Úvod do informatiky

- Přednášející: **Petr Hliněný** (`hlineny@fi.muni.cz`)
- Konzultanti kurzu: **Jan Holeček** (`holecek@fi.muni.cz`)

  Jan Obdržálek (`xobdrzal@fi.muni.cz`)
- Materiály a informace dostupné na

  `http://www.fi.muni.cz/~hlineny/Vyuka/UINF.html`

  Studenti jsou aktuální informace na tomto webu povinni pravidelně číst!

Smysl a formát kurzu

- Elementárně pojatý úvod do matematiky „vysokoškolského typu“ s ohledem na potřeby studia informatiky.
- Aplikace zapamatovaných vzorců a vět (SŠ) $\times$ styl definice-věta-důkaz (VŠ).
- Úvodní pasáže jsou obsahově velmi podobné kurzu „Matematické základy I“.
- Cíl: příprava pro další studium, vyjasnění základních pojmů.
- Evaluace:
  - 30% dvě písemky v semestru ($2\times$ 15%),
  - 70% zkoušková písemka,
  - až 5% bonus za řešení jednoho domácího projektu (nenárokové!), který bude zadaný příště.
- K absolvování kurzu je třeba zhruba 50% z celkového počtu bodů.

Úvod do formálního dokazování

- Uvažme matematickou větu tvaru „Jestliže platí *předpoklady*, pak platí *závěr*“.
- Důkaz této věty je konečná posloupnost tvrzení, kde
  - každé tvrzení je buď
    - předpoklad, nebo
    - obecně přijatá „pravda“ – axiom, nebo
    - plyne z předchozích a dříve dokázaných tvrzení podle nějakého „akceptovaného“ logického principu – odvozovacího pravidla;
  - poslední tvrzení je *závěr*.

Sudé číslo je celé číslo dělitelné 2, tj. tvaru $2k$.
Liché číslo je celé číslo nedělitelné 2, tj. tvaru $2k+1$.

Příklad 1. Věta: *Jestliže x je součtem dvou lichých čísel, pak x je sudé.*

Důkaz.

	tvrzení	zdůvodnění
1)	$a = 2k+1$, k celé	předpoklad
2)	$b = 2l+1$, l celé	předpoklad
3)	$x = a+b = 2k+2l+1+1$	1,2) a komutativita sčítání (axiom)
4)	$x = 2(k+l)+2\cdot 1$	3) a distributivnost násobení
5)	$x = 2(k+l+1)$	4) a opět distributivnost násobení

□

Příklad 2. Věta: *Jestliže* x *a* y *jsou racionální čísla pro která platí* $x < y$*, pak existuje racionální číslo* z *pro které platí* $x < z < y$*.*

Důkaz.

- Nechť $z = \frac{x+y}{2} = x + \frac{y-x}{2} = y - \frac{y-x}{2}$.
- Číslo z je racionální, neboť x a y jsou racionální.
- Platí $z > x$, neboť $\frac{y-x}{2} > 0$.
- Dále platí $z < y$, opět neboť $\frac{y-x}{2} > 0$.
- Celkem $x < z < y$.

□

Alternativní formulace Věty 2:

- Pro každé $x, y \in \mathbb{Q}$, kde $x < y$, existuje $z \in \mathbb{Q}$ takové, že $x < z < y$.
- Množina racionálních čísel je hustá.

Struktura matematických vět

- První krok formálního důkazu je uvědomit si, co se má dokázat, tedy co je předpoklad a co závěr.
- Příklady:
  - Věta: Konečná množina má konečně mnoho podmnožin.
  - Věta: $\sin^2(\alpha) + \cos^2(\alpha) = 1$.
  - Věta: Graf je rovinný jestliže neobsahuje podrozdělení K_5 nebo $K_{3,3}$.
- Často pomůže pouhé rozepsání definic pojmů, které se v dané větě vyskytují.
- Jak „moc formální“ mají důkazy vlastně být?
  - Záleží na tom, komu je důkaz určen — „konzument“ musí být schopen „snadno“ ověřit korektnost každého tvrzení v důkazu a plně pochopit, z čeho vyplývá.

Konstruktivní a existenční důkazy

- Důkaz Věty 2 je konstruktivní. Dokázali jsme nejen, že číslo z existuje, ale podali jsme také návod, jak ho pro dané x a y sestrojit.
- Existenční důkaz je takový, kde se prokáže existence nějakého objektu bez toho, aby byl podán návod na jeho konstrukci.

Příklad 3. Věta: *Existuje program, který vypíše na obrazovku čísla tažená v 25. tahu sportky v roce 2006.*

Důkaz. Existuje pouze konečně mnoho možných výsledků losování 25. tahu sportky v roce 2006. Pro každý možný výsledek existuje program, který tento daný výsledek vypíše na obrazovku. Mezi těmito programy je tedy jistě ten, který vypíše právě ten výsledek, který bude v 25. tahu sportky v roce 2006 skutečně vylosován. □

(„Podvod“, nebo ne? Je to prostě tak. . .)

- Pro informatické disciplíny (teorie automatů, teorie složitosti, návrh algoritmů, apod.) je důležitá konstruktivnost důkazů vět, které se zde objevují.
- V matematice jsou mnohé příklady užitečných *existenčních důkazů*, třeba tzv. pravděpodobnostní důkazy.

Příklad 4. Věta: *Na dané množině bodů je zvoleno libovolně* n^2 *podmnožin, každá o* n *prvcích. Dokažte pro dostatečně velká* n*, že všechny body lze obarvit dvěma barvami tak, aby žádná množina nebyla jednobarevná.*

Důkaz. U každého bodu si „hodíme mincí“ a podle výsledku zvolíme barvu červeně nebo modře. (Nezávislé volby s pravděpodobností $\frac{1}{2}$.) Pravděpodobnost, že zvolených n bodů vyjde jednobarevných, je jen $\frac{2}{2^n} = 2^{1-n}$.

Pro n^2 podmnožin tak je pravděpodobnost, že některá z nich vyjde jednobarevná, shora ohraničená součtem

$$\underbrace{2^{1-n} + \ldots + 2^{1-n}}_{n^2} = \frac{n^2}{2^{n-1}} \to 0\,.$$

Jelikož je toto číslo (pravděpodobnost) pro $n \geq 7$ menší než 1, bude existovat obarvení bez jednobarevných zvolených podmnožin. □

Základní důkazové techniky

- Přímé odvození. To je to, o čem jsme se dosud bavili.
- Kontrapozice (také obrácením či nepřímý důkaz). Místo věty

  „Jestliže platí *předpoklady*, pak platí *závěr*.“

  budeme dokazovat větu

  „Jestliže neplatí *závěr*, pak neplatí alespoň jeden z *předpokladů*.“
- Důkaz sporem. Místo věty

  „Jestliže platí *předpoklady*, pak platí *závěr*.“

  budeme dokazovat větu

  „Jestliže platí *předpoklady* a platí *opak závěru*, pak platí opak jednoho z *předpokladů* (nebo platí jiné *zjevně nepravdivé tvrzení*).“
- Matematická indukce. Pokročilá technika. . .

Příklad důkazu kontrapozicí

Prvočíslo $\mathrm{p} > 1$ nemá jiné dělitele než 1 a p.

Příklad 5. Věta: *Jestliže* p *je prvočíslo větší než* 2*, pak* p *je liché.*

Důkaz. Kontrapozicí. (Budeme tedy dokazovat, že je-li p sudé, pak p buď není větší než 2, nebo p není prvočíslo.) Jsou dvě možnosti:

- $\mathrm{p} \leq 2$. Pak p není větší než 2.
- $\mathrm{p} > 2$. Pak $\mathrm{p} = 2.\mathrm{k}$ pro nějaké celé $\mathrm{k} > 1$, tedy p není prvočíslo. □

Důkazy kontrapozicí pracují s negací (opakem) *předpokladů* a *závěru*. Je-li např. *závěr* komplikované tvrzení tvaru

„z toho, že z A a B plyne C vyplývá, že z A nebo C plyne A a B“,

není pouhou intuicí snadné zjistit, co je vlastně jeho negací. Jak uvidíme, užitím jednoduché induktivní metody lze podobná tvrzení negovat zcela mechanicky.

Příklady důkazu sporem

Příklad 6. Věta: *Jestliže* p *je prvočíslo větší než* 2*, pak* p *je liché.*

Důkaz. Sporem. Nechť tedy p je prvočíslo větší než 2, které je sudé. Pak $\mathrm{p} = 2.\mathrm{k}$ pro nějaké $\mathrm{k} > 1$, tedy p není prvočíslo, spor (s předpokladem, že p je prvočíslo). □

Příklad 7. Věta: *Číslo* $\sqrt{2}$ *není racionální.*

Důkaz. Sporem. Nechť tedy $\sqrt{2}$ je racionální, tj. nechť existují nesoudělná celá kladná čísla r, s taková, že $\sqrt{2} = \mathrm{r}/s$. Pak $2 = \mathrm{r}^2/s^2$, tedy $\mathrm{r}^2 = 2.s^2$, proto r^2 je dělitelné dvěma. Z toho plyne, že i r je dělitelné dvěma (proč?). Jelikož r je dělitelné dvěma, je r^2 dělitelné dokonce čtyřmi, tedy $\mathrm{r}^2 = 4.\mathrm{m}$ pro nějaké m. Pak ale také $4.\mathrm{m} = 2.s^2$, tedy $2.\mathrm{m} = s^2$ a proto s^2 je dělitelné dvěma. Z toho plyne, že s je také dělitelné dvěma. Celkem dostáváme, že r i s jsou dělitelné dvěma, jsou tedy soudělná, spor. □

„Nevíte-li, jak nějakou větu dokázat, zkuste důkaz sporem.“

Matematická indukce

- Důkazová technika aplikovatelná na tvrzení tohoto typu:

  „Pro každé přirozené (celé) $n \geq k_0$ *platí* $T(n)$*.“*

  Zde k_0 je nějaké pevné přir. číslo a $T(n)$ je tvrzení parametrizované číslem n.

- Příklad:
  Pro každé $n \geq 0$ platí, že n přímek dělí rovinu nejvýše na $\frac{1}{2}n(n+1)+1$ oblastí.

- Princip matematické indukce říká (coby axiom), že k důkazu věty

  „Pro každé přirozené $n \geq k_0$ *platí* $T(n)$*.“*

  stačí ověřit platnost těchto dvou tvrzení:

  * $T(k_0)$ (tzv. *báze* neboli základ indukce)
  * *Pro každé* $n \geq k_0$*; jestliže platí* $T(n)$*,* *(*indukční předpoklad*)*
    pak platí také $T(n+1)$*.* *(*indukční krok*)*

Příklady důkazů indukcí

Příklad 8. Věta: *Pro každé* $n \geq 1$ *je stejná pravděpodobnost, že při současném hodu* n *kostkami bude výsledný součet sudý, jako, že bude lichý.*

Důkaz. Základ indukce je zde zřejmý: Na jedné kostce (poctivé!) jsou tři lichá a tři sudá čísla, takže obě skupiny padají se stejnou pravděpodobností.

Indukční krok pro $n \geq 1$: Nechť p_n^s pravděpodobnost, že při hodu n kostkami bude výsledný součet sudý, a p_n^l je pravděpodobnost lichého. Podle indukčního předpokladu je $p_n^s = p_n^l = \frac{1}{2}$.

Hoďme navíc $(n+1)$-ní kostkou. Podle toho, zda na ní padne liché nebo sudé číslo, je pravděpodobnost celkového sudého součtu rovna

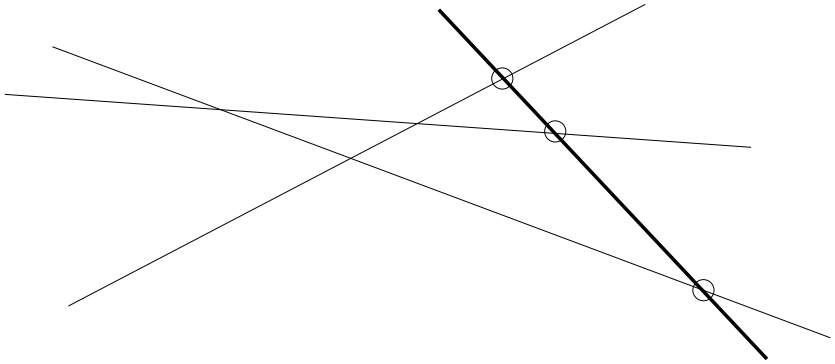
$$\frac{3}{6}p_n^l + \frac{3}{6}p_n^s = \frac{1}{2}$$

a stejně pro pravděpodobnost celkového lichého součtu. □

Příklad 9. Věta: *Pro každé* $n \geq 0$ *platí, že* n *přímek dělí rovinu nejvýše na*

$$\frac{1}{2}n(n+1)+1$$

oblastí.



Důkaz. Pro bázi indukce stačí, že 0 přímek dělí rovinu na jednu část. (Možná je lépe si ještě ověřit, že 1 přímka dělí rovinu na dvě části, jen pro lepší pochopení důkazu.)

Mějme nyní rovinu rozdělenou n přímkami na nejvýše $\frac{1}{2}n(n+1)+1$ částí. Další, $(n+1)$-ní přímka je rozdělena průsečíky s předchozími přímkami na nejvýše $n+1$ úseků a každý z nich oddělí novou část roviny. Celkem tedy bude rovina rozdělena našimi přímkami na nejvýše tolik částí:

$$\frac{1}{2}n(n+1)+1 \;+\; (n+1) \;=\; \frac{1}{2}n(n+1)+\frac{1}{2}\cdot 2(n+1) \;+\; 1 \;=\; \frac{1}{2}(n+1)(n+2) \;+\; 1$$

□

Příklad 10. Věta: *Pro každé* $n \geq 0$ *platí* $\sum_{j=0}^{n} j = \frac{n(n+1)}{2}$.

Důkaz. Indukcí vzhledem k n.

- Báze: Musíme dokázat tvrzení $T(0)$, což je v tomto případě rovnost $\sum_{j=0}^{0} j = \frac{0(0+1)}{2}$. Tato rovnost (zjevně) platí.

- Indukční krok: Musíme dokázat následující tvrzení:

  Jestliže platí $\sum_{j=0}^{i} j = \frac{i(i+1)}{2}$ *kde* $i \geq 0$, *pak platí* $\sum_{j=0}^{i+1} j = \frac{(i+1)(i+2)}{2}$.

  Předpokládejme tedy, že $\sum_{j=0}^{i} j = \frac{i(i+1)}{2}$ a pokusme se dokázat, že pak také $\sum_{j=0}^{i+1} j = \frac{(i+1)(i+2)}{2}$. To už plyne úpravou:

$$\sum_{j=0}^{i+1} j = \sum_{j=0}^{i} j + (i+1) = \frac{i(i+1)}{2} + (i+1) = \frac{i(i+1) + 2(i+1)}{2} = \frac{(i+1)(i+2)}{2}$$

□

Matematická indukce (pokračování)

- Základní trik všech důkazů matematickou indukcí je vhodná reformulace tvrzení $T(i+1)$ tak, aby se „odvolávalo“ na tvrzení $T(i)$.
  - Dobře se vždy podívejte, v čem se liší tvrzení $T(i+1)$ od tvrzení $T(i)$. Tento „rozdíl“ budete muset zdůvodnit.
- Pozor, občas je potřeba „zesílit“ tvrzení $T(n)$, aby indukční krok fungoval.
- Často se chybuje v důkazu indukčního kroku, neboť ten bývá většinou výrazně obtížnější než báze,
  ale o to zrádnější jsou chyby v samotné zdánlivě snadné bázi(!)
  - Dejte si pozor, od které hodnoty $n \geq k_0$ je indukční krok univerzálně platný...

Příklad 11. Věta: *Pro každé* $n \geq 1$ *platí*

$$s(n) = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \ldots + \frac{1}{n(n+1)} < 1\,.$$

Důkaz. Báze indukce je zřejmá, $\frac{1}{1 \cdot 2} < 1$.

Co však indukční krok? Předpoklad $s(n) < 1$ je sám „příliš slabý“ na to, aby bylo možno tvrdit $s(n+1) = s(n) + \frac{1}{(n+1)(n+2)} < 1$.

Neznamená to ještě, že by tvrzení nebylo platné, jen je potřeba náš indukční předpoklad zesílit. Budeme dokazovat

„Pro každé přirozené $n \geq 1$ *platí* $s(n) \leq 1 - \frac{1}{n+1} < 1$ *.“*

To platí pro $n = 1$ a dále algebraickou úpravou dokončíme zesílený indukční krok:

$$\begin{aligned} s(n+1) \;&=\; s(n) + \frac{1}{(n+1)(n+2)} \;\leq\; 1 - \frac{1}{n+1} + \frac{1}{(n+1)(n+2)} = \\ &= 1 + \frac{-(n+2)+1}{(n+1)(n+2)} \;=\; 1 - \frac{1}{n+2} \end{aligned}$$

□

Příklad 12. Věta:(„nevěta“)
V každém stádu o $n \geq 1$ *koních mají všichni koně stejnou barvu.*

Důkaz. Indukcí vzhledem k n.

- Báze: Ve stádu o jednom koni mají všichni koně stejnou barvu.
- Indukční krok: Nechť $S = \{K_1, \cdots, K_{i+1}\}$ je stádo o $i+1$ koních. Dokážeme, že všichni koně mají stejnou barvu. Uvažme dvě menší stáda:
  - $S' = \{K_1, \cdots, K_i\}$
  - $S'' = \{K_2, \cdots, K_{i+1}\}$

  Podle indukčního předpokladu mají všichni koně ve stádu S' stejnou barvu B'. Podobně všichni koně ve stádu S'' mají podle indukčního předpokladu stejnou barvu B''. Dokážeme, že $B' = B''$, tedy že všichni koně ve stádu S mají stejnou barvu. To ale plyne z toho, že koně $K_2, \cdots, K_i$ patří jak do stáda S', tak i do stáda S''. □

(Ale to už je podvod! Vidíte, kde?)

Věty typu „tehdy a jen tehdy“

- Jde o věty tvaru

  > „Nechť platí *předpoklady P*. Pak *tvrzení A* platí tehdy a jen tehdy, platí-li *tvrzení B*.“

- Příklady jiných formulací téže věty:
  - Za *předpokladů P* je *tvrzení B* nutnou a postačující podmínkou pro platnost *tvrzení A*.
  - Za *předpokladů P* je *tvrzení A* nutnou a postačující podmínkou pro platnost *tvrzení B*.
  - Nechť platí *předpoklady P*. Pak *tvrzení A* platí právě když platí *tvrzení B*.
- Důkaz vět tohoto tvaru má vždy dvě části. Je třeba dokázat:
  - Jestliže platí *předpoklady P* a *tvrzení A*, pak platí *tvrzení B*.
  - Jestliže platí *předpoklady P* a *tvrzení B*, pak platí *tvrzení A*.

