

Bonusová domáca úloha č. 2

(Termín odovzdania úlohy: **do pondelka 3. apríla 2023, 14:50**, t. j. do začiatku ôsmych cvičení.)

Nech S je množina a $R \subseteq S \times S$. *Relácia ekvivalencie na S generovaná reláciou R* je najmenšia relácia ekvivalencie $[R]$ na S taká, že $R \subseteq [R]$. Dá sa ukázať, že reláciu $[R]$ možno charakterizovať ako prienik všetkých relácií ekvivalencie na S obsahujúcich R . Pri jej výpočte možno postupovať nasledovne:

1. Nájdeme *symetrický uzáver* $S = R \cup R^{-1}$ relácie R – do relácie R teda pridáme všetky dvojice $(y, x) \in S \times S$ také, že $(x, y) \in R$.
2. Vypočítame reflexívno-tranzitívny uzáver S^* relácie S , ktorý je rovný hľadanej relácii $[R]$.

Cieľom tejto úlohy je napísať triedu reprezentujúcu reláciu ekvivalencie na konečnej množine $\{0, \dots, n-1\}$ pre nejaké prirodzené číslo n . Inštanciu tejto triedy bude možné vytvoriť zadaním čísla n a generujúcej relácie – čiže nejakej množiny dvojíc prvkov $\{0, \dots, n-1\}$. Do veľkej miery budeme využívať fakt, že každá binárna relácia R na $\{0, \dots, n-1\}$ zodpovedá orientovanému grafu na množine vrcholov $\{0, \dots, n-1\}$, v ktorom vedie hrana z vrcholu u do vrcholu v práve vtedy, keď $(u, v) \in R$. Každá symetrická relácia – a teda aj každá relácia ekvivalencie – takto zodpovedá neorientovanému grafu.

Priložený ZIP archív obsahuje balík `graphs` a v ňom triedy pre grafy z prednášky, ako aj kosru triedy `EquivalenceRelation` reprezentujúcej reláciu ekvivalencie. V tejto triede doprogramujte:

- Konštruktor, ktorý ako argumenty vezme prirodzené číslo n a množinu `generators` orientovaných hrán na množine vrcholov $\{0, \dots, n-1\}$. Na základe týchto argumentov konštruktor vytvorí inštanciu reprezentujúcu reláciu ekvivalencie na množine $\{0, \dots, n-1\}$ generovanú množinou dvojíc zodpovedajúcou množine hrán z argumentu – každá hrana z u do v sa teda interpretuje iba ako dvojica (u, v) a z danej množiny takýchto dvojíc sa nageneruje relácia ekvivalencie.

Prípadná neskoršia zmena množiny použitej ako argument tohto konštruktora by nijako nemala ovplyvniť vnútorný stav ním vytvorenej inštancie.

- Metódu `areEquivalent`, ktorá ako argumenty vezme dve čísla a, b z množiny $\{0, \dots, n-1\}$ pre n z argumentu konštruktora a na výstupe vráti logickú hodnotu `true` práve vtedy, keď sú čísla a, b vzhľadom na reláciu reprezentovanú príslušnou inštanciou ekvivalentné.
- Metódu `inducedPartition` bez argumentov, ktorá na výstupe vráti *rozklad* množiny $\{0, \dots, n-1\}$ indukovaný reláciou ekvivalencie reprezentovanou príslušnou inštanciou. Výstupným typom teda bude *množina množín* celých čísel (presnejšie pôjde o množinu množín prvkov $\{0, \dots, n-1\}$).

Pripomeňme si definíciu rozkladu indukovaného reláciou ekvivalencie R na množine S : ak pre $x \in S$ označíme triedu ekvivalencie $[x]_R = \{y \in S \mid (x, y) \in R\}$, je rozklad množiny S indukovaný reláciou R daný ako

$$\{[x]_R \mid x \in S\}$$

(ak pre nejaké $x, y \in S$ platí $[x]_R = [y]_R$, je táto trieda, samozrejme, prvkom rozkladu iba raz).

Množinu získanú ako výstup tejto metódy by nemalo byť možné použiť na zmenu vnútorného stavu inštancie, pre ktorú bola táto metóda volaná.

- Metódu `asGraph` bez argumentov, ktorá na výstupe vráti reprezentovanú reláciu ekvivalencie v podobe neorientovaného grafu. Tento graf teda bude mať množinu vrcholov $\{0, \dots, n-1\}$ pre n z argumentu konštruktora a medzi vrcholmi u, v bude viesť hrana práve vtedy, keď sú u a v vzhľadom na reprezentovanú reláciu ekvivalentné.

V prípade potreby môžete v triede `EquivalenceRelation` definovať aj ďalšie pomocné metódy a premenné. Môžete predpokladať korektnosť argumentov metód a konštruktora.

Na testovač odovzdávajte iba súbor `EquivalenceRelation.java` obsahujúci kód vami doplnenej triedy.

Príklad. Predpokladajme, že inštanciu triedy `EquivalenceRelation` vytvoríme volaním jej konštruktora pre $n = 6$ a pre množinu pozostávajúcu z orientovaných hrán $(0, 1)$, $(1, 2)$ a $(4, 5)$.

Metóda `areEquivalent` potom napríklad:

- pre $a = 0$ a $b = 0$ vráti `true`;
- pre $a = 0$ a $b = 2$ vráti `true`;
- pre $a = 0$ a $b = 3$ vráti `false`;
- pre $a = 0$ a $b = 5$ vráti `false`;
- pre $a = 2$ a $b = 0$ vráti `true`;
- pre $a = 2$ a $b = 3$ vráti `false`;
- pre $a = 3$ a $b = 4$ vráti `false`;
- pre $a = 4$ a $b = 4$ vráti `true`;
- pre $a = 4$ a $b = 5$ vráti `true`;
- pre $a = 5$ a $b = 4$ vráti `true`.

Metóda `inducedPartition` vráti množinu množín s textovou reprezentáciou

`[[0, 1, 2], [3], [4, 5]]`

(poradie množín a prvkov jednotlivých množín v textovej reprezentácii môže byť aj odlišné.)

Metóda `asGraph` napokon vráti neorientovaný graf znázornený na nasledujúcom obrázku.

