Einstaklingsverkefni 5 (e. Individual Assignment 5)

T-117-STR1, Strjál stærðfræði I, 2024-3

Reykjavik University - Department of Computer Science, Menntavegi 1, IS-101 Reykjavík, Iceland

Kennari: Harpa Guðjónsdóttir

harpagud@ru.is

Skilafrestur (e. Deadline): 22.10.2024

Hér er Einstaklingsverkefni 5. Skilafrestur er þriðjudaginn 22.október 2024 kl. 23:59. Þetta eru ein af 5 einstaklingsskilum. Þau gilda alls 20% af lokaeinkunn, en lægstu einkunn er sleppt. Nemendur skila lausnum í einni skrá ***relations.py*** á Gradescope.

Þetta verkefni er samblanda af stærðfræði og forritun. Markmið verkefnisins er að nemendur nái að átta sig á tengingunni milli stærðfræði og forritun með því að koma frá sér stærðfræðilegu verkefni með því að forrita lausnirnar.

English version:

("This is the fifth individual assignment. The deadline is Tuesday, October 22th 2024 at 23:59. This is one of 5 individual assignments. All in all, their weight is 20% of the final grade, but the lowest grade is dropped. Students hand in their solution in one file ***relations.py*** in Gradescope.

This assignment is a combination of math and programming. The goal of the project is for the students to understand the connection between mathematics and programming by completing a mathematical project by programming the solutions.")

Reglur varðandi skil og yfirferð: ("Submission and grading rules:")

- Eins og í öllum verkefnum þá er mikilvægt að hafa í huga reglur HR um nám og námsmat. Rifjið upp reglurnar áður en þið hefjist handa við að vinna þetta verkefni, sjá nánar í Canvas undir Reglur HR um nám og námsmat sem þið finnið í Modules/Um áfangann/Verkefnaskil/Reglur HR um nám og námsmat. ("As in all assignments, it is necessary to keep in mind the RU rules on study and assesment. Please review the rules before you start working on this assignment, see details in Canvas under HR Rules for Learning and Assessment which you can find in Modules/Course Information/Assignment hand-ins/RU rules on study and assesment.")
- Það eru engin takmörk á því hversu oft þú getur skilað fram að skilafrest. ("There is no limit to how many times you can submit until the deadline.")
- Kóðinn þinn verður keyrður í gegnum sjálfvirka yfirferð, helmingur prófanna (e. tests) í sjálfvirku yfirferðinni (e. autograder) mun veita endurgjöf um kóðann. Fyrir hinn helming prófanna þá munuð þið einungis sjá ef prófið mistókst (e. test failed) en ekki nánari upplýsingar um inntak (e. input) og vænt úttak (e. expected output). Nýtið þá endurgjöf sem þið fáið. ("Your code will be run through an autograder, half of the tests in the autograder will provide feedback on the code. For the other half of the tests, you will only see that if a test failed but not detailed information about input (e.input) and expected output (e. expected output). Use the feedback you receive.")
- Ekki er leyfilegt nota nein python libraries eða utanaðkomandi pakka (e. external packages). ("It is not allowed to use any python libraries or external packages.")
- Lausn við öllum dæmum skal skilað í einni python skrá, ***relations.py***. Þið fáið snið mát (e. template) að skránni, notið skránna sem grunn fyrir ykkar lausn. ("Solutions to all problems must be submitted in one python file, ***relations.py***. You get a template file, use the file as a base for your solution.")
- Fyrir sérhvert dæmi þá gildir réttmæti kóðans 50% stiga og rökstuðningur 50% stiga. Rökstyðjið lausnirnar ykkar með því að lýsa kóðanum ykkar (e. comment your code). Rökstuðningurinn þarf að lýsa því sem þið gerið í kóðanum og afhverju. (" For each problem, the correctness of the your code counts for 50% of the points and your arguments 50% points. Give arguments for your solutions by commenting on your code. The argument you provide in the comments must be such that you are describing what you do in the code and why.")
- Það eru 10 bónusstig í boði, þau fást með því að fá fullt hús stiga úr sjálfvirku yfirferðinni (e. autograder). ("There are 10 bonus points available, they are obtained by getting a full house of points from the autograder.")

Skiladæmi (e. Hand-in problems) :

Note: English version is below the Icelandic version in each problem.

Dæmi 1 (e. Problem 1) (8%+8%+8%+8%)

a) Útfærið fall sem skilar hvort vensl séu spegilvirk.

- **Heiti falls og inntaks:** is_reflexive(defined_set, relation_on_set)

- **Inntak:** Fyrra inntakið (defined_set) er listi sem táknar mengið sem venslin eru á. Seinna inntakið (relation_on_set) er listi af túplum sem tákna vensli á mengið.
- **Úttak:** Boolean gildi, True ef venslin (relation_on_set) eru spegilvirk á mengið (defined_set), en annars False.
- **Dæmi:**
  - is_reflexive ([1, 2, 3, 4], [(1, 1), (1, 3), (2, 2), (2, 3), (3, 3), (4, 4)]) ætti að skila boolean gildinu True.
  - is_reflexive (['Alice', 'Bob', 'David', 'Eve'], [('Alice','Bob'),('Bob','Alice')]) ætti að skila boolean gildinu False.

b) Útfærið fall sem skilar hvort vensl séu samhverf.