Pojem množiny

- Co je *množina*?
  - Naivní teorie množin: „Množina je soubor prvků a je svými prvky plně určena.“
  - Množiny mohou být prvky jiných množin (!)
  - $\emptyset, \quad \{a, b\}, \quad \{b, a\}, \quad \{a, b, a\}, \quad \{\{a, b\}\}, \quad \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$
  - $\{x \mid x$ je liché přirozené číslo$\}$
- Počet prvků (*mohutnost*) množiny $|A|$.
  - $|\emptyset| = 0, \quad |\{\emptyset\}| = 1, \quad |\{a, b, c\}| = 3, \quad |\{\{a, b\}, c\}| = 2$
- Množiny jsou „stejné“ právě když mají stejné prvky.
  - $x \in M$ „x je *prvkem* množiny M“
  - $a \in \{a, b\}, \quad a \notin \{\{a, b\}\}, \quad \{a, b\} \in \{\{a, b\}\}, \quad \emptyset \in \{\emptyset\}, \quad \emptyset \notin \emptyset$
  - $\{a, b\} = \{b, a\} = \{a, b, a\}, \quad \{a, b\} \neq \{\{a, b\}\}$

- Množina A je *podmnožinou* množiny B, právě když každý prvek A je prvkem B. Píšeme $A \subseteq B$ nebo také $B \supseteq A$; říkáme také, že se jedná o *inkluzi*.
  - $\{a\} \subseteq \{a\} \subseteq \{a, b\} \not\subseteq \{\{a, b\}\}, \quad \emptyset \subseteq \{\emptyset\}$
  - $A \subset B$ právě když $A \subseteq B$ a $A \neq B$ (A je *vlastní* podmnožinou B)
- Podle definice jsou množiny A a B stejné, mají-li stejné prvky.
  - Platí tedy $A = B$ právě když $A \subseteq B$ a $B \subseteq A$.
  - Důkaz toho, že $A = B$, má obvykle dvě části. Odděleně se dokáží inkluze $A \subseteq B$ a $B \subseteq A$.

Množinové operace

- Sjednocení

$$A \cup B = \{x \mid x \in A \text{ nebo } x \in B\}$$

  * $\{a, b, c\} \cup \{a, d\} = \{a, b, c, d\}$
  * $\bigcup_{i \in I} A_i = \{x \mid x \in A_i \text{ pro nějaké } i \in I\}$
  * Nechť $A_i = \{2.i\}$ pro každé $i \in \mathbb{N}_0$. Pak $\bigcup_{i \in \mathbb{N}_0} A_i$ je množina všech sudých přirozených čísel.

- Průnik

$$A \cap B = \{x \mid x \in A \text{ a současně } x \in B\}$$

  * $\{a, b, c\} \cap \{a, d\} = \{a\}$
  * $\bigcap_{i \in I} A_i = \{x \mid x \in A_i \text{ pro každé } i \in I\}$
  * Nechť $A_i = \{x \mid x \in \mathbb{N}, x \geq i\}$ pro každé $i \in \mathbb{N}$. Pak $\bigcap_{i \in \mathbb{N}} A_i = \emptyset$.

- Rozdíl

$$A \setminus B = \{x \mid x \in A \text{ a současně } x \notin B\}$$

  * $\{a, b, c\} \setminus \{a, b, d\} = \{c\}$
  * .....

- Doplněk

  Nechť $A \subseteq M$. *Doplněk* A *vzhledem k* M je množina $\overline{A} = M \setminus A$.

  * Jedná se o poněkud specifickou operaci, která musí být vztažena vzhledem k *nosné množině* M !
  * Je-li $M = \{a, b, c\}$, pak $\overline{\{a, b\}} = \{c\}$.
  * Je-li $M = \{a, b\}$, pak $\overline{\{a, b\}} = \emptyset$.

- Kartézský součin

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

  * kde uspořádaná dvojice (a, b) je množina $\{\{a\}, \{a, b\}\}$.
  * Platí $(a, b) = (c, d)$ právě když $a = c$ a současně $b = d$.
  * Mnemotechnická pomůcka

$$|A \times B| = |A| \cdot |B| .$$

  * $\{a, b\} \times \{a\} = \{(a, a), (b, a)\}$.
  * $\emptyset \times M = \emptyset$ pro každou množinu M.
  * Co je podle definice (a, a)?
    $(a, a) = \{\{a\}, \{a, a\}\} = \{\{a\}, \{a\}\} = \{\{a\}\}$

- Skládání součinu
  - Pro každé $k \in \mathbb{N}$ definujeme uspořádanou k-tici $(a_1, \cdots, a_k)$ induktivně takto:
    - $(a_1) = a_1$
    - $(a_1, \cdots, a_i, a_{i+1}) = ((a_1, \cdots, a_i), a_{i+1})$
  - Platí $(a_1, \cdots, a_k) = (b_1, \cdots, b_k)$ právě když $a_i = b_i$ pro každé $i \in \mathbb{N}$ kde $1 \leq i \leq k$.
  - Pro každé $k \in \mathbb{N}$ definujeme
    $A_1 \times \cdots \times A_k = \{(a_1, \cdots, a_k) \mid a_i \in A_i \text{ pro každé } 1 \leq i \leq k\}$.
  - Podle uvedené definice není součin asociativní, tj. obecně nemusí platit, že $A \times (B \times C) = (A \times B) \times C$.
  - V matematické praxi je někdy výhodnější uvažovat „upravenou“ definici, podle níž součin asociativní je. Pro účely této přednášky není podstatné, k jaké definici se přikloníme. Prezentované definice a věty „fungují“ pro obě varianty.
  - $\mathbb{N}^3 = \mathbb{N} \times \mathbb{N} \times \mathbb{N} = \{(i, j, k) \mid i, j, k \in \mathbb{N}\}$.
  - Co je $\mathbb{N}^0$? $\{\emptyset\}$

- Potenční množina (množina všech podmnožin)

$$2^A = \{B \mid B \subseteq A\}$$

  * Platí $|2^A| = 2^{|A|}$.
  * $2^{\{a,b\}} = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$
  * $2^{\emptyset} = \{\emptyset\}$
  * $2^{\{\emptyset,\{\emptyset\}\}} = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$
  * $2^{\{a\}\times\{a,b\}} = \{\emptyset, \{(a, a)\}, \{(a, b)\}, \{(a, a), (a, b)\}\}$

- Charakteristický vektor (pod)množiny
  * Používaný v případech, kdy všechny uvažované množiny jsou podmnožinami nějaké *nosné* množiny X.
  * Pro $X = \{x_1, x_2, \ldots, x_n\}$ a $A \subseteq X$ definujeme charakteristický vektor χ_A jako

$$\chi_A = (c_1, c_2, \ldots, c_n), \text{ kde } c_i = 1 \text{ pro } x_i \in A \text{ a } c_i = 0 \text{ jinak.}$$

Důkaz rovnosti množin

Příklad 13. Věta: *Pro každé tři množiny* A, B, C *platí*

$$A \setminus (B \cap C) = (A \setminus B) \cup (A \setminus C).$$

Důkaz.

- $A \setminus (B \cap C) \subseteq (A \setminus B) \cup (A \setminus C)$:
  - Je-li $x \in A \setminus (B \cap C)$, pak $x \in A$ a zároveň $x \notin (B \cap C)$, neboli $x \notin B$ nebo $x \notin C$.
  - Pro první možnost máme $x \in (A \setminus B)$, pro druhou $x \in (A \setminus C)$.
- $A \setminus (B \cap C) \supseteq (A \setminus B) \cup (A \setminus C)$:
  - Je-li $x \in (A \setminus B) \cup (A \setminus C)$, pak $x \in (A \setminus B)$ nebo $x \in (A \setminus C)$.
  - Pro první možnost máme $x \in A$ a zároveň $x \notin B$, z čehož plyne $x \in A$ a zároveň $x \notin (B \cap C)$, a tudíž $x \in A \setminus (B \cap C)$.
  - Druhá možnost je analogická.

□

Relace mezi (nad) množinami

- Nechť $k \in \mathbb{N}$.
  Relace mezi množinami $A_1, \cdots, A_k$ je podmnožina součinu $A_1 \times \cdots \times A_k$.
  Pokud $A_1 = \cdots = A_k = A$, hovoříme o k-ární relaci na A.

- Příklady
  - $\{(1, a), (2, a)\}$ je relace mezi $\{1, 2, 3\}$ a $\{a, b\}$
  - $\{(i, 2.i) \mid i \in \mathbb{N}\}$ je binární relace na $\mathbb{N}$.
  - $\{(i, j, i + j) \mid i, j \in \mathbb{N}\}$ je ternární relace na $\mathbb{N}$.
  - $\{3.i \mid i \in \mathbb{N}\}$ je unární relace na $\mathbb{N}$.

- $2^{A_1 \times \cdots \times A_k}$ je tedy množina všech relací mezi $A_1, \cdots, A_k$.

Reprezentace konečných relací pomocí tabulek

- Definujme následující množiny („elementární typy“)
  - ZNAK $= \{a, \cdots, z, A, \cdots, Z, mezera\}$
  - CISLICE $= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- Dále definujeme tyto množiny („odvozené typy“)
  - JMENO $=$ ZNAK^{15}, PRIJMENI $=$ ZNAK^{20},
  - VEK $=$ CISLICE^3,
  - ZAMESTNANEC $=$ JMENO $\times$ PRIJMENI $\times$ VEK.
- Relaci „typu“ ZAMESTNANEC pak lze reprezentovat tabulkou:

JMENO	PRIJMENI	VEK
Jan	Novák	42
Petr	Vichr	28
Pavel	Zíma	26

- Relační databáze je konečná množina tabulek. Schéma databáze je (zjednodušeně řečeno) množina „typů“ jednotlivých tabulek.

Funkce mezi množinami

- (Totální) funkce z množiny A do množiny B je relace f mezi A a B taková, že pro každé $x \in A$ existuje právě jedno $y \in B$ takové, že $(x, y) \in f$.

  „*Neformálně řečeno, ve funkci* f *je každé vstupní hodnotě* x *přiřazena jednoznačně výstupní hodnota* y*.“*
  (V obecné relaci počty „přiřazených“ dvojic neomezujeme. . .)

- Místo $(x, y) \in f$ píšeme obvykle $f(x) = y$.

- Množina A se nazývá *definiční obor* a množina B *obor hodnot* funkce f.

  Zápis $f : A \to B$ říká, že f je funkce s definičním oborem A a oborem hodnot B.

- Parciální funkce. Pokud naší definici funkce upravíme tak, že požadujeme pro každé $x \in A$ nejvýše jedno $y \in B$ takové, že $(x, y) \in f$, obdržíme definici *parciální funkce* z A do B.

  V parciální funkci p *nemusí být pro některé „vstupní“ hodnoty* x *funkční hodnota definována.*
  Pro nedefinovanou hodnotu používáme znak $\perp$.

Příklady funkcí

- Definujeme funkci $f : \mathbb{N} \to \mathbb{N}$ předpisem $f(x) = x + 8$. Tj. $f = \{(x, x + 8) \mid x \in \mathbb{N}\}$.
- Definujeme funkci $plus : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ předpisem $plus(i, j) = i + j$. Tj. $plus = \{(i, j, i + j) \mid i, j \in \mathbb{N}_0\}$.
- Definujeme parciální funkci $f : \mathbb{Z} \to \mathbb{N}$ předpisem

$$f(x) = \begin{cases} 3 + x & \text{jestliže } x \geq 0, \\ \bot & \text{jinak.} \end{cases}$$

  Tj. $f = \{(x, 3 + x) \mid x \in \mathbb{N}_0\}$.
- Také funkce $f : \mathbb{R} \to \mathbb{R}$ daná běžným analytickým předpisem

$$f(x) = \sqrt{x}$$

  je jen parciální – není definována pro $x < 0$.
- Co je relace, přiřazující lidem v ČR jejich rodná čísla?

Funkcím se také říká zobrazení.

Vlastnosti funkcí

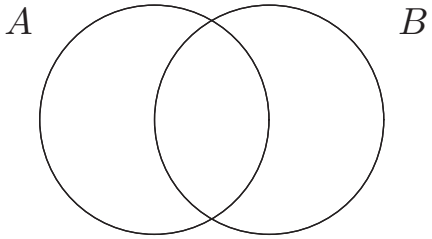
- Funkce $f : A \to B$ je
  - *injektivní* (nebo také prostá) právě když pro každé $x, y \in A$, $x \neq y$ platí, že $f(x) \neq f(y)$;
  - *surjektivní* (nebo také „na“) právě když pro každé $y \in B$ existuje $x \in A$ takové, že $f(x) = y$;
  - *bijektivní* (vzáj. jednoznačná) právě když je injektivní a současně surjektivní.
- Příklady:
  - Funkce $plus : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ je surjektivní, ale není prostá.
  - Funkce $g : \mathbb{Z} \to \mathbb{N}_0$ daná předpisem

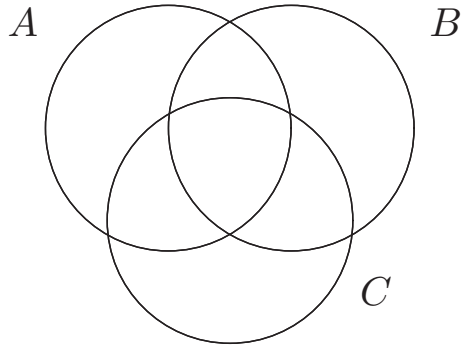
$$g(x) = \begin{cases} -2x - 1 & \text{jestliže } x < 0, \\ 2x & \text{jinak} \end{cases}$$

    je bijektivní.
  - Funkce $\emptyset : \emptyset \to \emptyset$ je bijektivní.
  - Funkce $\emptyset : \emptyset \to \{a, b\}$ je injektivní, ale není surjektivní.

Princip inkluze a exkluze

Někdy také nazýván „princip *zapojení a vypojení*“. . .





Věta 1. *Počet prvků ve sjednocení dvou či tří množin spočítáme:*

$$|A \cup B| = |A| + |B| - |A \cap B|$$

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

- Z 1000 televizí jich při první kontrole na výrobní lince má 5 vadnou obrazovku, 10 je poškrábaných a 12 má jinou závadu. Přitom 3 televize mají současně všechny tři vady a 4 jiné jsou poškrábané a mají jinou vadu.
  Kolik televizí je celkem vadných? 17

O mohutnosti a nekonečných množinách

- Množiny A je „nejvýše tak velká“ jako množina B, právě když existuje injektivní funkce $f : A \to B$.
- Množiny A a B jsou „stejně velké“ právě když mezi nimi existuje bijekce.
- Tyto definice „fungují“ i pro nekonečné množiny. Např. $\mathbb{N}$ a $\mathbb{Z}$ jsou „stejně velké“ (tzv. *spočetně nekonečné*).
- Lze snadno ukázat, že i $\mathbb{Q}$ je spočetně nekonečná, tj. existuje bijekce $f : \mathbb{N} \to \mathbb{Q}$.
- Existují ale i nekonečné množiny, které jsou „striktně větší“ než libovolná spočetná množina (příkladem je $\mathbb{R}$).
- Dokážeme, že „existuje nekonečná posloupnost nekonečných množin, z nichž každá je striktně větší než všechny předchozí“.

Věta 2. *Neexistuje žádné surjektivní (tudíž ani bijektivní) zobrazení* $\mathrm{g} : \mathbb{N} \to \mathbb{R}$.

Neformálně řečeno, reálných čísel je striktně více než přirozených.

Důkaz. Dokazujeme sporem. Nechť takové g existuje a pro zjednodušení se omezme jen na funkční hodnoty v intervalu $(0,1)$. Podle hodnot zobrazení g si takto můžeme „uspořádat“ dekadické zápisy všech reálných čísel v intervalu $(0,1)$ po řádcích do tabulky:

$$\begin{array}{lllllllllllll}
\mathrm{g}(0) = & 0. & 1 & 5 & 4 & 2 & 7 & 5 & 7 & 8 & 3 & 2 & 5 & \ldots \\
\mathrm{g}(1) = & 0. & & 2 & & & & & & & & & & \ldots \\
\mathrm{g}(2) = & 0. & & & 1 & & & & & & & & & \ldots \\
\mathrm{g}(3) = & 0. & & & & 3 & & & & & & & & \ldots \\
\mathrm{g}(4) = & 0. & & & & & 9 & & & & & & & \ldots \\
\vdots & \vdots & & & & & & \ddots & & & & & &
\end{array}$$

Nyní sestrojíme číslo $\alpha \in (0,1)$ následovně; jeho i-tá číslice za desetinnou čárkou bude 1, pokud v i-tém řádku tabulky na diagonále není 1, jinak to bude 2. V našem příkladě $\alpha = 0.21211\ldots$

Kde se naše číslo α v tabulce nachází? (Nezapomeňme, g byla surjektivní, takže tam α musí být!) Kostrukce však ukazuje, že α se od každého čísla v tabulce liší na aspoň jednom desetinném místě, to je spor.
(Až na drobný technický detail s rozvojem $\ldots\bar{9}$.) □

V obecnosti lze dokonce podobným způsobem dokázat následovné.

Věta 3. *Buď M libovolná množina. Pak existuje injektivní zobrazení $f : M \to 2^M$, ale neexistuje žádné bijektivní zobrazení $g : M \to 2^M$.*

Důkaz. Dokážeme nejprve existenci f. Stačí ale položit $f(x) = \{x\}$ pro každé $x \in M$. Pak $f : M \to 2^M$ je zjevně injektivní.

Neexistenci g dokážeme sporem. Předpokládejme tedy naopak, že existuje bijekce $g : M \to 2^M$. Uvažme množinu $K \subseteq M$ definovanou takto:

$$K = \{x \in M \mid x \notin g(x)\}.$$

Jelikož g je bijektivní a $K \in 2^M$, musí existovat $x \in M$ takové, že $g(x) = K$. Nyní rozlišíme dvě možnosti:

- $x \in g(x)$. Tj. $x \in K$. Pak ale $x \notin g(x)$ z definice K, spor.
- $x \notin g(x)$. To podle definice K znamená, že $x \in K$, tj. $x \in g(x)$, spor. □

Poznámky.

- Z toho, že nekonečna mohou být „různě velká“, lze lehce odvodit řadu dalších faktů. V jistém smyslu je např. množina všech „problémů“ větší než množina všech algoritmů (obě množiny jsou nekonečné), proto nutně existují problémy, které nejsou algoritmicky řešitelné. Musíme však být opatrní. . .

