas má velké rozptýlení dat

Podle toho, jaký parametr chceme odhadnout a jaký parametr známe a zda jsou splněné předpoklady, určujeme intervalový odhad podle následující tabulky:

Odhadovaný parametr		Předpoklady	Meze oboustranného intervalového odhadu		Dolní mez levostranného intervalového odhadu	Horní mez pravostranného intervalového odhadu
			T_D	T_H	T_D	T_H
Míra polohy	μ	normalita, známe σ	$\bar{x}-\frac{\sigma}{\sqrt{n}}z_{1-\frac{\alpha}{2}}$	$\bar{x}+\frac{\sigma}{\sqrt{n}}z_{1-\frac{\alpha}{2}}$	$\bar{x}-\frac{\sigma}{\sqrt{n}}z_{1-\alpha}$	$\bar{x}-\frac{\sigma}{\sqrt{n}}z_{1-\alpha}$
		normalita, neznáme σ	$\bar{x}-\frac{s}{\sqrt{n}}t_{1-\frac{\alpha}{2}}$	$\bar{x}+\frac{s}{\sqrt{n}}t_{1-\frac{\alpha}{2}}$	$\bar{x}-\frac{s}{\sqrt{n}}t_{1-\alpha}$	$\bar{x}+\frac{s}{\sqrt{n}}t_{1-\alpha}$
Míry variability	σ^2	normalita	$\frac{(n-1)s^2}{\chi_{1-\frac{\alpha}{2}}}$	$\frac{(n-1)s^2}{\chi_{\frac{\alpha}{2}}}$	$\frac{(n-1)s^2}{\chi_{1-\alpha}}$	$\frac{(n-1)s^2}{\chi_{\alpha}}$
	σ	normalita	$\sqrt{\frac{(n-1)s^2}{\chi_{1-\frac{\alpha}{2}}}}$	$\sqrt{\frac{(n-1)s^2}{\chi_{\frac{\alpha}{2}}}}$	$\sqrt{\frac{(n-1)s^2}{\chi_{1-\alpha}}}$	$\sqrt{\frac{(n-1)s^2}{\chi_{\alpha}}}$
Relativní četnost	π	$n>30$, $\frac{n}{N}<0{,}05$, $n>\frac{9}{p(1-p)}$	$p-z_{1-\frac{\alpha}{2}}\sqrt{\frac{p(1-p)}{n}}$	$p+z_{1-\frac{\alpha}{2}}\sqrt{\frac{p(1-p)}{n}}$	$p-z_{1-\alpha}\sqrt{\frac{p(1-p)}{n}}$	$p+z_{1-\alpha}\sqrt{\frac{p(1-p)}{n}}$

Intervalové odhady můžeme použít také ke srovnávání středních hodnot, rozptylů, resp. relativních četností dvou populací.

Princip testování hypotéz

Mezi základní metody statistické indukce patří intervalové odhady a **testování hypotéz**. Testování hypotéz umožňuje usoudit, zda experimentálně získaná data nepopírají předpoklad, který jsme před provedením testování učinili.

Cílem výzkumů mnohdy bývá srovnání účinnosti různých metod (např. srovnání úmrtnosti u klasických a laparoskopických operací) či srovnání výsledků různých skupin (např. porovnání výsledků srovnávacích testů u absolventů odborných učilišť a gymnázií). Jinými slovy, cílem bývá prokázat nějaký rozdíl parametrů náhodných veličin. Náš předpoklad ohledně efektu nazýváme **statistickou hypotézou**. Pro ověření správnosti vyslovené hypotézy použijeme vhodný výběrový soubor. Proces ověřování správnosti statistické hypotézy pomocí výsledků získaných z výběrového šetření se nazývá **testování hypotéz.**

Základní pojmy

Statistická hypotéza – předpoklad (výrok, tvrzení) o rozdělení pozorované náhodné veličiny zakládající se na předchozí zkušenosti, na rozboru dosavadních znalostí nebo na pouhé domněnce. Dělí se na:

- **Parametrická hypotéza** – pojednává o parametrech rozdělení náhodné veličiny (střední hodnota, medián, rozptyl)
    - Hypotézy o parametrech dvou populací (srovnávací testy**)**
    - Hypotézy o parametrech více než dvou populací (ANOVA, Kruskalův-Wallisův test)
- **Neparametrická hypotéza** – týká se jiných vlastností náhodné veličiny (typ rozdělení, nezávislost výběru)
    - Hypotézy o typu rozdělení NV, hypotézy o závislosti NV

Příklady statistických hypotéz

- Střední životnost žárovek je nižší než výrobcem udávaných 5 let
- Mortalita je u laparoskopických operací nižší než u operací konvenčních

- Pořízený datový soubor je výběrem z populace mající normální rozdělení.
- Můžeme zapisovat jako rovnosti mezi testovaným parametrem a jeho předpokládanou hodnotou: střední hodnota obsahu cholesterolu je 4,7

Testování hypotéz je rozhodovací proces, v němž proti sobě stojí dvě tvrzení

- **Nulová hypotéza H_0** (testovaná hypotéza) představuje tvrzení, že sledovaný efekt je nulový a bývá vyjádřena rovností mezi testovaným parametrem a jeho očekávanou hodnotou. Efekt je nulový, resp. neexistuje závislost, že data mají určitý typ rozdělení
- **Alternativní hypotéza H_A**, která popírá tvrzení dané nulovou hypotézou. Obvykle to, co chceme dokázat.