- Technika použitá v důkazech Vět 2 a 3 se nazývá *Cantorova diagonální metoda*, nebo také zkráceně diagonalizace.
  Konstrukci množiny K lze znázornit pomocí následující tabulky:

	a	b	c	d	$\cdots$
$\mathrm{g}(\mathrm{a})$	$\surd$	$-$	$-$	$\surd$	$\cdots$
$\mathrm{g}(\mathrm{b})$	$\surd$	$-$	$-$	$\surd$	$\cdots$
$\mathrm{g}(\mathrm{c})$	$-$	$\surd$	$-$	$\surd$	$\cdots$
$\mathrm{g}(\mathrm{d})$	$-$	$-$	$\surd$	$\surd$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

  Symbol $\surd$ resp. $-$ říká, že prvek uvedený v záhlaví sloupce patří resp. nepatří do množiny uvedené v záhlaví řádku. Tedy např. $\mathrm{d} \in \mathrm{g}(\mathrm{b})$ a $\mathrm{a} \notin \mathrm{g}(\mathrm{d})$.

„Naivní“ množinové paradoxy

- Uvážíme-li nyní nekonečnou posloupnost množin

  $$A_1, A_2, \cdots$$

  kde $A_1 = \mathbb{N}$ a $A_{i+1} = 2^{A_i}$ pro každé $i \in \mathbb{N}$, je vidět, že všechny množiny jsou nekonečné a každá je „striktně větší“ než libovolná předchozí.

- Kde však v tomto řazení mohutností bude „množina všech množin“?
  - Toto byl první Cantorův paradox nově vznikající teorie množin.
  - Brzy se však ukázalo, že je ještě mnohem hůř. . .

- Russelův paradox:

  Není pravda, že každý soubor prvků lze považovat za množinu.

  * $X = \{M \mid M$ je množina taková, že $M \notin M\}$
  * Platí $X \in X$?
  * Ano. Tj. $X \in X$. Pak ale X splňuje $X \notin X$.
  * Ne. Pak X splňuje vlastnost $X \notin X$, tedy X je prvkem X, tj., $X \in X$.
  * Obě možné odpovědi vedou ke sporu. X tedy nelze prohlásit za množinu.

  Vidíte zde podobnost přístupu s Cantorovou diagonalizací?

- Pro ilustraci, znáte toho „*holiče v malém městečku, který holí právě ty muže městečka, kteří se sami neholí*“?

- Tyto paradoxy naivní teorie množin zatím v tomto kurzu nerozřešíme, ale zapamatujeme si, že většina matematických a informatických disciplín vystačí s „intuitivně bezpečnými“ množinami.

Algoritmická neřešitelnost problému zastavení

- Program v každém programovacím jazyce je konečná posloupnost složená z konečně mnoha symbolů (písmena, číslice, mezery, speciální znaky, apod.) Nechť Σ je množina všech těchto symbolů. Množina všech programů je tedy jistě podmnožinou množiny $\bigcup_{i \in \mathbb{N}} \Sigma^i$, která je spočetně nekonečná. Existuje tedy bijekce f mezi množinou $\mathbb{N}$ a množinou všech programů. Pro každé $i \in \mathbb{N}$ označme symbolem P_i program $f(i)$. Pro každý program P tedy existuje $j \in \mathbb{N}$ takové, že $P = P_j$.
- Každý možný vstup každého možného programu lze zapsat jako konečnou posloupnost symbolů z konečné mnočiny Γ. Množina všech možných vstupů je tedy spočetně nekonečná a existuje tedy bijekce g mezi množinou $\mathbb{N}$ a množinou všech vstupů. Pro každé $i \in \mathbb{N}$ označme symbolem V_i vstup $g(i)$.
- Předpokládejme, že existuje program *Halt*, který pro dané $i, j \in \mathbb{N}$ zastaví s výstupem *ano*/*ne* podle toho, zda P_i pro vstup V_j zastaví, nebo ne.

- Uvažme program $Diag$ s následujícím kódem:

  input(k);
  if $(Halt(\mathrm{k},\mathrm{k}) == ano)$ then while true do skip

  Program $Diag(\mathrm{k})$ má na rozdíl od $Halt$ jen jeden vstup k, což bude důležité.

- Fungování programu $Diag$ lze znázornit pomocí následující tabulky:

	P_1	P_2	P_3	P_4	$\cdots$
V_1	$\surd$	$-$	$-$	$\surd$	$\cdots$
V_2	$\surd$	$-$	$-$	$\surd$	$\cdots$
V_3	$-$	$\surd$	$-$	$\surd$	$\cdots$
V_4	$-$	$-$	$\surd$	$\surd$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

  Symbol $\surd$ resp. $-$ říká, že program uvedený v záhlaví sloupce zastaví resp. nezastaví pro vstup uvedený v záhlaví řádku. Program $Diag$ „obrací“ diagonálu této tabulky.

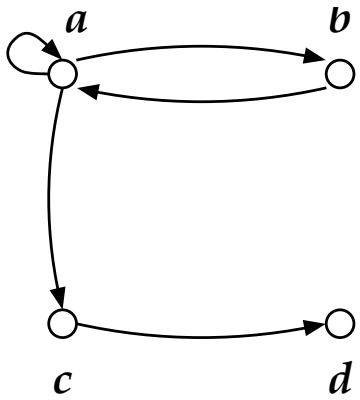
- Podle dříve uvedeného pozorování existuje $j \in \mathbb{N}$ takové, že $Diag = P_j$. Zastaví $Diag$ pro vstup V_j?
  - Ano. Podle kódu $Diag$ pak ale tento program vstoupí do nekonečné smyčky, tedy nezastaví.
  - Ne. Podle kódu $Diag$ pak ale if test neuspěje, a tento program tedy zastaví.

  Předpoklad existence programu $Halt$ tedy vede ke sporu. □

- Otázkami algoritmické (ne)řešitelnosti problémů se zabývá teorie vyčíslitelnosti.
- Metoda diagonalizace se také často využívá v teorii složitosti k důkazu toho, že dané dvě složitostní třídy jsou různé.

Reprezentace binárních relací na množině

- Danou binární relaci $\mathrm{R} \subseteq \mathrm{M} \times \mathrm{M}$ na M lze znázornit jejím grafem:
  - Prvky M znázorníme jako body v rovině.
  - Prvek $(\mathrm{a}, \mathrm{b}) \in \mathrm{R}$ znázorníme jako orientovanou hranu („šipku“) z a do b. Je-li $\mathrm{a} = \mathrm{b}$, pak je touto hranou „smyčka“ na a.
- Pozor, nejedná se o „grafy funkcí“ známé z analýzy.
  Příklad: Nechť $\mathrm{M} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$ a $\mathrm{R} = \{(\mathrm{a}, \mathrm{a}), (\mathrm{a}, \mathrm{b}), (\mathrm{b}, \mathrm{a}), (\mathrm{a}, \mathrm{c}), (\mathrm{c}, \mathrm{d})\}$.



- V případě, že R je nekonečná nebo „velká“, může být reprezentace R jejím grafem nepraktická (záleží pak na míře „pravidelnosti“ R).

Inverzní relace a skládání relací

- Nechť $R \subseteq A \times B$ je binární relace mezi A a B. Inverzní relace k relaci R se značí R^{-1} a je definována takto:

  $$R^{-1} = \{(b, a) \mid (a, b) \in R\}$$

  R^{-1} je tedy relace mezi B a A.

- Nechť $R \subseteq A \times B$ a $S \subseteq B \times C$ jsou binární relace. Kompozice (složení) relací R a S je relace $S \circ R \subseteq A \times C$ (čteme „S po R“) definovaná takto:

  $$S \circ R = \{(a, c) \mid \text{existuje } b \in B \text{ takové, že } (a, b) \in R, (b, c) \in S\}$$

  * Příklad: Je-li
    - $A = \{a, b\}, \quad B = \{1, 2\}, \quad C = \{X\},$
    - $R = \{(a, 1), (b, 1), (b, 2)\}, \quad S = \{(1, X)\},$

    pak $S \circ R = \{(a, X), (b, X)\}$.

Příklady pro relace–funkce:

- Inverzí bijektivní funkce $f(x) = x + 1$ na $\mathbb{Z}$ je funkce
$$f^{-1}(x) = x - 1.$$
Inverzí prosté funkce $f(x) = e^x$ na $\mathbb{R}$ je parciální funkce
$$f^{-1}(x) = \ln x.$$

- Funkce $g(x) = x \bmod 3$ není prostá na $\mathbb{N}$, a proto její inverzí je jen relace
$$g^{-1} = \{(a, b) \mid a = b \bmod 3\}.$$
Konkrétně $g^{-1} = \{(0,0), (0,3), (0,6), \ldots, (1,1), (1,4), \ldots, (2,2), (2,5), \ldots\}$.

- Složením funkcí $f(x) = x + 1$ a $h(x) = x^2$ na $\mathbb{R}$ vznikne funkce
$$f \circ h\,(x) = f(h(x)) \;=\; x^2 + 1.$$
Pozor! Složením „naopak“ ale vznikne funkce
$$h \circ f\,(x) = h(f(x)) \;=\; (x + 1)^2.$$

Vlastnosti binárních relací na množině

Nechť $R \subseteq M \times M$. Relace R je

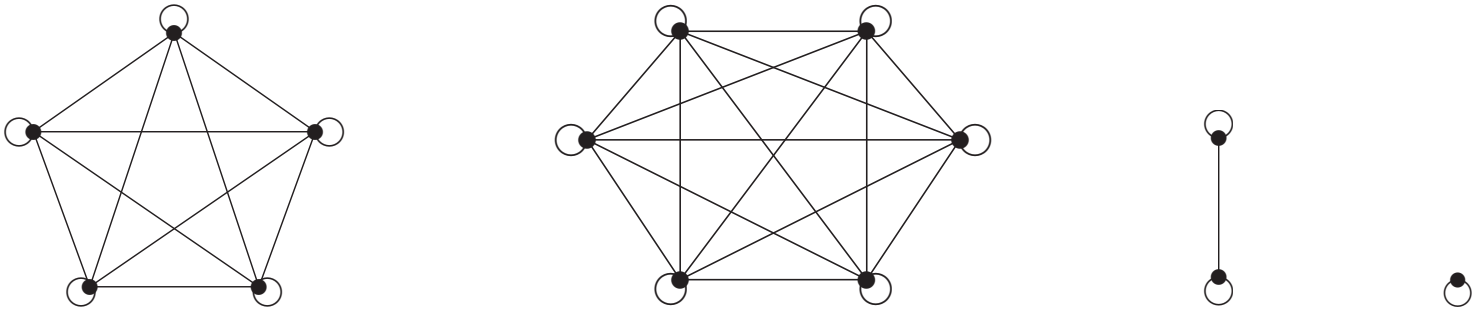
- *reflexivní*, právě když pro každé $a \in M$ platí $(a, a) \in R$;
- *symetrická*, právě když pro každé $a, b \in M$ platí, že jestliže $(a, b) \in R$, pak také $(b, a) \in R$;
- *antisymetrická*, právě když pro každé $a, b \in M$ platí, že jestliže $(a, b), (b, a) \in R$, pak $a = b$;
- *tranzitivní*, právě když pro každé $a, b, c \in M$ platí, že jestliže $(a, b), (b, c) \in R$, pak také $(a, c) \in R$;
- *ekvivalence*, právě když je reflexivní, symetrická a tranzitivní;
- *(částečné) uspořádání*, právě když je reflexivní, antisymetrická a tranzitivní.

Vlastnosti binárních relací na množině — příklady

- Buď M množina všech studentů 1. ročníku FI. Uvažme postupně relace $R \subseteq M \times M$ definované takto
  - $(x, y) \in R$ právě když x a y mají stejné rodné číslo;
  - $(x, y) \in R$ právě když x má stejnou výšku jako y;
  - $(x, y) \in R$ právě když výška x a y se neliší více jak o 2 cm;
  - $(x, y) \in R$ právě když x má alespoň takovou výšku jako y;
  - $(x, y) \in R$ právě když x má jinou výšku než y;
  - $(x, y) \in R$ právě když x je zamilován(a) do y.
- Buď $R \subseteq \mathbb{N} \times \mathbb{N}$ definovaná takto $(x, y) \in R$ právě když x dělí y.
- Buď $R \subseteq \mathbb{N}_0 \times \mathbb{N}_0$ definovaná takto $(x, y) \in R$ právě když x a y mají stejný zbytek po dělení číslem 5.
- Nechť $M = \{a, b\}$ a nechť $F = \{f \mid f : M \to M\}$. Buď $R \subseteq F \times F$ definovaná takto $(f, g) \in R$ právě když $f \circ g = g \circ f$. Jaké má tato relace vlastnosti?

Ekvivalence

- $R \subseteq M \times M$ je *ekvivalence* právě když R je reflexivní, symetrická a tranzitivní. Tyto tři vlastnosti je tedy třeba ověřit k důkazu toho, že daná relace R je ekvivalence.
- Jak vypadá graf ekvivalence?



- Neformálně řečeno: ekvivalence je relace $R \subseteq M \times M$, taková, že $(x, y) \in R$ právě když x a y jsou v nějakém smyslu „stejné“.

Příklady:

- Buď M množina všech studentů 1. ročníku FI. Uvažme postupně relace $R \subseteq M \times M$ definované takto
  - $(x, y) \in R$ právě když x má stejnou výšku jako y;
  - $(x, y) \in R$ právě když x má stejnou barvu vlasů jako y;
  - $(x, y) \in R$ právě když x, y mají stejnou výšku a stejnou barvu vlasů;
  - $(x, y) \in R$ právě když x, y mají buď stejnou výšku nebo stejnou barvu vlasů. (tato relace obecně není ekvivalence !)
- Buď $R \subseteq \mathbb{N}_0 \times \mathbb{N}_0$ definovaná takto: $(x, y) \in R$ právě když $|x - y|$ je dělitelné třemi. (V jakém smyslu jsou x a y „stejné“ ?)

Rozklady a jejich vztah k ekvivalencím

Definice 4. *Buď M množina. Rozklad (na) M je množina $\mathcal{N} \subseteq 2^M$ taková, že platí následující tři podmínky:*

- $\emptyset \notin \mathcal{N}$ *(tj. každý prvek $\mathcal{N}$ je neprázdná podmnožina M);*
- *pokud $A, B \in \mathcal{N}$, pak buď $A = B$ nebo $A \cap B = \emptyset$;*
- $\bigcup_{A \in \mathcal{N}} A = M$.

Prvkům $\mathcal{N}$ se také říká *třídy rozkladu*.

Příklady:

- Buď $M = \{a, b, c, d\}$. Pak $\mathcal{N} = \{\{a\}, \{b, c\}, \{d\}\}$ je rozklad na M.
- Nechť $A_0 = \{k \in \mathbb{N}_0 \mid k \bmod 3 = 0\}$, $A_1 = \{k \in \mathbb{N}_0 \mid k \bmod 3 = 1\}$, $A_2 = \{k \in \mathbb{N}_0 \mid k \bmod 3 = 2\}$.
  Pak $\mathcal{N} = \{A_0, A_1, A_2\}$ je rozklad na $\mathbb{N}_0$.

Každý rozklad $\mathcal{N}$ na M jednoznačně určuje jistou ekvivalenci $R_{\mathcal{N}}$ na M.

Věta 5. *Buď M množina a $\mathcal{N}$ rozklad na M. Nechť $R_{\mathcal{N}} \subseteq M \times M$ je relace na M definovaná takto*

$$(x, y) \in R_{\mathcal{N}} \text{ právě když existuje } A \in \mathcal{N} \text{ taková, že } x, y \in A.$$

Pak $R_{\mathcal{N}}$ je ekvivalence na M.

Důkaz. Dokážeme, že $R_{\mathcal{N}}$ je reflexivní, symetrická a tranzitivní.

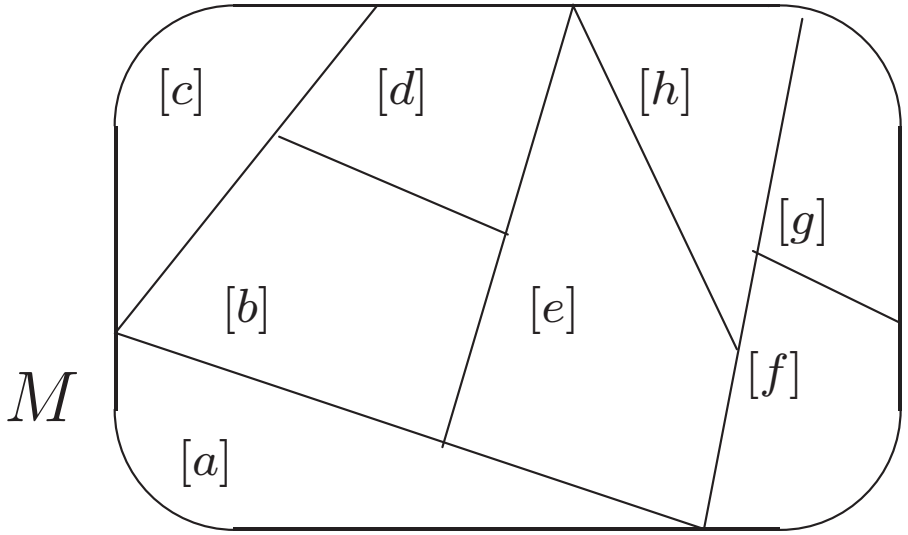
- Reflexivita: Buď $x \in M$ libovolné. Jelikož $\mathcal{N}$ je rozklad na M, musí existovat $A \in \mathcal{N}$ takové, že $x \in A$ (jinak spor se třetí podmínkou z Definice 4). Proto $(x, x) \in R_{\mathcal{N}}$, tedy $R_{\mathcal{N}}$ je reflexivní.
- Symetrie: Nechť $(x, y) \in R_{\mathcal{N}}$. Podle definice $R_{\mathcal{N}}$ pak existuje $A \in \mathcal{N}$ taková, že $x, y \in A$. To ale znamená, že také $(y, x) \in R_{\mathcal{N}}$ podle definice $R_{\mathcal{N}}$, tedy $R_{\mathcal{N}}$ je symetrická.
- Tranzitivita: Nechť $(x, y), (y, z) \in R_{\mathcal{N}}$. Podle definice $R_{\mathcal{N}}$ existují $A, B \in \mathcal{N}$ takové, že $x, y \in A$ a $y, z \in B$. Jelikož $A \cap B \neq \emptyset$, podle druhé podmínky z Definice 4 platí $A = B$. Tedy $x, z \in A = B$, proto $(x, z) \in R_{\mathcal{N}}$ podle definice $R_{\mathcal{N}}$. □

Každá ekvivalence R na M jednoznačně určuje jistý rozklad M/R na M.

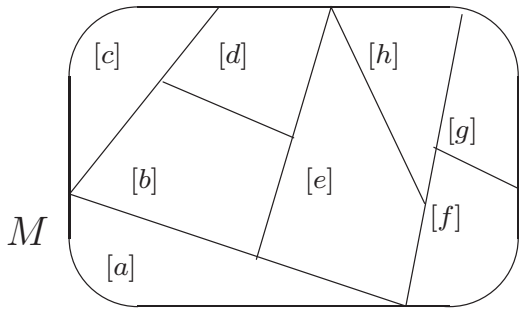
Věta 6. *Buď* M *množina a* R *ekvivalence na* M. *Pro každé* $x \in \mathrm{M}$ *definujeme množinu*

$$[x] = \{y \in \mathrm{M} \mid (x, y) \in \mathrm{R}\}.$$

Pak $\mathrm{M}/\mathrm{R} = \{[x] \mid x \in \mathrm{M}\}$ *je rozklad na* M.



Důkaz. Dokážeme, že M/R splňuje podmínky Definice 4.



- Pro každé $[x] \in M/R$ platí $[x] \neq \emptyset$, neboť $x \in [x]$.
- Nechť $[x], [y] \in M/R$. Ukážeme, že pokud $[x] \cap [y] \neq \emptyset$, pak $[x] = [y]$. Jestliže $[x] \cap [y] \neq \emptyset$, existuje $z \in M$ takové, že $z \in [x]$ a $z \in [y]$. Podle definice $[x]$ a $[y]$ to znamená, že $(x, z), (y, z) \in R$. Jelikož R je symetrická a $(y, z) \in R$, platí $(z, y) \in R$. Jelikož $(x, z), (z, y) \in R$ a R je tranzitivní, platí $(x, y) \in R$. Proto také $(y, x) \in R$ (opět ze symetrie R). Nyní dokážeme, že $[x] \subseteq [y]$ a $[y] \subseteq [x]$.
  - „$[x] \subseteq [y]$:“ Nechť $v \in [x]$. Pak $(x, v) \in R$ podle definice $[x]$. Dále $(y, x) \in R$ (viz výše), tedy $(y, v) \in R$ neboť R je tranzitivní. To podle definice $[y]$ znamená, že $v \in [y]$.
  - „$[y] \subseteq [x]$:“ Nechť $v \in [y]$. Pak $(y, v) \in R$ podle definice $[y]$. Dále $(x, y) \in R$ (viz výše), tedy $(x, v) \in R$ neboť R je tranzitivní. To podle definice $[x]$ znamená, že $v \in [x]$.
- Platí $\bigcup_{[x] \in M/R} [x] = M$, neboť $x \in [x]$ pro každé $x \in M$. □

Relace mezi (nad) množinami – zopakování

- Nechť $k \in \mathbb{N}$. Relace mezi množinami $A_1, \cdots, A_k$ je libovolná podmnožina kartézského součinu

$$A_1 \times \cdots \times A_k .$$

  Pokud $k = 2$ a $A_1 = A_2 = M$, hovoříme o *binární relaci na* M.

- Znovu krátce vlastnosti binárních relací
  - *reflexivní*, platí $(a, a) \in R$;
  - *symetrická*, jestliže $(a, b) \in R$, pak také $(b, a) \in R$;
  - *antisymetrická*, jestliže $(a, b), (b, a) \in R$, pak $a = b$;
  - *tranzitivní*, jestliže $(a, b), (b, c) \in R$, pak také $(a, c) \in R$.

Uspořádání a předuspořádání

- $R \subseteq M \times M$ je uspořádání právě když R je reflexivní, antisymetrická a tranzitivní.
- $R \subseteq M \times M$ je předuspořádání (také kvaziuspořádání, nebo polouspořádání) právě když R je reflexivní a tranzitivní.
- (Před)uspořádaná množina je dvojice $(M, \sqsubseteq)$, kde M je množina a $\sqsubseteq$ je (před)uspořádání na M.
- Neformálně řečeno: uspořádání je taková relace $R \subseteq M \times M$, kde $(x, y) \in R$ právě když x je v nějakém smyslu „menší nebo rovno“ než y.
  Mohou ovšem existovat taková $x, y \in M$, kde neplatí $(x, y) \in R$ ani $(y, x) \in R$ (pak říkáme, že x a y jsou nesrovnatelné).
- Uspořádání R na M je lineární (také úplné) pokud každé dva prvky M jsou vzhledem k R srovnatelné.
- Rozdíl mezi uspořádáním a předuspořádáním je (neformálně řečeno!) v tom, že u předuspořádání srovnáváme prvky podle takového kritéria, které není pro daný prvek jedinečné.

Uspořádání a předuspořádání – příklady

- Buď M množina všech studentů 1. ročníku FI. Uvažme postupně relace $R \subseteq M \times M$ definované takto
  - $(x, y) \in R$ právě když x a y mají stejné rodné číslo;
  - $(x, y) \in R$ právě když x má alespoň takovou výšku jako y;
  - $(x, y) \in R$ právě když y má alespoň takovou výšku jako x.
- $(\mathbb{N}_0, \leq)$ je lineárně uspořádaná množina (kde $\leq$ má „obvyklý“ význam).
- $(\mathbb{N}, |)$, kde $|$ je relace dělitelnosti, je uspořádaná množina. Toto uspořádání není lineární.
- Buď M množina. Pak $(2^M, \subseteq)$ je uspořádaná množina.
- Nechť M je množina a $(A, \leq_A)$ uspořádaná množina. Nechť $\mathcal{F} = \{f \mid f : M \to A\}$. Definujme binární relaci $\sqsubseteq$ na $\mathcal{F}$ předpisem

  $$f \sqsubseteq g \quad \text{právě když} \quad \text{pro každé } x \in M \text{ platí } f(x) \leq_A g(x).$$

  Pak $(\mathcal{F}, \sqsubseteq)$ je uspořádaná množina. Uspořádání $\sqsubseteq$ se nazývá „po bodech“.

- Nechť $(A, \leq_A)$ a $(B, \leq_B)$ jsou uspořádané množiny. Definujme binární relaci $\sqsubseteq$ na $A \times B$ předpisem

  $$(a, b) \sqsubseteq (a', b') \quad \text{právě když} \quad a \leq_A a' \text{ a } b \leq_B b'.$$

  Pak $(A \times B, \sqsubseteq)$ je uspořádaná množina.
  Toto uspořádání se nazývá „po složkách“.

- Nechť $(A, \leq_A)$ a $(B, \leq_B)$ jsou uspořádané množiny. Definujme binární relaci $\sqsubseteq$ na $A \times B$ předpisem

  $$(a, b) \sqsubseteq (a', b') \quad \text{právě když} \quad \text{buď } a \leq_A a' \text{ a } a \neq a', \text{ nebo } a = a' \text{ a } b \leq_B b'.$$

  Pak $(A \times B, \sqsubseteq)$ je uspořádaná množina. Navíc pokud $\leq_A$ i $\leq_B$ jsou lineární, je i $\sqsubseteq$ lineární.
  Toto uspořádání se nazývá „lexikografické“.

- Jsou-li $(A_1, \leq_1), \cdots, (A_n, \leq_n)$ uspořádané množiny, kde $n \geq 2$, pak množinu $A_1 \times \cdots \times A_n$ lze uspořádat po složkách nebo lexikograficky.

  $*$ Všimněte si, že lexikograficky se řadí slova ve slovníku. . .

- Je-li $\sqsubseteq$ předuspořádání na M, můžeme definovat relaci $\sim$ na M předpisem

  $$x \sim y \quad \text{právě když} \quad x \sqsubseteq y \text{ a } y \sqsubseteq x.$$

  Pak $\sim$ je ekvivalence na M, která se nazývá jádro předuspořádání $\sqsubseteq$.
  Na rozkladu $M/\sim$ pak lze zavést relaci $\preceq$ definovanou takto

  $$[x] \preceq [y] \quad \text{právě když } x \sqsubseteq y.$$

  Pak $(M/\sim, \preceq)$ je uspořádaná množina.

  $*$ Pro příklad si vezměme relaci dělitelnosti na $\mathbb{Z}$. Zde třeba $-2 \sim 2$.
  Jádrem tedy jsou dvojice čísel stejné absolutní hodnoty.

Další pojmy související s uspořádáním

Buď $(M, \sqsubseteq)$ uspořádaná množina.

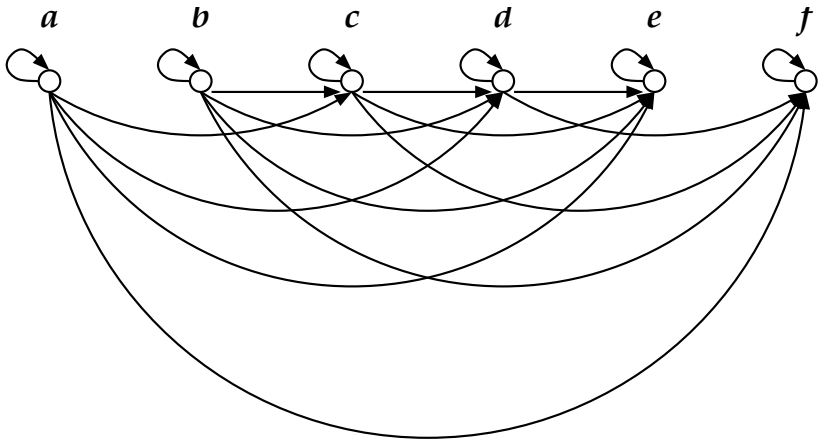
- $x \in M$ je minimální právě když pro každé $y \in M$ platí, že jestliže $y \sqsubseteq x$, pak $x \sqsubseteq y$.
  (Tj. x je minimální právě když neexistuje žádný prvek ostře menší než x.)
- $x \in M$ je maximální právě když pro každé $y \in M$ platí, že jestliže $x \sqsubseteq y$, pak $y \sqsubseteq x$.
  (Tj. x je maximální právě když neexistuje žádný prvek ostře větší než x.)
- $x \in M$ je nejmenší právě když pro každé $y \in M$ platí, že $x \sqsubseteq y$.
- $x \in M$ je největší právě když pro každé $y \in M$ platí, že $y \sqsubseteq x$.
- $x \in M$ pokrývá $y \in M$ právě když $x \neq y$, $y \sqsubseteq x$ a neexistuje žádné $z \in M$ takové, že $x \neq z \neq y$ a $y \sqsubseteq z \sqsubseteq x$.

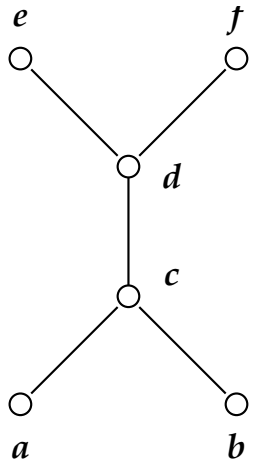
- $x \in M$ je horní závora (mez) množiny $A \subseteq M$ právě když $y \sqsubseteq x$ pro každé $y \in A$.
- $x \in M$ je dolní závora (mez) množiny $A \subseteq M$ právě když $x \sqsubseteq y$ pro každé $y \in A$.
- $x \in M$ je supremum množiny $A \subseteq M$, právě když x je nejmenší horní závora množiny A.
- $x \in M$ je infimum množiny $A \subseteq M$ právě když x je největší dolní závora množiny A.
- $A \subseteq M$ je řetězec v uspořádání $\sqsubseteq$ právě když $(A, \sqsubseteq)$ je lineárně uspořádaná množina.

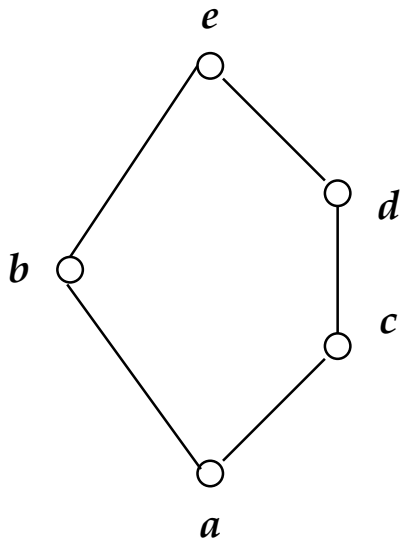
Pozor! Některé uvedené definice mají dosti „netriviální chování“ na nekonečných množinách. Proto je budeme obvykle uvažovat jen nad konečnými množinami. . .

Hasseovské diagramy

- Hasseovské diagramy uspořádaných množin jsou přehlednější než grafy relací.
- Hasseovský diagram konečné uspořádané množiny $(M, \sqsubseteq)$ vznikne takto
  - do první „horizontální vrstvy“ zakreslíme body odpovídající mininálním prvkům $(M, \sqsubseteq)$ (tj. které nepokrývají nic);
  - máme-li již zakreslenu „vrstvu“ i, pak do „vrstvy“ $i + 1$ (která je „nad“ vrstvou i) zakreslíme všechny nezakreslené prvky, které pokrývají pouze prvky „vrstev“ $\leq i$. Pokud prvek x „vrstvy“ $i + 1$ pokrývá prvek y „vrstvy“ $\leq i$, spojíme x a y neorientovanou hranou (tj. „čárou“).
- Příklad:



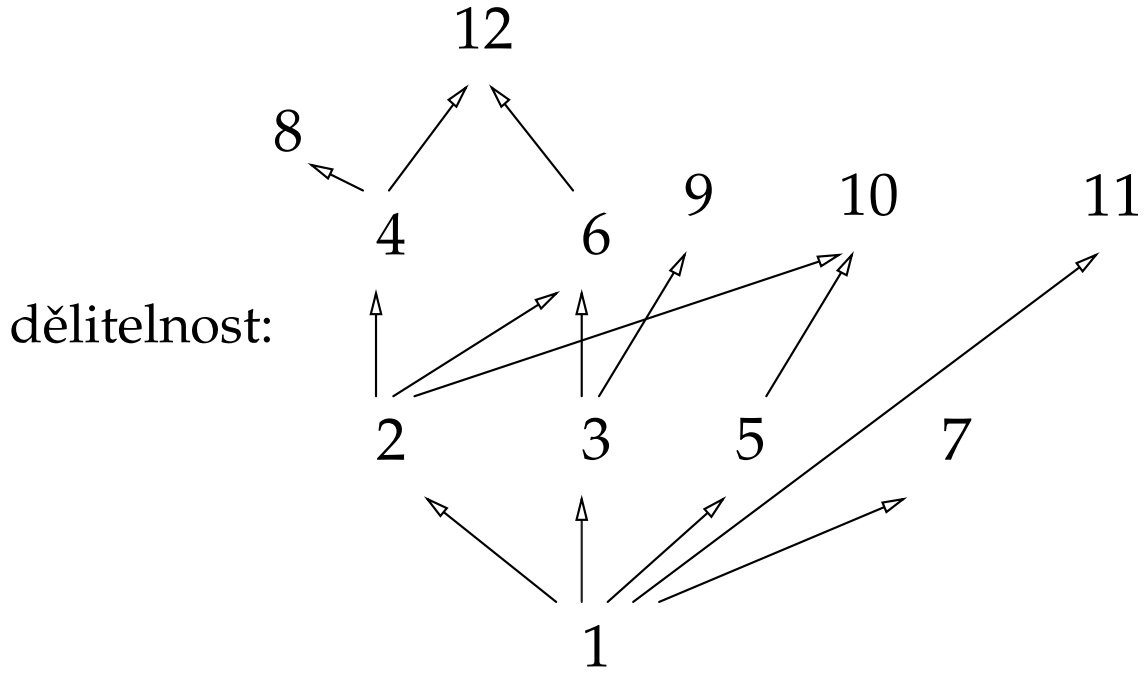




- Pozor! Původní popis Hasseova diagramu měl problém, který je vidět na obrázku:

  Ve skutečnosti se nelze odvolávat na prvky pokrávající jen některý prvek předchozí vrstvy, ale zařadit prvky do další vrstvy až tehdy, když všechny jím pokrývané prvky jsou v předchozích vrstvách.
  Také pojem *„vrstvy“* je jen velmi neformální, důležité je, že větší (pokrývající) prvky jsou nad menšími (pokrývanými).

- Další příklad



Uzávěry relací

Buď $\mathcal{V}$ (nějaká) vlastnost binárních relací. Řekneme, že $\mathcal{V}$ je vhodně definovaná, pokud splňuje následující podmínky:

- Pro každou množinu M a každou relaci $R \subseteq M \times M$ existuje alespoň jedna relace $S \subseteq M \times M$, která má vlastnost $\mathcal{V}$ a pro kterou platí $R \subseteq S$.
- Nechť I je množina a nechť $R_i \subseteq M \times M$ je relace mající vlastnost $\mathcal{V}$ pro každé $i \in I$. Pak relace $\bigcap_{i \in I} R_i$ má vlastnost $\mathcal{V}$.

Příklady:

- Reflexivita, symetrie, tranzitivita. Libovolná kombinace těchto tří vlastností je vhodně definovaná vlastnost.
- Antisymetrie není vhodně definovaná vlastnost.

Definice 7. *Nechť $\mathcal{V}$ je vhodně definovaná vlastnost binárních relací. Buď M množina a R binární relace na M. Pak existuje* nejmenší *(vzhledem k inkluzi!) relace obsahující R, která má vlastnost $\mathcal{V}$. Tuto relaci nazýváme* $\mathcal{V}$ uzávěr *relace* R.

Příklady: Buď R binární relace na M

- Reflexivní uzávěr R je přesně relace $R \cup \{(x,x) \mid x \in M\}$.
- Symetrický uzávěr R je přesně relace $\overset{\leftrightarrow}{R} = \{(x,y) \mid (x,y) \in R$ nebo $(y,x) \in R\}$.
- Buď $\mathcal{T}$ funkce, která pro každou binární relaci S vrátí relaci
$$\mathcal{T}(S) = S \cup \{(x,z) \mid \text{ existuje } y \text{ takové, že } (x,y),(y,z) \in S\}.$$
- Tranzitivní uzávěr R je přesně relace $R^+ = \bigcup_{i=1}^{\infty} \mathcal{T}^i(R)$, kde $\mathcal{T}^i = \underbrace{\mathcal{T} \circ \cdots \circ \mathcal{T}}_{i}$.
- Reflexivní a tranzitivní uzávěr R je přesně relace $R^* = \bigcup_{i=1}^{\infty} \mathcal{T}^i(Q)$, kde Q je reflexivní uzávěr R.
- Reflexivní, symetrický a tranzitivní uzávěr R (tj. nejmenší ekvivalence obsahující R) je přesně relace $(\overset{\leftrightarrow}{Q})^+$, kde Q je reflexivní uzávěr R.
- Buď $R \subseteq \mathbb{N} \times \mathbb{N}$ definovaná takto: $R = \{(i, i+1) \mid i \in \mathbb{N}\}$. Pak R^* je běžné $\leq$.