Přiklad:

Zadání problému: Ověřte, zda průměrný plat je větší než 24000Kč.
Populace: všichni občané ČR pobírající mzdu
Náhodná veličina: mzda
Nulová hypotéza: $\mu = 24000$
Alternativní hypotéza: $\mu > 24000$

Zadání problému: Ověřte, zda průměrné mzdy ve strojírenství a v hutnictví jsou stejné.
Populace 1: všichni občané pracují ve strojírenství
Populace 2: všichni občané pracují ve hutnictví
Náhodná veličina: mzda
Nulová hypotéza: $\mu_S = \mu_H$
Alternativní hypotéza: $\mu_S <> \mu_H$

Test statistické hypotézy

- Nulovou hypotézu považujeme za pravdivou až do okamžiku, kdy nás informace získané z výběrového souboru přesvědčí o opaku.
- Můžeme dojít ke dvěma rozhodnutím: Zamítáme nulovou hypotézu ve prospěch alternativní hypotézy (kritický obor) nebo nezamítáme nulovou hypotézu (obor přijetí)
- **Při rozhodování nastane vždy některý z těchto případů:**
    - Chyba I druhu: nulová hypotéza je ve skutečnosti platná, ale mi ji zamítneme. Pravděpodobnost, že k takové chybě dojde se nazývá **hladina významnosti α**
    - Chyba II. Druhu – nezamítnutí nulové hypotézy v případě, že je platná hypotéza alternativní.
    - Chceme mít testy s nízkou hladinou významnosti a vysokou sílou testu
    - S klesající hladinou významnosti roste pravděpodobnost chyby 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)

Při testování hypotéz se běžně setkáváme se dvěma přístupy: **klasickým testem** a **čistým testem významnosti.**

Čistý test významnosti

- Formulace nulové a alternativní hypotézy
- Volba testové statistiky (kritéria)
- Ověření předpokladů testu
- Výpočet pozorované hodnoty x_{OBS} testové statistiky
- Výpočet p-hodnoty
- Rozhodnutí o výsledku testu na základě p-hodnoty:

$p-hodnota$	Rozhodnutí
$p-hodnota < \alpha$	Zamítáme H_0 ve prospěch H_A.
$p-hodnota \geq \alpha$	Nezamítáme H_0.

- P-hodnota je nejnižší hladina významnosti, na níž můžeme nulovou hypotézu zamítnout
- P-hodnota říká, jaká je minimální hladina významnosti, na níž bychom při daném výběrovém souboru mohli nulovou hypotézu zamítnout.

Něco navíc

Jednovýběrové neparametrické testy

- Rozhodnutí, zda neznámý parametr rozdělení populace (střední hodnota, rozptyl nebo relativní četnost) je roven nějaké konkrétní hodnotě

Typ proměnné	Požadovaný typ analýzy	Předpoklady	Testy, resp. intervalové odhady
Spojitá proměnná	Ověření variability	Normalita	Test o rozptylu (test o směr. odchylce)
			Intervalový odhad rozptylu (směr. odchylky)
	Ověření polohy	Normalita	Studentův t-test, (test o střední hodnotě)
			Intervalový odhad střední hodnoty
		Výběr většího rozsahu	Znaménkový test (test o mediánu)
		Symetrické rozdělení	Wilcoxonův test (test o mediánu)
Dichotomická proměnná (0-1)	Ověření shody relativní četnosti s očekávanou pravděpodobností	$n > \frac{9}{p(1-p)}$	Test o parametru π binomického rozdělení
			Intervalový odhad parametru π binomického rozdělení

Dvouvýběrové testy parametrických hypotéz

- Pro nezávislé výběry umožňují na základě dvou nezávislých výběrů porovnat neznámé parametry dvou populací

Typ proměnné	Požadovaný typ analýzy	Předpoklady		Testy, resp. intervalové odhady
Dvě nezávislé spojité proměnné	Ověření shody rozptylů (homoskedasticity)	Normalita		F-test (test shody rozptylů)
				Intervalový odhad *poměru* rozptylů, resp. směr. odchylek
		—		Leveneův test
	Ověření shody měr polohy (středních hodnot, resp. mediánů)	Normalita	Shoda rozptylů (homoskedasticita)	Dvouvýběrový Studentův t-test (test shody stř. hodnot)
				Intervalový odhad rozdílu stř.hodnot
			Různé rozptyly (heteroskedasticita)	Aspinové-Welchův test (test shody stř. hodnot)
				Intervalový odhad rozdílu stř.hodnot
		—		Mannův-Whitneyův test test shody mediánů
Párová (spojitá) data	Ověření shody úrovně párových dat	Normalita		Párový studentův t-test
				Intervalový odhad střední hodnoty rozdílů
		Výběry většího rozsahu		Párový znaménkový test
		Symetrické rozdělení		Wilcoxnův párový test
Dvě dichotomické proměnné	Ověření shody pravděpodobností	$n_i > \frac{9}{p_i(1-p_i)}, i = 1,2$		Test homogenity dvou binomických rozdělení
				Intervalový odhad rozdílu parametru binomických rozdělení

Vícevýběrové testy parametrických hypotéz

- ANOVA test
    - Předpoklady:
        - Nezávislost výběru
        - Normalita rozdělení
        - Homoskedasticita (identické rozptyly)
- Kruskalův-Wallisův test
    - Když nejsou splněny předpoklady ANOVA testu, tam to musí splňovat jen nezávislost výběru

Testy dobré shody – Slouží k rozhodnutí, zda dané rozdělení se shoduje s teoretickým rozdělením
Kolmogorovův-Smirnovův test – používá se k ověření hypotézy, zda pořízený výběr pochází z rozdělení se spojitou distribuční funkcí