Výroky a jejich struktura v „přirozené“ podobě

- Výrok je tvrzení, o kterém má smysl prohlásit, že je buď pravdivé nebo nepravdivé.
  - Dnes v Brně pršelo.
  - $2 + 3 = 6$
  - To je bez problémů.
  - $x > 3$
  - Pro každé celé číslo x platí, že $x > 3$.
- Z jednoduchých výroků můžeme vytvářet výroky složitější pomocí logických spojek.
  - Kateřina přijede ve 12:00 a půjdeme spolu do kina.
  - Množina $\{a, b\}$ má více než jeden prvek a není nekonečná.
  - Jestliže má Karel přes 90 kilo váhy, nepojedu s ním výtahem.
  - Jestliže má kráva 10 nohou, mají všechny domy modrou střechu.

- Schopnost porozumět takovýmto větám je součást lidského způsobu uvažování a z tohoto hlediska nemá přímou souvislost s matematikou („přirozená logika“).
- Formální logika definuje jazyk matematiky a odstraňuje nejednoznačnosti přirozeného jazyka.

(Formální) výroková logika

Syntaxe:

- Buď $At = \{A, B, C, \ldots\}$ spočetně nekonečná množina *výrokových proměnných*.
- Množina *výrokových formulí* Φ je definována *induktivně* následujícími pravidly:
  (1) $At \subseteq \Phi$.
  (2) Jestliže $\varphi, \psi \in \Phi$, pak také $\neg(\varphi) \in \Phi$ a $(\varphi) \Rightarrow (\psi) \in \Phi$.
  (3) Každý prvek Φ vznikne konečně mnoha aplikacemi pravidel (1) a (2).
- Příklady:
  * $A, \quad (A) \Rightarrow (B), \quad ((A) \Rightarrow (\neg(B))) \Rightarrow ((\neg(B)) \Rightarrow (C))$
  * $A \Rightarrow B, \quad A \Rightarrow B \Rightarrow C, \quad \neg A \Rightarrow B$ (nejsou správně výrokové formule)
- Úmluva 1: Pro zvýšení čitelnosti budeme závorky vynechávat, pokud to nepovede k nejednoznačnostem za předpokladu, že negace $\neg$ má „vyšší prioritu“ než $\Rightarrow$. Touto úmluvou se nemění množina Φ; mění se způsob reprezentace jejích prvků.

- Úmluva 2:
  - $\varphi \vee \psi$ je jiný zápis formule $\neg\varphi \Rightarrow \psi$
  - $\varphi \wedge \psi$ je jiný zápis formule $\neg(\neg\varphi \vee \neg\psi)$
  - $\varphi \Leftrightarrow \psi$ je jiný zápis formule $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$

Sémantika:

- *Valuace* (ohodnocení) je funkce $\nu : At \to \{true, false\}$.
- Pro každou valuaci ν definujeme funkci $\mathcal{S}_\nu : \Phi \to \{true, false\}$ induktivně takto:
  - $\mathcal{S}_\nu(A) = \nu(A)$ pro každé $A \in At$
  - $\mathcal{S}_\nu(\neg\varphi) = \begin{cases} true & \text{jestliže } \mathcal{S}_\nu(\varphi) = false; \\ false & \text{jinak.} \end{cases}$
  - $\mathcal{S}_\nu(\varphi \Rightarrow \psi) = \begin{cases} false & \text{jestliže } \mathcal{S}_\nu(\varphi) = true \text{ a } \mathcal{S}_\nu(\psi) = false; \\ true & \text{jinak.} \end{cases}$

  Tento předpis podává nejen definici funkce $\mathcal{S}_\nu$, ale také návod na to, jak ji pro daný argument vypočítat.

- Důsledkem této definice je to, že
  - $\mathcal{S}_\nu(\varphi \vee \psi) = \mathit{true}$ právě když $\mathcal{S}_\nu(\varphi) = \mathit{true}$ nebo $\mathcal{S}_\nu(\psi) = \mathit{true}$;
  - $\mathcal{S}_\nu(\varphi \wedge \psi) = \mathit{true}$ právě když $\mathcal{S}_\nu(\varphi) = \mathit{true}$ a současně $\mathcal{S}_\nu(\psi) = \mathit{true}$;
  - $\mathcal{S}_\nu(\varphi \Leftrightarrow \psi) = \mathit{true}$ právě když platí jedna z následujících podmínek
    - $\mathcal{S}_\nu(\varphi) = \mathit{true}$ a současně $\mathcal{S}_\nu(\psi) = \mathit{true}$,
    - $\mathcal{S}_\nu(\varphi) = \mathit{false}$ a současně $\mathcal{S}_\nu(\psi) = \mathit{false}$.
- Formule $\varphi \in \Phi$ je *pravdivá* (také výroková tautologie), psáno $\vDash \varphi$, pokud pro každou valuaci ν platí, že $\mathcal{S}_\nu(\varphi) = \mathit{true}$.
  - $A \vee \neg A$
  - $\neg\neg A \Longleftrightarrow A$
  - $(A \wedge (A \Rightarrow B)) \Rightarrow B$
  - $(\neg B \Rightarrow \neg A) \Rightarrow (A \Rightarrow B)$
  - $(\neg A \Rightarrow (B \wedge \neg B)) \Rightarrow A$
- Řekneme, že formule $\varphi, \psi \in \Phi$ jsou *ekvivalentní*, právě když $\vDash \varphi \Leftrightarrow \psi$.

Důkazy indukcí ke struktuře formule

- Důkazová technika aplikovatelná na tvrzení typu „*Pro každou formuli* $\varphi \in \Phi$ *platí* $\mathsf{T}(\varphi)$“.
- Princip strukturální indukce říká, že k důkazu věty

  Pro každou formuli $\varphi \in \Phi$ *platí* $\mathsf{T}(\varphi)$

  stačí ověřit platnost těchto tří tvrzení:

  * Pro každé $A \in At$ platí $\mathsf{T}(A)$.
  * Jestliže platí $\mathsf{T}(\varphi)$, pak platí také $\mathsf{T}(\neg\varphi)$.
  * Jestliže platí $\mathsf{T}(\varphi)$ a $\mathsf{T}(\psi)$, pak platí také $\mathsf{T}(\varphi \Rightarrow \psi)$.

  (Dokazujeme tak vlastně indukcí podle „délky“ zápisu formule.)

Věta 8. *Buď* $\mathcal{F} : \Phi \to \Phi$ *funkce definovaná induktivně takto:*

- $\mathcal{F}(A) \quad = \quad \neg\neg A \qquad (A \in At)$
- $\mathcal{F}(\neg\varphi) \quad = \quad \neg\mathcal{F}(\varphi)$
- $\mathcal{F}(\varphi \Rightarrow \psi) \quad = \quad \neg\mathcal{F}(\psi) \Rightarrow \neg\mathcal{F}(\varphi)$

Pak pro každé $\theta \in \Phi$ *platí, že* θ *a* $\mathcal{F}(\theta)$ *jsou ekvivalentní.*

Důkaz. Indukcí ke struktuře θ.
($\equiv$ je „definiční rovnítko“ pro formule.)

- $\theta \equiv A$. Formule A a $\neg\neg A$ jsou ekvivalentní.
- $\theta \equiv \neg\varphi$. Potřebujeme dokázat, že $\neg\varphi$ a $\mathcal{F}(\neg\varphi)$ jsou ekvivaletní. Podle definice $\mathcal{F}$ platí $\mathcal{F}(\neg\varphi) = \neg\mathcal{F}(\varphi)$, stačí tedy ověřit ekvivalenci $\neg\varphi$ a $\neg\mathcal{F}(\varphi)$. Podle indukčního předpokladu jsou φ a $\mathcal{F}(\varphi)$ ekvivalentní, proto jsou ekvivalentní také $\neg\varphi$ a $\neg\mathcal{F}(\varphi)$.

- $\theta \equiv \varphi \Rightarrow \psi$. Musíme dokázat ekvivalenci formulí $\varphi \Rightarrow \psi$ a $\mathcal{F}(\varphi \Rightarrow \psi)$. Podle definice $\mathcal{F}$ to znamená dokázat, že pro každou valuaci ν platí

  $$S_\nu\big((\varphi \Rightarrow \psi) \Leftrightarrow (\neg\mathcal{F}(\psi) \Rightarrow \neg\mathcal{F}(\varphi))\big) \quad = \quad \mathit{true} \qquad (*)$$

  Můžeme přitom předpokládat, že φ je ekvivalentní $\mathcal{F}(\varphi)$, a že ψ je ekvivalentní $\mathcal{F}(\psi)$. Buď ν valuace. Rozlišíme tři možnosti:

  - $S_\nu(\varphi) = \mathit{true}$ a $S_\nu(\psi) = \mathit{false}$. Pak také $S_\nu(\mathcal{F}(\varphi)) = \mathit{true}$ a $S_\nu(\mathcal{F}(\psi)) = \mathit{false}$. Tedy $S_\nu(\varphi \Rightarrow \psi) = \mathit{false}$ a $S_\nu(\neg\mathcal{F}(\psi) \Rightarrow \neg\mathcal{F}(\varphi)) = \mathit{false}$, což znamená, že platí (*).
  - $S_\nu(\varphi) = \mathit{true}$ a $S_\nu(\psi) = \mathit{true}$. Pak také $S_\nu(\mathcal{F}(\varphi)) = \mathit{true}$ a $S_\nu(\mathcal{F}(\psi)) = \mathit{true}$. Tedy $S_\nu(\varphi \Rightarrow \psi) = \mathit{true}$ a $S_\nu(\neg\mathcal{F}(\psi) \Rightarrow \neg\mathcal{F}(\varphi)) = \mathit{true}$, což znamená, že platí (*).
  - $S_\nu(\varphi) = \mathit{false}$. Pak také $S_\nu(\mathcal{F}(\varphi)) = \mathit{false}$, Tedy $S_\nu(\varphi \Rightarrow \psi) = \mathit{true}$ a $S_\nu(\neg\mathcal{F}(\psi) \Rightarrow \neg\mathcal{F}(\varphi)) = \mathit{true}$, což znamená, že platí (*).

□

Co znamená „znegovat formuli“ ?

- Přesný význam formulí se zanořenými negacemi je někdy obtížné zjistit (podobně jako v běžné řeči).
- „Není pravda, že nemohu neříct, že není pravda, že tě nemám nerad.“
- Výrokové formule se proto obvykle prezentují v *normálním tvaru*, kde se negace vyskytují pouze u výrokových proměnných.
- Každou výrokou formuli lze převést do normálního tvaru, pokud povolíme užívání odvozených spojek $\wedge$ a $\vee$.
  - Pro ilustraci $\neg(A \Rightarrow B)$ je ekvivalentní $A \wedge \neg B$,
  - $\neg\big(C \wedge (\neg A \Rightarrow B)\big)$ je ekvivalentní $\neg C \vee (\neg A \wedge \neg B)$.
- „Znegováním formule“ se obvykle myslí převod $\neg\varphi$ do normálního tvaru.

Formální postup negace

- Induktivně definujeme funkce $\mathcal{F}$ a $\mathcal{G}$ předpisy

$$\begin{array}{lcl|lcl}
\mathcal{F}(A) & = & A & \mathcal{G}(A) & = & \neg A \\
\mathcal{F}(\neg\varphi) & = & \mathcal{G}(\varphi) & \mathcal{G}(\neg\varphi) & = & \mathcal{F}(\varphi) \\
\mathcal{F}(\varphi \wedge \psi) & = & \mathcal{F}(\varphi) \wedge \mathcal{F}(\psi) & \mathcal{G}(\varphi \wedge \psi) & = & \mathcal{G}(\varphi) \vee \mathcal{G}(\psi) \\
\mathcal{F}(\varphi \vee \psi) & = & \mathcal{F}(\varphi) \vee \mathcal{F}(\psi) & \mathcal{G}(\varphi \vee \psi) & = & \mathcal{G}(\varphi) \wedge \mathcal{G}(\psi) \\
\mathcal{F}(\varphi \Rightarrow \psi) & = & \mathcal{F}(\varphi) \Rightarrow \mathcal{F}(\psi) & \mathcal{G}(\varphi \Rightarrow \psi) & = & \mathcal{F}(\varphi) \wedge \mathcal{G}(\psi) \\
\mathcal{F}(\varphi \Leftrightarrow \psi) & = & \mathcal{F}(\varphi) \Leftrightarrow \mathcal{F}(\psi) & \mathcal{G}(\varphi \Leftrightarrow \psi) & = & (\mathcal{F}(\varphi) \wedge \mathcal{G}(\psi)) \vee (\mathcal{G}(\varphi) \wedge \mathcal{F}(\psi))
\end{array}$$

- Pro libovolnou formuli φ platí, že

  $\mathcal{F}(\varphi)$ je jí ekvivalentní formule v normálním tvaru

  a $\mathcal{G}(\varphi)$ je formule v normálním tvaru ekvivalentní negaci $\neg\varphi$.

- Uvedené formální předpisy takto vyjadřují „intuitivní postup negace“ v matematicky přesném tvaru.

Příklad:

- Uvažme formuli $\neg(A \Rightarrow \neg(B \vee \neg(B \Rightarrow \neg A)))$. Platí

$$\begin{array}{lcll}
\mathcal{F}(\neg(A \Rightarrow \neg(B \vee \neg(B \Rightarrow \neg A)))) & = & \mathcal{G}(A \Rightarrow \neg(B \vee \neg(B \Rightarrow \neg A))) & = \\
\mathcal{F}(A) \wedge \mathcal{G}(\neg(B \vee \neg(B \Rightarrow \neg A))) & = & A \wedge \mathcal{F}(B \vee \neg(B \Rightarrow \neg A)) & = \\
A \wedge (\mathcal{F}(B) \vee \mathcal{F}(\neg(B \Rightarrow \neg A))) & = & A \wedge (B \vee \mathcal{G}(B \Rightarrow \neg A)) & = \\
A \wedge (B \vee (\mathcal{F}(B) \wedge \mathcal{G}(\neg A))) & = & A \wedge (B \vee (B \wedge \mathcal{F}(A))) & = \\
A \wedge (B \vee (B \wedge A)) & & &
\end{array}$$

- Formuli $A \wedge (B \vee (B \wedge A))$ lze dále zjednodušit na (ekvivalentní) formuli $A \wedge B$. To ale je již matematicky neformální (heuristický) postup.

Problém splnitelnosti výrokových formulí

Definice 9. *Formule* $\varphi \in \Phi$ *je* splnitelná*, právě když existuje valuace* ν *taková, že* $\mathcal{S}_\nu(\varphi) = true$.

Problém: Splnitelnost výrokových formulí (SAT)
Instance: $\varphi \in \Phi$
Otázka: Je φ splnitelná?

Je problém splnitelnosti algoritmicky řešitelný?

- Ano. Obsahuje-li daná $\varphi \in \Phi$ právě n výrokových proměnných, stačí vyzkoušet všechny možné valuace těchto proměnných, kterých je 2^n.
- Výše uvedený algoritmus není příliš praktický. Předpokládejme, že jednu valuaci lze vyzkoušet za 1 nanosekundu. Uvažme formuli, které má 100 výrokových proměnných. Pak všechny valuace vyzkoušíme zhruba za 10^{47} let.

- Počet „časových jednotek“, které počítač potřebuje k realizaci uvedeného algoritmu řešení SAT pro formuli s n proměnnými, je exponenciální v n.
  Otázka, zda existuje nějaký „lepší“ algoritmus, který vyžaduje pouze $p(n)$ časových jednotek (kde p je nějaký polynom v proměnné n), je ekvivalentní otázce zda $\mathcal{P} = \mathcal{NP}$, což je nejslavnější otevřený problém teoretické informatiky.

- Pomocí „rychlého“ algoritmu pro problém SAT by bylo možné „rychle“ řešit celou řadu dalších problémů, dá se říci že skoro všechny „praktické“ problémy. (Např. rozložit dané přirozené číslo na součin prvočinitelů, tedy „definitivně“ prolomit RSA šifru. Stejně tak pro DSA šifry.)

- Pro zajímavost, existuje matematický model hypotetického výpočetního zařízení (kvantový počítač), na němž je možné prvočíselný rozklad spočítat „rychle“. Princip fungování kvantového počítače je značně komplikovaný a není jasné, zda a kdy se ho technicky podaří sestrojit.
  Pro rychlejší řešení problému SAT však, zdá se, ani kvantový počítač nepomůže.

Neformální zmínka o predikátové logice

- Predikátová logika je obecnější než logika výroková; každá formule výrokové logiky je i formulí predikátové logiky, ale ne obráceně.
- Predikátová logika pracuje s *predikáty*. Predikáty jsou „parametrizované výroky“, které jsou buď pravdivé nebo nepravdivé pro každou konkrétní volbu parametrů. Výrokové proměnné lze chápat jako predikáty bez parametrů.
  - $x > 3$
  - R je ekvivalence na M
  - čísla x a y jsou nesoudělná
  - $P(x, y, z)$
- Z predikátů lze vytvářet komplikovanější formule pomocí výrokových spojek a *kvantifikátorů*.
  - $\forall x \,.\, \varphi$ „pro každou volbu parametru x platí formule φ“
  - $\exists x \,.\, \varphi$ „existuje alespoň jedna volba parametru x, pro kterou platí φ“

- Poukud není z kontextu jasné, co lze za daný parametr dosazovat, užívá se notace $\forall x \in M \,.\, \varphi$ a $\exists x \in M \,.\, \varphi$ (jen pokud je daný parametr prvkem nějaké množiny!).
- Tečka za symbolem kvantifikátoru se někdy vynechává (při vhodném uzávorkování formule), nebo se používá symbol „ : “.
- Místo $\forall x_1 \,.\, \forall x_2 \,.\, \cdots \forall x_n \,.\, \varphi$ se někdy krátce píše $\forall x_1, x_2, \cdots, x_n \,.\, \varphi$.
  Podobně u existenčního kvantifikátoru.
- Příklady:
  - Každé prvočíslo větší než 2 je liché.

$$\forall x \in \mathbb{N} \,.\; (P(x) \wedge x{>}2) \;\Rightarrow\; L(x)$$

  - Každé číslo n, které není prvočíslem, je dělitelné nějakým číslem y kde $n \neq y$ a $y > 1$.

$$\forall n. \,(\neg P(n) \Rightarrow\; \exists y \,.\, (y|n \wedge n{\neq}y \wedge y{>}1))$$

  - Jsou-li R a S ekvivalence na M, je také $R \cup S$ ekvivalence na M.

$$\forall M \; \forall R, S \;:\; (E(M,R) \wedge E(M,S)) \;\Rightarrow\; E(M, R{\cup}S)$$

- Je-li každá proměnná v dané formuli kvantifikovaná (tj. formule je *uzavřená*), pak je celá formule buď pravdivá nebo nepravdivá.

- Jak negovat formule predikátové logiky?

$$\begin{array}{lcllcl} \mathcal{F}(P(x_1,\ldots,x_n)) & = & P(x_1,\ldots,x_n) & \mathcal{G}(P(x_1,\ldots,x_n)) & = & \neg P(x_1,\ldots,x_n) \\ \mathcal{F}(\forall x\,.\,\varphi) & = & \forall x\,.\,\mathcal{F}(\varphi) & \mathcal{G}(\forall x\,.\,\varphi) & = & \exists x\,.\,\mathcal{G}(\varphi) \\ \mathcal{F}(\exists x\,.\,\varphi) & = & \exists x\,.\,\mathcal{F}(\varphi) & \mathcal{G}(\exists x\,.\,\varphi) & = & \forall x\,.\,\mathcal{G}(\varphi) \end{array}$$

- Uvažme například formuli

$$\neg(\forall M\ \forall R,S\ :\ (E(M,R)\wedge E(M,S))\ \Rightarrow\ E(M,R\cup S))\,.$$

Pak

$$\mathcal{F}(\neg(\forall M\ \forall R,S\ :\ (E(M,R)\wedge E(M,S))\ \Rightarrow\ E(M,R\cup S))) =$$

$$\mathcal{G}(\forall M\ \forall R,S\ :\ (E(M,R)\wedge E(M,S))\ \Rightarrow\ E(M,R\cup S)) =$$

$$\exists M\ \exists R,S.\ \mathcal{G}((E(M,R)\wedge E(M,S))\ \Rightarrow\ E(M,R\cup S)) =$$

$$\exists M\ \exists R,S\,.\ (E(M,R)\ \wedge\ E(M,S)\ \wedge\ \neg E(M,R\cup S))$$

Induktivní definice množin

Buď N množina. Induktivní definice (nějaké) množiny $M \subseteq N$ má obecně tento tvar:

- Vymezení *bázových* prvků M. (Každý bázový prvek musí být prvkem N.)
- Vymezení konečně mnoha *induktivních pravidel* (neboli uzávěrových vlastností) množiny M.

  Induktivní pravidlo je určeno funkcí $f : N^n \to N$, kde $n \in \mathbb{N}$, a je tohoto tvaru:

  Jestliže $x_1, \cdots, x_n \in M$, pak také $f(x_1, \cdots, x_n) \in M$.

  (Říkáme také, že M je uzavřená na f).
- Posledním bodem induktivní definice, který se často vynechává, je požadavek, že každý prvek M vznikne z bázových prvků konečně mnoha aplikacemi induktivních pravidel.

Definice 10. *Řekneme, že daná induktivní definice množiny M je* jednoznačná*, právě když každý prvek M lze odvodit z bázových prvků pomocí induktivních pravidel právě jedním způsobem.*

Příklady:

- Definujme množinu $M \subseteq \mathbb{N}$ induktivně takto:
  - $3 \in M$.
  - Jestliže $x \in M$, pak $x + 2 \in M$.

  Pak M je přesně množina všech lichých čísel větších než 1.

- Pro každé $y \in \mathbb{N}$ definujme množinu $M_y \subseteq \mathbb{N}$ induktivně takto:
  - $y \in M_y$.
  - Jestliže $x \in M_y$ a $x + 1$ je liché, pak $x + 2 \in M$.

  Pak např. $M_3 = \{3\}$, $M_4 = \{4 + 2i \mid i \in \mathbb{N}_0\}$.

- Definujme množinu $M \subseteq \mathbb{N}$ induktivně takto:
  - $3, 11 \in M$
  - Jestliže $x, y, z \in M$, pak také $(x + y)^z$, $x.y$, $x^{x.y.z}$ jsou prvky M.

Pro každou množinu Σ označíme symbolem Σ^* množinu všech konečných posloupností složených z prvků Σ. Např. $\mathtt{ababab} \in \{\mathtt{a}, \mathtt{b}\}^*$, $(3) \oplus (2) \in \{(,), \oplus, 2, 3\}^*$, apod.

- Nechť $\Sigma = At \cup \{\neg, \Longrightarrow, (,)\}$. Množina $\Phi \subseteq \Sigma^*$ všech výrokových formulí byla definována induktivně.

- Nechť $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \odot, \oplus, (,)\}$ Definujme množinu jednoduchých výrazů $SExp \subseteq \Sigma^*$ induktivně takto:
  - Dekadický zápis každého přirozeného čísla je prvek $SExp$.
  - Jestliže $\mathtt{x}, \mathtt{y} \in SExp$, pak také $(\mathtt{x}) \odot (\mathtt{y})$ a $(\mathtt{x}) \oplus (\mathtt{y})$ jsou prvky $SExp$.

Strukturální indukce

- Důkazová technika aplikovatelná na tvrzení typu

  pro každé $x \in M$ platí $T(x)$,

  kde $M \subseteq N$ je definovaná induktivně.

- Princip strukturální indukce říká, že k důkazu věty

  Pro každé $x \in M$ platí $T(x)$

  stačí ověřit platnost těchto tvrzení:

  * Pro každý bázový prvek x množiny M platí $T(x)$.
  * Pro každé induktivní pravidlo zadané funkcí $f : N^n \to N$ a každá $x_1, \cdots, x_n \in M$ platí, že jestliže $T(x_1), \cdots, T(x_n)$ jsou pravdivá, pak $T(f(x_1, \cdots, x_n))$ je také pravdivé.

  (Postupuje se vlastně indukcí podle „hloubky odvození“ každého prvku M.)

Induktivně definované funkce z induktivně definovaných množin

Nechť $M \subseteq N$ je induktivně definovaná, a nechť je tato definice jednoznačná. Induktivní definice funkcí $\mathcal{F}_1 : M \to K_1, \cdots, \mathcal{F}_k : M \to K_k$ má obecně tento tvar:

- Pro každé $1 \leq i \leq k$ a každý bázový prvek x množiny M se definuje hodnota $\mathcal{F}_i(x)$.
- Pro každé induktivní pravidlo a každé $1 \leq i \leq k$ se provede následující: Nechť $f : N^n \to N$ je funkce asociovaná s daným pravidlem. Pro každé $x_1, \cdots, x_n \in M$ se definuje hodnota $\mathcal{F}_i(f(x_1, \cdots, x_n))$ na základě hodnot $\mathcal{F}_1(x_1), \ldots, \mathcal{F}_1(x_n), \cdots, \mathcal{F}_k(x_1), \ldots, \mathcal{F}_k(x_n)$.

Platí, že pro každé $1 \leq i \leq k$ a každé $x \in M$ je tímto způsobem jednoznačně definováno $\mathcal{F}_i(x)$.

Příklady:

- Funkce $\mathcal{S}_{\mathrm{v}} : \Phi \to \{true, false\}$ byla definována induktivně.
- Funkce $\mathcal{F}, \mathcal{G} : \Phi \to \Phi$ byly definovány induktivně.
- Definujme funkci *Val* : $SExp \to \mathbb{N}_0$ induktivně takto:
  - *Val*(**n**) $= \mathrm{n}$, kde **n** je dekadický zápis přirozeného čísla n.
  - *Val*$((\mathrm{x}) \odot (\mathrm{y})) =$ *Val*$(\mathrm{x}) \cdot$ *Val*(y)
  - *Val*$((\mathrm{x}) \oplus (\mathrm{y})) =$ *Val*$(\mathrm{x}) +$ *Val*(y)

Dokazování vlastností programů

- Jak se přesvědčit, že je daný program „správný“?
- Co třeba ladění programů?
  Jelikož počet možných vstupních hodnot je (v principu) neohraničený, nelze „otestovat“ všechna možná vstupní data.
- Situace je zvláště komplikovaná v případě paralelních, randomizovaných, interaktivních a nekončících programů (operační systémy, systémy řízení letecké dopravy, apod.). Takové systémy mají nedeterministické chování a opakované „experimenty“ tudíž vedou k různým výsledkům (nelze je „ladit“).
- V některých případech je však třeba mít „naprostou jistotu“, že program funguje tak jak má, případně že splňuje základní bezpečnostní požadavky.
- Narůstající složitost programových systémů a zvýšené požadavky na jejich bezpečnost si vynucují vývoj „spolehlivých“ verifikačních metod.

Příklady programů

- Například následující kód v C počítá symetrický uzávěr relace R:

```
for (i=0; i<N; i++)
  for (j=0; j<N; j++)
    if (R[i][j])  R[j][i] = 1;
```

  Dokážete zdůvodnit jeho správnost?
  (Počítá tento kód zároveň i reflexivní uzávěr? Jak byste jej jednoduše upravili, aby uzávěr byl i reflexivní?)

- Co třeba dělá tento kód? Co je na něm problematického?

```
for (i=0; i? D[i-1]==0: 1; i++)
  D[i] = (D[i]+1)%10;
```

  (Inkrementuje dekadický zápis čísla D uloženého po číslicích v poli. Může však „přetéct“ hranici pole D.)

- A co myslíte, že dělá následující C kód pro celá a, b?

```
while (a>0 && b>0) {
  while (a<=b)  b = b-a;
  x = a; a = b; b = x;
}
printf("Vysledek je %d.\n",a+b);
```

  Nejprve se zamyslete, proč tato smyčka vždy skončí pro jakákoliv celá a, b. (Kód vypočítá největšího společného dělitele čísel a, b.)

Jak vidíme, není vždy na první pohled zřejmé, co i krátký programový kód „dělá“. Pokud chceme mít naprostou jistotu, co daný program dělá, musíme podat matematický důkaz správnosti.

Jednoduchý deklarativní programovací jazyk

- Nechť $Var = \{x, y, z, \ldots\}$ je spočetná množina proměnných.
- Nechť $Num = \{\mathbf{0}, \mathbf{1}, \mathbf{52}, \mathbf{397}, \ldots\}$ je množina všech dekadických zápisů přirozených čísel.
- Nechť $FVar = \{f, g, h, \ldots\}$ je spočetná množina funkčních symbolů. Ke každému $f \in FVar$ je přiřazeno číslo $a \in \mathbb{N}$, které nazýváme arita f. Dále předpokládáme, že pro každé $a \in \mathbb{N}$ existuje nekonečně mnoho $f \in FVar$ s aritou a.
- Množina *výrazů* Exp je (induktivně) definována následující abstraktní syntaktickou rovnicí:

$$
\begin{array}{rcl}
E & ::= & x \mid \mathbf{n} \\
 & \mid & E_1 + E_2 \mid E_1 - E_2 \mid E_1 * E_2 \mid E_1 \div E_2 \mid (E_1) \\
 & \mid & f(E_1, \cdots, E_a) \\
 & \mid & \textbf{if } E_1 \textbf{ then } E_2 \textbf{ else } E_3
\end{array}
$$

V uvedené rovnici je $x \in Var$, $\mathbf{n} \in Num$, $f \in FVar$ a $a \in \mathbb{N}$ je arita daného f.

- Takováto specifikace syntaxe je *abstraktní* v tom smyslu, že se nezabývá tím, jak výrazy jednoznačně zapsat do řádku jako posloupnost symbolů. Je na nás, abychom napsali dostatečně mnoho závorek a případně stanovili prioritu operátorů tak, aby bylo zcela jasné, jak daný výraz podle uvedené rovnice vznikl.
- Příklady:
  - $\mathbf{2} + \mathbf{3} * \mathbf{4}$
  - $f(\mathbf{2} + x, g(y, \mathbf{3} * y))$
  - **if** $x - \mathbf{1}$ **then** $(\mathbf{2} + f(y))$ **else** $g(x, x)$
    (Vyhodnocení podmínky v „if“ testuje nenulovost argumentu.)
- Deklarace je konečný systém rovnic tvaru

$$\begin{array}{lcl} f_1(x_1, \cdots, x_{a_1}) & = & E_1 \\ \quad \vdots & & \vdots \\ f_n(x_1, \cdots, x_{a_n}) & = & E_n \end{array}$$

kde pro každé $1 \leq i \leq n$ platí, že $f_i \in \mathit{FVar}$, a_i je arita f_i, $x_1, \cdots, x_{a_i} \in \mathit{Var}$ a E_i je výraz, v němž se mohou vyskytovat pouze proměnné $x_1, \cdots, x_{a_i}$ a funkční symboly $f_1, \cdots, f_n$.

- Příklady:
  - $f(x) = \textbf{if } x \textbf{ then } x * f(x-1) \textbf{ else } 1$
  - $\begin{aligned} f(x) &= g(x-1, x) \\ g(x, y) &= \textbf{if } x \textbf{ then } f(y) \textbf{ else } 3 \end{aligned}$

    (Jak uvidíme formálně později, konvencí našich výpočtů je neužívat záporná čísla, místo toho $0 - 1$ *„=“* 0*.)*
  - $f(x) = f(x)$

    (Nezapisuje toto náhodou „nekonečnou smyčku“?)

Formalizace pojmů „výpočetní krok“ a „výpočet“

- Buď Δ deklarace. Symbolem $Exp(\Delta)$ označíme množinu všech výrazů $\mathtt{E}$, které splňují tyto dvě podmínky:
  - $\mathtt{E}$ neobsahuje žádnou proměnnou;
  - jestliže $\mathtt{E}$ obsahuje funkční symbol $\mathtt{f}$, pak $\mathtt{f}$ byl v Δ deklarován.
- Množinu $Exp(\Delta)$ lze definovat také induktivně:

  $$\mathtt{E} \quad ::= \quad \mathbf{n} \mid \mathtt{E}_1 + \mathtt{E}_2 \mid \mathtt{E}_1 - \mathtt{E}_2 \mid \mathtt{E}_1 * \mathtt{E}_2 \mid \mathtt{E}_1 \div \mathtt{E}_2 \mid \mathtt{f}(\mathtt{E}_1, \cdots, \mathtt{E}_a) \mid \textbf{if } \mathtt{E}_1 \textbf{ then } \mathtt{E}_2 \textbf{ else } \mathtt{E}_3$$

  V uvedené rovnici je $\mathbf{n} \in Num$, $\mathtt{f}$ je funkční symbol deklarovaný v Δ a $a \in \mathbb{N}$ je arita daného $\mathtt{f}$.
- Induktivně definujeme funkci „krok výpočtu“ $\mapsto : Exp(\Delta) \to Exp(\Delta)$; místo $\mapsto(\mathtt{E}) = \mathtt{F}$ budeme psát $\mathtt{E} \mapsto \mathtt{F}$.
  - $\mathbf{n} \mapsto \mathbf{n}$ pro každé $\mathbf{n} \in Num$.

* Pro $\underline{\mathtt{E} = \mathtt{E}_1 + \mathtt{E}_2}$ definujeme krok výpočtu takto:
  - Jestliže $\mathtt{E}_1, \mathtt{E}_2 \in \mathit{Num}$, pak $\mathtt{E}_1 + \mathtt{E}_2 \mapsto \mathbf{z}$ kde $\mathbf{z}$ je dekadický zápis čísla $\mathtt{E}_1 + \mathtt{E}_2$.
  - Jestliže $\mathtt{E}_1 \notin \mathit{Num}$, pak $\mathtt{E}_1 + \mathtt{E}_2 \mapsto \mathtt{F} + \mathtt{E}_2$ kde $\mathtt{E}_1 \mapsto \mathtt{F}$.
  - Jestliže $\mathtt{E}_1 \in \mathit{Num}$ a $\mathtt{E}_2 \notin \mathit{Num}$, pak $\mathtt{E}_1 + \mathtt{E}_2 \mapsto \mathtt{E}_1 + \mathtt{F}$ kde $\mathtt{E}_2 \mapsto \mathtt{F}$.
* Pro $\underline{\mathtt{E} = \mathtt{E}_1 - \mathtt{E}_2}$ definujeme krok výpočtu takto:
  - Jestliže $\mathtt{E}_1, \mathtt{E}_2 \in \mathit{Num}$, pak $\mathtt{E}_1 - \mathtt{E}_2 \mapsto \mathbf{z}$ kde $\mathbf{z}$ je dekadický zápis čísla $\max\{0, \mathtt{E}_1 - \mathtt{E}_2\}$ (nezápornost výsledku!).
  - Jestliže $\mathtt{E}_1 \notin \mathit{Num}$, pak $\mathtt{E}_1 - \mathtt{E}_2 \mapsto \mathtt{F} - \mathtt{E}_2$ kde $\mathtt{E}_1 \mapsto \mathtt{F}$.
  - Jestliže $\mathtt{E}_1 \in \mathit{Num}$ a $\mathtt{E}_2 \notin \mathit{Num}$, pak $\mathtt{E}_1 - \mathtt{E}_2 \mapsto \mathtt{E}_1 - \mathtt{F}$ kde $\mathtt{E}_2 \mapsto \mathtt{F}$.
* Pro $\underline{\mathtt{E} = \mathtt{E}_1 * \mathtt{E}_2}$ definujeme krok výpočtu takto:
  - Jestliže $\mathtt{E}_1, \mathtt{E}_2 \in \mathit{Num}$, pak $\mathtt{E}_1 * \mathtt{E}_2 \mapsto \mathbf{z}$ kde $\mathbf{z}$ je dekadický zápis čísla $\mathtt{E}_1 * \mathtt{E}_2$.
  - Jestliže $\mathtt{E}_1 \notin \mathit{Num}$, pak $\mathtt{E}_1 * \mathtt{E}_2 \mapsto \mathtt{F} * \mathtt{E}_2$ kde $\mathtt{E}_1 \mapsto \mathtt{F}$.
  - Jestliže $\mathtt{E}_1 \in \mathit{Num}$ a $\mathtt{E}_2 \notin \mathit{Num}$, pak $\mathtt{E}_1 * \mathtt{E}_2 \mapsto \mathtt{E}_1 * \mathtt{F}$ kde $\mathtt{E}_2 \mapsto \mathtt{F}$.

- Pro $\underline{\mathtt{E} = \mathtt{E}_1 \div \mathtt{E}_2}$ definujeme krok výpočtu takto:
  - Jestliže $\mathtt{E}_1, \mathtt{E}_2 \in \mathit{Num}$, pak $\mathtt{E}_1 \div \mathtt{E}_2 \mapsto \mathbf{z}$ kde $\mathbf{z}$ je dekadický zápis celé části čísla $\mathtt{E}_1/\mathtt{E}_2$. Pokud $\mathtt{E}_2 = \mathbf{0}$, je $\mathbf{z} = \mathbf{0}$.
  - Jestliže $\mathtt{E}_1 \notin \mathit{Num}$, pak $\mathtt{E}_1 \div \mathtt{E}_2 \mapsto \mathtt{F} \div \mathtt{E}_2$ kde $\mathtt{E}_1 \mapsto \mathtt{F}$.
  - Jestliže $\mathtt{E}_1 \in \mathit{Num}$ a $\mathtt{E}_2 \notin \mathit{Num}$, pak $\mathtt{E}_1 \div \mathtt{E}_2 \mapsto \mathtt{E}_1 \div \mathtt{F}$ kde $\mathtt{E}_2 \mapsto \mathtt{F}$.
- Pro $\underline{\mathtt{E} = \textbf{if } \mathtt{E}_1 \textbf{ then } \mathtt{E}_2 \textbf{ else } \mathtt{E}_3}$ definujeme krok výpočtu takto:
  - Jestliže $\mathtt{E}_1 \in \mathit{Num}$ a $\mathtt{E}_1 = \mathbf{0}$, pak **if** $\mathtt{E}_1$ **then** $\mathtt{E}_2$ **else** $\mathtt{E}_3 \mapsto \mathtt{E}_3$.
  - Jestliže $\mathtt{E}_1 \in \mathit{Num}$ a $\mathtt{E}_1 \neq \mathbf{0}$, pak **if** $\mathtt{E}_1$ **then** $\mathtt{E}_2$ **else** $\mathtt{E}_3 \mapsto \mathtt{E}_2$.
  - Jestliže $\mathtt{E}_1 \notin \mathit{Num}$, pak **if** $\mathtt{E}_1$ **then** $\mathtt{E}_2$ **else** $\mathtt{E}_3 \mapsto$ **if** $\mathtt{F}$ **then** $\mathtt{E}_2$ **else** $\mathtt{E}_3$ kde $\mathtt{E}_1 \mapsto \mathtt{F}$.
- Pro $\underline{\mathtt{E} = \mathtt{f}(\mathtt{E}_1, \cdots, \mathtt{E}_k)}$ definujeme krok výpočtu takto:
  - Jestliže $\mathtt{E}_1, \cdots, \mathtt{E}_k \in \mathit{Num}$, pak $\mathtt{f}(\mathtt{E}_1, \cdots, \mathtt{E}_k) \mapsto \mathtt{E}(\mathtt{x}_1 \restriction \mathtt{E}_1, \cdots, \mathtt{x}_k \restriction \mathtt{E}_k)$
  - Jinak $\mathtt{f}(\mathtt{E}_1, \cdots, \mathtt{E}_k) \mapsto \mathtt{f}(\mathtt{E}_1, \cdots, \mathtt{E}_{i-1}, \mathtt{F}, \mathtt{E}_{i+1}, \cdots, \mathtt{E}_k)$, kde i je nejmenší index pro který platí $\mathtt{E}_i \notin \mathit{Num}$ a $\mathtt{E}_i \mapsto \mathtt{F}$.
- Reflexivní a tranzitivní uzávěr relace $\mapsto$ značíme $\mapsto^*$ („výpočet“).

Příklady výpočtů

- Uvažme deklaraci $f(x) = \textbf{if } x \textbf{ then } x * f(x-1) \textbf{ else } 1$. Pak např. $f(3) \mapsto^* 6$, neboť

$$\begin{array}{lcllcllcl}
f(3) & \mapsto & \textbf{if } 3 \textbf{ then } 3 * f(3-1) \textbf{ else } 1 & \mapsto & 3 * f(3-1) & \mapsto \\
3 * f(2) & \mapsto & 3 * (\textbf{if } 2 \textbf{ then } 2 * f(2-1) \textbf{ else } 1) & \mapsto & 3 * (2 * f(2-1)) & \mapsto \\
3 * (2 * f(1)) & \mapsto & 3 * (2 * (\textbf{if } 1 \textbf{ then } 1 * f(1-1) \textbf{ else } 1)) & \mapsto & 3 * (2 * (1 * f(1-1))) & \mapsto \\
3 * (2 * (1 * f(0))) & \mapsto & 3 * (2 * (1 * \textbf{if } 0 \textbf{ then } 0 * f(0-1) \textbf{ else } 1)) & \mapsto & 3 * (2 * (1 * 1)) & \mapsto \\
3 * (2 * 1) & \mapsto & 3 * 2 & \mapsto & 6 &
\end{array}$$

- Uvažme deklaraci $f(x) = g(x-1, x)$, $g(x, y) = \textbf{if } x \textbf{ then } f(y) \textbf{ else } 3$. Pak např. $f(3) \mapsto^* f(3)$, neboť

$$f(3) \mapsto g(3-1, 3) \mapsto g(2, 3) \mapsto \textbf{if } 2 \textbf{ then } f(3) \textbf{ else } 3 \mapsto f(3)$$

- Uvažme deklaraci $f(x) = f(x)$. Pak pro každé $n \in Num$ platí $f(n) \mapsto f(n)$ a podobně $f(f(n)) \mapsto f(f(n))$. Ale $f(f(2+3)) \mapsto f(f(5)) \mapsto f(f(5))$.

Důkaz správnosti programu

Příklad 14. Věta: *Uvažme deklaraci Δ obsahující pouze rovnici*

$$\mathrm{f}(\mathrm{x}) = \textbf{\textit{if}}\ \mathrm{x}\ \textbf{\textit{then}}\ \mathrm{x} * \mathrm{f}(\mathrm{x} - \mathbf{1})\ \textbf{\textit{else}}\ \mathbf{1}$$

Pak pro každé $\mathrm{n} \in \mathbb{N}_0$ *platí* $\mathrm{f}(\mathbf{n}) \mapsto^* \mathbf{m}$*, kde* $\mathbf{m} \equiv \mathrm{n}!$*.*

Důkaz. Indukcí vzhledem k n.

- $\mathrm{n} = 0$. Platí $\mathrm{f}(\mathbf{0}) \ \mapsto\ $ **if 0 then** $\mathbf{0} * \mathrm{f}(\mathbf{0} - \mathbf{1})$ **else** $\mathbf{1}\ \mapsto\ \mathbf{1}$.
- Indukční krok. Nechť $\mathrm{n} + 1 \equiv \mathbf{k}$. Pak

  $$\mathrm{f}(\mathbf{k}) \quad \mapsto \quad \textbf{if } \mathbf{k} \textbf{ then } \mathbf{k} * \mathrm{f}(\mathbf{k} - \mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \mathbf{k} * \mathrm{f}(\mathbf{k} - \mathbf{1}) \quad \mapsto \quad \mathbf{k} * \mathrm{f}(\mathbf{w})$$

  kde $\mathbf{w} \equiv \mathrm{n}$. Podle I.P. platí $\mathrm{f}(\mathbf{w}) \ \mapsto^*\ \mathbf{u}$, kde $\mathbf{u} \equiv \mathrm{n}!$. Proto $\mathbf{k} * \mathrm{f}(\mathbf{w}) \ \mapsto^*\ \mathbf{k} * \mathbf{u} \ \mapsto\ \mathbf{v}$, kde $\mathbf{v} \equiv (\mathrm{n} + 1) \cdot \mathrm{n}! = (\mathrm{n} + 1)!$.

□

Důkazy „neukončenosti“ výpočtů

Věta 11. *Buď Δ deklarace. Pro každé $\mathtt{i} \in \mathbb{N}$ definujeme relaci $\mapsto^{\mathtt{i}} \subseteq Exp(\Delta) \times Exp(\Delta)$ předpisem $\mapsto^{\mathtt{i}} = \underbrace{\mapsto \circ \cdots \circ \mapsto}_{\mathtt{i}}$. Dále definitoricky klademe $\mapsto^{0} = \{(\mathtt{E}, \mathtt{E}) \mid \mathtt{E} \in Exp(\Delta)\}$. Pak $\mapsto^{*} = \bigcup_{\mathtt{i}=0}^{\infty} \mapsto^{\mathtt{i}}$.*

Podle předchozí věty platí, že $\mathtt{E} \mapsto^{*} \mathtt{F}$ právě když $\mathtt{E} \mapsto^{\mathtt{i}} \mathtt{F}$ pro nějaké $\mathtt{i} \in \mathbb{N}_0$. Navíc musí existovat nejmenší $\mathtt{i}$ s touto vlastností. Toto pozorování může být užitečné v důkazech „neukončenosti“ výpočtů.

Příklad 15. Věta: *Uvažme deklaraci $\mathtt{f}(\mathtt{x}) = \mathtt{f}(\mathtt{x})$. Pro každé $\mathbf{n} \in Num$ platí, že neexistuje žádné $\mathbf{m} \in Num$ takové, že $\mathtt{f}(\mathbf{n}) \mapsto^{*} \mathbf{m}$.*

Důkaz. Sporem. Předpokládejme, že existují $\mathbf{n}, \mathbf{m} \in Num$ takové, že $\mathtt{f}(\mathbf{n}) \mapsto^{*} \mathbf{m}$. Pak existuje nejmenší $\mathtt{i} \in \mathbb{N}_0$ takové, že $\mathtt{f}(\mathbf{n}) \mapsto^{\mathtt{i}} \mathbf{m}$. Jelikož výrazy $\mathtt{f}(\mathbf{n})$ a $\mathbf{m}$ jsou různé, platí $\mathtt{i} > 0$. Jelikož $\mapsto^{\mathtt{i}} = \mapsto^{\mathtt{i}-1} \circ \mapsto$ a $\mathtt{f}(\mathbf{n}) \mapsto \mathtt{f}(\mathbf{n})$, platí $\mathtt{f}(\mathbf{n}) \mapsto^{\mathtt{i}-1} \mathbf{m}$, což je spor s minimalitou $\mathtt{i}$. □

Další příklady I („fixace parametru“)

Příklad 16. Věta: *Uvažme deklaraci Δ obsahující pouze rovnici*

$$\mathrm{g}(\mathrm{x},\mathrm{y}) = \textbf{\textit{if}}\ \mathrm{x}\ \textbf{\textit{then}}\ \mathrm{y} + \mathrm{g}(\mathrm{x}-\mathbf{1},\mathrm{y})\ \textbf{\textit{else}}\ \mathbf{0}$$

Pak pro každé $\mathrm{m},\mathrm{n} \in \mathbb{N}_0$ *platí* $\mathrm{g}(\mathbf{m},\mathbf{n}) \mapsto^* \mathbf{z}$*, kde* $\mathbf{z} \equiv \mathrm{m}\cdot\mathrm{n}$*.*

Důkaz. Buď $\mathrm{n} \in \mathbb{N}_0$ libovolné ale pro další úvahy pevné. Dokážeme, že pro každé $\mathrm{m} \in \mathbb{N}_0$ platí $\mathrm{g}(\mathbf{m},\mathbf{n}) \mapsto^* \mathbf{z}$, kde $\mathbf{z} \equiv \mathrm{m}\cdot\mathrm{n}$. Indukcí vzhledem k m.

- $\mathrm{m} = 0$. Platí $\mathrm{g}(\mathbf{0},\mathbf{n}) \mapsto$ **if** $\mathbf{0}$ **then** $\mathbf{n} + \mathrm{g}(\mathbf{0}-\mathbf{1},\mathbf{n})$ **else** $\mathbf{0} \mapsto \mathbf{0}$.
- Indukční krok. Nechť $\mathrm{m}+1 \equiv \mathbf{k}$. Pak

  $$\mathrm{g}(\mathbf{k},\mathbf{n}) \quad\mapsto\quad \textbf{if}\ \mathbf{k}\ \textbf{then}\ \mathbf{n} + \mathrm{g}(\mathbf{k}-\mathbf{1},\mathbf{n})\ \textbf{else}\ \mathbf{0} \quad\mapsto\quad \mathbf{n} + \mathrm{g}(\mathbf{k}-\mathbf{1},\mathbf{n}) \quad\mapsto\quad \mathbf{n} + \mathrm{g}(\mathbf{w},\mathbf{n})$$

  kde $\mathbf{w} \equiv \mathrm{m}$. Podle I.P. platí $\mathrm{g}(\mathbf{w},\mathbf{n}) \mapsto^* \mathbf{u}$, kde $\mathbf{u} \equiv \mathrm{m}\cdot\mathrm{n}$. Dále $\mathbf{n} + \mathrm{g}(\mathbf{w},\mathbf{n}) \mapsto^* \mathbf{n}+\mathbf{u} \mapsto \mathbf{v}$, kde $\mathbf{v} \equiv \mathrm{n} + (\mathrm{m}\cdot\mathrm{n}) = (\mathrm{m}+1)\cdot\mathrm{n}$.

□

Další příklady II („indukce k součtu parametrů“)

Příklad 17. Věta: *Uvažme deklaraci Δ obsahující pouze rovnici*

$$\mathrm{g}(\mathrm{x},\mathrm{y}) = \textbf{\textit{if}}\ \mathrm{x}\ \textbf{\textit{then}}\ (\textbf{\textit{if}}\ \mathrm{y}\ \textbf{\textit{then}}\ \mathrm{g}(\mathrm{x}-\mathbf{1},\mathrm{y}) + \mathrm{g}(\mathrm{x},\mathrm{y}-\mathbf{1})\ \textbf{\textit{else}}\ \mathbf{0})\ \textbf{\textit{else}}\ \mathbf{0}$$

Pak pro každé $\mathrm{m},\mathrm{n} \in \mathbb{N}_0$ *platí* $\mathrm{g}(\mathbf{m},\mathbf{n}) \mapsto^* \mathbf{0}$*.*

Důkaz. Tvrzení

Pro každé $\mathrm{m},\mathrm{n} \in \mathbb{N}_0$ platí $\mathrm{g}(\mathbf{m},\mathbf{n}) \mapsto^* \mathbf{0}$

nelze dokázat indukcí vzhledem k m ani indukcí vzhledem k n. Důkaz lze ovšem vést indukcí k „součtu“ m a n. To znamená, že výše uvedené tvrzení nejprve přeformulujeme do následující (ekvivalentní) podoby:

Pro každé $\mathrm{i} \in \mathbb{N}_0$ platí, že jestliže $\mathrm{i} = \mathrm{m}+\mathrm{n}$, kde $\mathrm{m},\mathrm{n} \in \mathbb{N}_0$, pak $\mathrm{g}(\mathbf{m},\mathbf{n}) \mapsto^* \mathbf{0}$

Toto tvrzení nyní dokážeme indukcí vzhledem k i:

- $i = 0$. Jestliže $i = m + n$, kde $m, n \in \mathbb{N}_0$, pak $m = n = 0$. Dokážeme, že $g(\mathbf{0}, \mathbf{0}) \mapsto^* \mathbf{0}$. Platí

  $$g(\mathbf{0},\mathbf{0}) \quad \mapsto \quad \textbf{if } \mathbf{0} \textbf{ then } (\textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{0}-\mathbf{1},\mathbf{0}) + g(\mathbf{0},\mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{0}) \textbf{ else } \mathbf{0} \quad \mapsto \quad \mathbf{0}$$

- Indukční krok. Nechť $i+1 = m+n$, kde $m, n \in \mathbb{N}_0$. Nyní rozlišíme tři možnosti:
  - $m = 0$. Pak platí

    $$g(\mathbf{0},\mathbf{n}) \quad \mapsto \quad \textbf{if } \mathbf{0} \textbf{ then } (\textbf{if } \mathbf{n} \textbf{ then } g(\mathbf{0}-\mathbf{1},\mathbf{n}) + g(\mathbf{0},\mathbf{n}-\mathbf{1}) \textbf{ else } \mathbf{0}) \textbf{ else } \mathbf{0} \quad \mapsto \quad \mathbf{0}$$

  - $m > 0, n = 0$. Pak platí

    $$g(\mathbf{m},\mathbf{0}) \quad \mapsto \quad \textbf{if } \mathbf{m} \textbf{ then } (\textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{m}-\mathbf{1},\mathbf{0}) + g(\mathbf{m},\mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{0}) \textbf{ else } \mathbf{0} \quad \mapsto$$
    $$\textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{m}-\mathbf{1},\mathbf{0}) + g(\mathbf{m},\mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{0} \quad \mapsto \quad \mathbf{0}$$

  - $m > 0, n > 0$. Pak platí

    $$g(\mathbf{m},\mathbf{n}) \quad \mapsto \quad \textbf{if } \mathbf{m} \textbf{ then } (\textbf{if } \mathbf{n} \textbf{ then } g(\mathbf{m}-\mathbf{1},\mathbf{n}) + g(\mathbf{m},\mathbf{n}-\mathbf{1}) \textbf{ else } \mathbf{0}) \textbf{ else } \mathbf{0} \quad \mapsto$$
    $$\textbf{if } \mathbf{n} \textbf{ then } g(\mathbf{m}-\mathbf{1},\mathbf{n}) + g(\mathbf{m},\mathbf{n}-\mathbf{1}) \textbf{ else } \mathbf{0} \quad \mapsto \quad g(\mathbf{m}-\mathbf{1},\mathbf{n}) + g(\mathbf{m},\mathbf{n}-\mathbf{1})$$

    Podle I.P. platí $g(\mathbf{m}-\mathbf{1},\mathbf{n}) \mapsto^* \mathbf{0}$ a současně $g(\mathbf{m},\mathbf{n}-\mathbf{1}) \mapsto^* \mathbf{0}$, proto

    $$g(\mathbf{m}-\mathbf{1},\mathbf{n}) + g(\mathbf{m},\mathbf{n}-\mathbf{1}) \quad \mapsto^* \quad \mathbf{0} + g(\mathbf{m},\mathbf{n}-\mathbf{1}) \quad \mapsto^* \quad \mathbf{0}+\mathbf{0} \quad \mapsto^* \quad \mathbf{0}\,.$$

Tím jsme s důkazem matematickou indukcí hotovi. □

* Udělejme si předchozí nudný příklad trochu zajímavějším (ale co se týče důkazu stále v podstatě stejným. . .):

Příklad 18. Věta: *Uvažme deklaraci Δ obsahující pouze rovnici*

$$\mathrm{g}(\mathrm{x},\mathrm{y}) = \textbf{\textit{if}}\ \mathrm{x}\ \textbf{\textit{then}}\ (\textbf{\textit{if}}\ \mathrm{y}\ \textbf{\textit{then}}\ \mathrm{g}(\mathrm{x}-\mathbf{1},\mathrm{y}) + \mathrm{g}(\mathrm{x},\mathrm{y}-\mathbf{1})\ \textbf{\textit{else}}\ \mathbf{1})\ \textbf{\textit{else}}\ \mathbf{1}$$

Pak pro každé $\mathrm{m},\mathrm{n} \in \mathbb{N}_0$ *platí* $\mathrm{g}(\mathbf{m},\mathbf{n}) \mapsto^* \mathbf{k}$*, kde* $\mathrm{k} = \binom{\mathrm{m}+\mathrm{n}}{\mathrm{m}}$ *(kombinační číslo).*

Důkaz. Toto tvrzení opět dokážeme indukcí vzhledem k $\mathrm{i} = \mathrm{m} + \mathrm{n}$.

Vzpoměňte si nejprve na známý *Pascalův trojúhelník* kombinačních čísel, který je definovaný rekurentním vztahem

$$\binom{\mathrm{a}+1}{\mathrm{b}+1} = \binom{\mathrm{a}}{\mathrm{b}+1} + \binom{\mathrm{a}}{\mathrm{b}}.$$

Nepřipomíná to trochu naši deklaraci? Je však třeba správně „nastavit“ význam parametrů a,b.

- $\mathrm{i} = 0$. Jestliže $\mathrm{i} = \mathrm{m} + \mathrm{n}$, kde $\mathrm{m},\mathrm{n} \in \mathbb{N}_0$, pak $\mathrm{m} = \mathrm{n} = 0$. Dokážeme, že $\mathrm{g}(\mathbf{0},\mathbf{0}) \mapsto^* \mathbf{1}$. Platí

$$\mathrm{g}(\mathbf{0},\mathbf{0}) \quad\mapsto\quad \textbf{if } \mathbf{0} \textbf{ then } (\textbf{if } \mathbf{0} \textbf{ then } \mathrm{g}(\mathbf{0}-\mathbf{1},\mathbf{0}) + \mathrm{g}(\mathbf{0},\mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{1}) \textbf{ else } \mathbf{1} \quad\mapsto\quad \mathbf{1}$$

- Indukční krok. Nechť $i+1 = m+n$, kde $m, n \in \mathbb{N}_0$. Nyní rozlišíme tři možnosti:
  - $m = 0$. Pak platí

    $$g(\mathbf{0}, \mathbf{n}) \quad \mapsto \quad \textbf{if } \mathbf{0} \textbf{ then } (\textbf{if } \mathbf{n} \textbf{ then } g(\mathbf{0}-\mathbf{1}, \mathbf{n}) + g(\mathbf{0}, \mathbf{n}-\mathbf{1}) \textbf{ else } \mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \mathbf{1}$$

  - $m > 0, n = 0$. Pak platí

    $$g(\mathbf{m}, \mathbf{0}) \quad \mapsto \quad \textbf{if } \mathbf{m} \textbf{ then } (\textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{m}-\mathbf{1}, \mathbf{0}) + g(\mathbf{m}, \mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto$$

    $$\textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{m}-\mathbf{1}, \mathbf{0}) + g(\mathbf{m}, \mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \mathbf{1}$$

  - $m > 0, n > 0$. Pak platí

    $$g(\mathbf{m}, \mathbf{n}) \quad \mapsto \quad \textbf{if } \mathbf{m} \textbf{ then } (\textbf{if } \mathbf{n} \textbf{ then } g(\mathbf{m}-\mathbf{1}, \mathbf{n}) + g(\mathbf{m}, \mathbf{n}-\mathbf{1}) \textbf{ else } \mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto$$

    $$\textbf{if } \mathbf{n} \textbf{ then } g(\mathbf{m}-\mathbf{1}, \mathbf{n}) + g(\mathbf{m}, \mathbf{n}-\mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad g(\mathbf{m}-\mathbf{1}, \mathbf{n}) + g(\mathbf{m}, \mathbf{n}-\mathbf{1})$$

    Podle I.P. platí $g(\mathbf{m}-\mathbf{1}, \mathbf{n}) \mapsto^* \binom{\mathbf{m}+\mathbf{n}-\mathbf{1}}{\mathbf{m}-\mathbf{1}}$ a současně $g(\mathbf{m}, \mathbf{n}-\mathbf{1}) \mapsto^* \binom{\mathbf{m}+\mathbf{n}-\mathbf{1}}{\mathbf{m}}$. Přitom z Pascalova trojúhelníka plyne

    $$\binom{m+n-1}{m-1} + \binom{m+n-1}{m} = \binom{m+n-1+1}{m} = \binom{m+n}{m},$$

    a proto

    $$g(\mathbf{m}-\mathbf{1}, \mathbf{n}) + g(\mathbf{m}, \mathbf{n}-\mathbf{1}) \quad \mapsto^* \quad \binom{\mathbf{m}+\mathbf{n}}{\mathbf{m}}.$$

□

Další příklady III („zesílení dokazovaného tvrzení“)

Příklad 19. Věta: *Uvažme deklaraci Δ obsahující tyto rovnice:*

$$\mathrm{f}(\mathrm{x}) = \textbf{\textit{if}}\ \mathrm{x}\ \textbf{\textit{then}}\ \mathrm{h}(\mathrm{x})\ \textbf{\textit{else}}\ \mathbf{1}$$
$$\mathrm{h}(\mathrm{x}) = \textbf{\textit{if}}\ \mathrm{x}\ \textbf{\textit{then}}\ \mathrm{f}(\mathrm{x}-\mathbf{1}) + \mathrm{h}(\mathrm{x}-\mathbf{1})\ \textbf{\textit{else}}\ \mathbf{1}$$

Pak pro každé $\mathrm{n} \in \mathbb{N}_0$ *platí* $\mathrm{f}(\mathbf{n}) \mapsto^* \mathbf{m}$*, kde* $\mathrm{m} = 2^{\mathrm{n}}$*.*

Důkaz. Tvrzení

> Pro každé $\mathrm{n} \in \mathbb{N}_0$ platí $\mathrm{f}(\mathbf{n}) \mapsto^* \mathbf{m}$ kde $\mathrm{m} = 2^{\mathrm{n}}$

nelze dokázat indukcí vzhledem k n. Řešením je přeformulování dokazovaného tvrzení do *silnější* podoby, kterou již indukcí dokázat lze:

> Pro každé $\mathrm{n} \in \mathbb{N}_0$ platí $\mathrm{f}(\mathbf{n}) \mapsto^* \mathbf{m}$ a $\mathrm{h}(\mathbf{n}) \mapsto^* \mathbf{m}$, kde $\mathrm{m} = 2^{\mathrm{n}}$.

Toto tvrzení již poměrně snadno dokážeme indukcí vzhledem k n:

- $n = 0$. Platí

$$f(\mathbf{0}) \quad \mapsto \quad \textbf{if } \mathbf{0} \textbf{ then } h(\mathbf{0}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \mathbf{1}$$
$$h(\mathbf{0}) \quad \mapsto \quad \textbf{if } \mathbf{0} \textbf{ then } f(\mathbf{0}-\mathbf{1}) + h(\mathbf{0}-\mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \mathbf{1}$$

- Indukční krok. Nechť $n + 1 \equiv \mathbf{k}$. Platí

$$f(\mathbf{k}) \quad \mapsto \quad \textbf{if } \mathbf{k} \textbf{ then } h(\mathbf{k}) \textbf{ else } \mathbf{1} \quad \mapsto \quad h(\mathbf{k}) \quad \mapsto$$
$$\textbf{if } \mathbf{k} \textbf{ then } f(\mathbf{k}-\mathbf{1}) + h(\mathbf{k}-\mathbf{1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad f(\mathbf{k}-\mathbf{1}) + h(\mathbf{k}-\mathbf{1}) \quad \mapsto$$
$$f(\mathbf{w}) + h(\mathbf{w})$$

kde $\mathbf{w} \equiv k-1 = n$. Podle I.P. platí $f(\mathbf{w}) \mapsto^* \mathbf{m}$, kde $m = 2^n$. Zároveň také (naše „zesílení“) platí i $h(\mathbf{w}) \mapsto^* \mathbf{m}$, a proto

$$f(\mathbf{w}) + h(\mathbf{w}) \quad \mapsto^* \quad \mathbf{m} + \mathbf{m} \quad \mapsto \quad \mathbf{2m} \quad \mapsto \quad \mathbf{q},$$

kde $q = m + m = 2 \cdot 2^n = 2^{n+1} = 2^k$. Proto $f(\mathbf{k}) \mapsto \mathbf{q}$ a první část našeho tvrzení platí i pro $n + 1 = k$.

Podobně je třeba ještě dokončit druhou část tvrzení

$$\begin{array}{lllll} \mathtt{h}(\mathbf{k}) & \mapsto & \textbf{if } \mathbf{k} \textbf{ then } \mathtt{f}(\mathbf{k}-\mathbf{1}) + \mathtt{h}(\mathbf{k}-\mathbf{1}) \textbf{ else } \mathbf{1} & \mapsto \\ \mathtt{f}(\mathbf{k}-\mathbf{1}) + \mathtt{h}(\mathbf{k}-\mathbf{1}) & \mapsto & \mathtt{f}(\mathbf{w}) + \mathtt{h}(\mathbf{w}) & \end{array}$$

kde $\mathbf{w} \equiv \mathtt{k}-1 = \mathtt{n}$. Podle I.P. platí $\mathtt{f}(\mathbf{w}) \mapsto^* \mathbf{m}$, kde $\mathtt{m} = 2^{\mathtt{n}}$, a také $\mathtt{h}(\mathbf{w}) \mapsto^* \mathbf{m}$, a proto

$$\mathtt{f}(\mathbf{w}) + \mathtt{h}(\mathbf{w}) \quad \mapsto^* \quad \mathbf{m} + \mathbf{m} \quad \mapsto \quad \mathbf{q},$$

kde $\mathtt{q} = \mathtt{m} + \mathtt{m} = 2 \cdot 2^{\mathtt{n}} = 2^{\mathtt{n}+1} = 2^{\mathtt{k}}$. Proto $\mathtt{h}(\mathbf{k}) \mapsto \mathbf{q}$ a i druhá část našeho tvrzení platí pro $\mathtt{n}+1 = \mathtt{k}$.

□

Euklidův algoritmus

Příklad 20. Věta: *Uvažme deklaraci Δ obsahující pouze rovnici*

$$g(x, y) = \textbf{\textit{if}}\ x - y\ \textbf{\textit{then}}\ g(x - y, y)\ \textbf{\textit{else}}\ (\textbf{\textit{if}}\ y - x\ \textbf{\textit{then}}\ g(x, y - x)\ \textbf{\textit{else}}\ x)$$

Pak pro každé nenulové $m, n \in \mathbb{N}$ platí $g(\mathbf{m}, \mathbf{n}) \mapsto^* \mathbf{z}$, kde z je největší společný dělitel čísel m, n.

Důkaz. Indukcí k $i = m + n$.
(Tj. dokazujeme následující tvrzení: Pro každé $i \geq 2$ platí, že jestliže $i = m + n$, kde $m, n \in \mathbb{N}$, pak z je největší společný dělitel čísel m, n.)

- $i = 2$. Pak $m, n = 1$ a platí

$$\begin{aligned} g(\mathbf{1}, \mathbf{1}) \quad &\mapsto \quad \textbf{if } \mathbf{1 - 1} \textbf{ then } g(\mathbf{1 - 1}, \mathbf{1}) \textbf{ else } (\textbf{if } \mathbf{1 - 1} \textbf{ then } g(\mathbf{1}, \mathbf{1 - 1}) \textbf{ else } \mathbf{1}) \quad \mapsto \\ &\textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{1 - 1}, \mathbf{1}) \textbf{ else } (\textbf{if } \mathbf{1 - 1} \textbf{ then } g(\mathbf{1}, \mathbf{1 - 1}) \textbf{ else } \mathbf{1}) \quad \mapsto \\ &\textbf{if } \mathbf{1 - 1} \textbf{ then } g(\mathbf{1}, \mathbf{1 - 1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \textbf{if } \mathbf{0} \textbf{ then } g(\mathbf{1}, \mathbf{1 - 1}) \textbf{ else } \mathbf{1} \quad \mapsto \quad \mathbf{1} \end{aligned}$$

- Indukční krok. Nechť $\mathrm{i}+1 = \mathrm{m}+\mathrm{n}$ kde $\mathrm{m},\mathrm{n} \in \mathbb{N}$. Jsou tři možnosti:
  - $\mathrm{m} = \mathrm{n}$. Pak

    $\mathrm{g}(\mathbf{m},\mathbf{n}) \quad \mapsto \quad$ **if** $\mathbf{m}-\mathbf{n}$ **then** $\mathrm{g}(\mathbf{m}-\mathbf{n},\mathbf{n})$ **else** (**if** $\mathbf{n}-\mathbf{m}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m}$) $\quad \mapsto$
    if $\mathbf{0}$ **then** $\mathrm{g}(\mathbf{m}-\mathbf{n},\mathbf{n})$ **else** (**if** $\mathbf{n}-\mathbf{m}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m}$) $\quad \mapsto$
    if $\mathbf{n}-\mathbf{m}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m} \quad \mapsto \quad$ **if** $\mathbf{0}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m} \quad \mapsto \quad \mathbf{m}$

  - $\mathrm{m} < \mathrm{n}$. Pak

    $\mathrm{g}(\mathbf{m},\mathbf{n}) \quad \mapsto \quad$ **if** $\mathbf{m}-\mathbf{n}$ **then** $\mathrm{g}(\mathbf{m}-\mathbf{n},\mathbf{n})$ **else** (**if** $\mathbf{n}-\mathbf{m}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m}$) $\quad \mapsto$
    if $\mathbf{0}$ **then** $\mathrm{g}(\mathbf{m}-\mathbf{n},\mathbf{n})$ **else** (**if** $\mathbf{n}-\mathbf{m}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m}$) $\quad \mapsto$
    if $\mathbf{n}-\mathbf{m}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m} \quad \mapsto \quad$ **if** $\mathbf{z}$ **then** $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m})$ **else** $\mathbf{m} \quad \mapsto$
    $\mathrm{g}(\mathbf{m},\mathbf{n}-\mathbf{m}) \quad \mapsto \quad \mathrm{g}(\mathbf{m},\mathbf{k})$

    kde $\mathbf{k} \equiv \mathrm{n}-\mathrm{m}$. Platí $\mathrm{m}+\mathrm{k} = \mathrm{m}+(\mathrm{n}-\mathrm{m}) = \mathrm{n} \leq \mathrm{i}$, takže podle I.P. také platí $\mathrm{g}(\mathbf{m},\mathbf{k}) \mapsto^* \mathbf{z}$, kde z je největší společný dělitel čísel m a $\mathrm{n}-\mathrm{m}$. Ověříme, že z je největší společný dělitel čísel m a n.

    – Jelikož číslo z dělí čísla m a $\mathrm{n}-\mathrm{m}$, dělí i jejich součet $(\mathrm{n}-\mathrm{m})+\mathrm{m} = \mathrm{n}$. Celkem z je společným dělitelem m a n.

    – Buď d nějaký společný dělitel čísel m a n. Pak d dělí také rozdíl $\mathrm{n}-\mathrm{m}$. Tedy d je společný dělitel čísel m a $\mathrm{n}-\mathrm{m}$. Jelikož z je *největší* společný dělitel čísel m a $\mathrm{n}-\mathrm{m}$, platí $\mathrm{d} \leq z$.

$*$ $\mathtt{m} > \mathtt{n}$. Pak

$$\mathtt{g}(\mathbf{m}, \mathbf{n}) \quad \mapsto^* \quad \mathtt{g}(\mathbf{m} - \mathbf{n}, \mathbf{n}) \quad \mapsto \quad \mathtt{g}(\mathbf{k}, \mathbf{n})$$

kde $\mathbf{k} \equiv \mathtt{m} - \mathtt{n}$. Podle I.P. platí $\mathtt{g}(\mathbf{k}, \mathbf{n}) \mapsto^* \mathbf{z}$, kde z je největší společný dělitel čísel $\mathtt{m} - \mathtt{n}$ a $\mathtt{n}$. Podobně jako výše ověříme, že z je největší společný dělitel čísel $\mathtt{m}$ a $\mathtt{n}$.

□

Jak byste výše uvedený zápis Euklidova algoritmu vylepšili, aby správně „počítal“ největšího společného dělitele i v případech, že $\mathtt{m} = 0$ nebo $\mathtt{n} = 0$? Co v takových případech selže při současném zápise?

Ještě příklad

Vraťme se k předchozí ukázce (velmi „hutného“) C kódu, který inkrementuje dekadický zápis čísla $\mathtt{D}$ uloženého po číslicích v poli.

```
for (i=0; i? D[i-1]==0: 1; i++)
  D[i] = (D[i]+1)%10;
```

Zapišme tento kód naší formální deklarací.

- Jelikož nejsou k dispozici proměnné typu pole, „pomůžeme si“ funkcí $\nu(\mathtt{i})$ udávající $\mathtt{i}$-tou dekadickou číslici výstupu.
- Obdobně pro zadání číslic vstupu použijeme funkční deklarace $z(\mathtt{i}) = \ldots$.
- Cyklus `for` nahradíme rekurzí (běžný postup).
- Nakonec „trikově“ nahradíme řídící podmínku našeho cyklu zavedením nové funkce $\mathtt{p}(\mathtt{i})$, která vyjadřuje „přenos“ do $\mathtt{i}$-tého řádu.
  ($\mathtt{p}(\mathtt{i}) \in \{0, 1\}$, na počátku $\mathtt{p}(0) = 1$.)

- Dohodneme se, že číslice jsou indexovány od nejnižšího řádu $\mathrm{i} = 0, 1, 2, \ldots$. Celá (formální) deklarace Δ bude vypadat následovně.

Příklad 21. Věta: *Uvažme deklaraci Δ obsahující tyto rovnice:*

$$\nu(\mathrm{i}) = \big(z(\mathrm{i}) + \mathrm{p}(\mathrm{i})\big)\%10$$

$$\mathrm{p}(\mathrm{i}) = \textbf{\textit{if}}\ \mathrm{i}\ \textbf{\textit{then}}\ \big(\textbf{\textit{if}}\ z(\mathrm{i}-1) + \mathrm{p}(\mathrm{i}-1) - 9\ \textbf{\textit{then}}\ 1\ \textbf{\textit{else}}\ 0\big)\ \textbf{\textit{else}}\ 1$$

Pak pro každé $\mathrm{i} \in \mathbb{N}_0$ platí, že $\nu(\mathrm{i})$ udává dekadickou číslici i-tého řádu zprava čísla $\mathrm{m}+1$, kde m má dekadický zápis po číslicích $\big(z(\mathrm{i})\, z(\mathrm{i}-1) \ldots z(1)\, z(0)\big)_{10}$.

Dokažte si toto sami za domácí úkol (diskutujte na IS).
Je potřeba použít matematickou indukci se zesíleným předpokladem, který se bude vhodně vyjadřovat i o významu hodnoty $\mathrm{p}(\mathrm{i})$ („přenos“).
Pochopitelně je třeba pro úplnou správnost řešení rozepsat operaci „modulo“ % pomocí povolených aritmetických operací, což si také za úkol vyzkoušejte.