- **Heiti falls og inntaks:** is_symmetric(relation_on_set)
- **Inntak:** Inntakið (relation_on_set) er listi af túplum sem tákna vensl.
- **Úttak:** Boolean gildi, True ef venslin (relation_on_set) eru samhverf, en annars False.
- **Dæmi:**
  - is_symmetric([(1, 1), (1, 3), (2, 2), (2, 3), (3, 3), (4, 4)]) ætti að skila boolean gildinu False.
  - is_symmetric([('Alice','Bob'),('Bob','Alice')]) ætti að skila boolean gildinu True.

c) Útfærið fall sem skilar hvort vensl séu andsamhverf.

- **Heiti falls og inntaks:** is_antisymmetric(relation_on_set)
- **Inntak:** Inntakið (relation_on_set) er listi af túplum sem tákna vensli á mengið.
- **Úttak:** Boolean gildi, True ef venslin (relation_on_set) eru andsamhverf, en annars False.
- **Dæmi:**
  - is_antisymmetric([(1, 1), (2, 2), (1, 2), (2, 3), (1, 3)]) ætti að skila boolean gildinu True.
  - is_antisymmetric([('Alice','Bob'),('Bob','Alice')]) ætti að skila boolean gildinu False.

d) Útfærið fall sem skilar hvort vensl séu gegnvirk.

- **Heiti falls og inntaks:** is_transitive (relation_on_set)
- **Inntak:** Inntakið (relation_on_set) er listi af túplum sem tákna vensli á mengið.
- **Úttak:** Boolean gildi, True ef venslin (relation_on_set) eru gegnvirk, en annars False.
- **Dæmi:**
  - is_transitive ([(1, 1), (2, 2)]) ætti að skila boolean gildinu True.
  - is_transitive ([('Alice','Bob'),('Bob','Alice')]) ætti að skila boolean gildinu False.

English version:

a) ("Implement a function that returns whether a relation is reflexive.

- **Name of function and input:** is_reflexive (defined_set, relation_on_set)
- **Input:** First input (defined_set) is a list representing the set on which the relations exist. Second input (relation_on_set) is a list of tuples representing a relation on the set.
- **Output:** Boolean value, True if the relation (relation_on_set) is reflexive for the set (defined_set), otherwise False.
- **Examples:**
  - is_reflexive ([1, 2, 3, 4], [(1, 1), (1, 3), (2, 2), (2, 3), (3, 3), (4, 4)]) should return the boolean value True.
  - is_reflexive (['Alice', 'Bob', 'David', 'Eve'], [('Alice','Bob'),('Bob','Alice')]) should return the boolean value False.

b) Implement a function that returns whether a relation is symmetric.

- **Name of function and input:** is_symmetric(relation_on_set)
- **Input:** The input (relation_on_set) is a list of tuples representing a relation.
- **Output:** Boolean value, True if the relation (relation_on_set) is symmetric, otherwise False.
- **Examples:**
  - is_symmetric([(1, 1), (1, 3), (2, 2), (2, 3), (3, 3), (4, 4)]) should return the boolean value False.
  - is_symmetric([('Alice','Bob'),('Bob','Alice')]) should return the boolean value True.

c) Implement a function that returns whether a relation is antisymmetric.

- **Name of function and input:** is_antisymmetric(relation_on_set)
- **Input:** The input (relation_on_set) is a list of tuples representing a relation.
- **Output:** Boolean value, True if the relation (relation_on_set) is antisymmetric, otherwise False.
- **Examples:**
  - is_antisymmetric([(1, 1), (2, 2), (1, 2), (2, 3), (1, 3)]) should return the boolean value True.
  - is_antisymmetric([('Alice','Bob'),('Bob','Alice')]) should return the boolean value False.

d) Implement a function that returns whether a relation is transitive.

- **Name of function and input:** is_transitive (relation_on_set)
- **Input:** The input (relation_on_set) is a list of tuples representing a relation.
- **Output:** Boolean value, True if the relation (relation_on_set) is transitive, otherwise False.
- **Examples:**
  - is_transitive ([(1, 1), (2, 2)]) should return the boolean value True.
  - is_transitive ([('Alice','Bob'),('Bob','Alice')]) should return the boolean value False.")

Dæmi 2 (e. Problem 2) (18%)

Útfærið fall sem skila hvort vensl séu jafngildisvensl. Nýtið ykkur föllinn ykkar úr dæmi 1.

- **Heiti falls og inntaks:** is_equivalence_relation(defined_set, relation_on_set)
- **Inntak:** Fyrra inntakið (defined_set) er listi sem táknar mengið sem venslin eru á. Seinna inntakið (relation_on_set) er listi af túplum sem tákna vensli á mengið.
- **Úttak:** Boolean gildi, True ef venslin (relation_on_set) eru jafngildisvensl á mengið (defined_set), en annars False.
- **Dæmi:**
  - is_equivalence_relation($[1, 2, 3, 4]$, $[(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 1)]$) ætti að skila boolean gildinu True.
  - is_equivalence_relation((['Alice', 'Bob', 'David', 'Eve'], [('Alice','Bob'),('Bob','Alice')]) ætti að skila boolean gildinu False.

English version:

("Implement a function that return whether a relation is an equivalence relations. Use the functions from example 1 in your implementation.

- **Name of function and input:** is_equivalence_relation(defined_set, relation_on_set)
- **Input:** First input (defined_set) is a list representing the set on which the relations exist. Second input (relation_on_set) is a list of tuples representing a relation on the set.
- **Output:** Boolean value, True if the relation (relation_on_set) is an equivalence relation for the set (defined_set), otherwise False.
- **Examples:**
  - is_equivalence_relation($[1, 2, 3, 4]$, $[(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 1)]$) should return the boolean value True.
  - is_equivalence_relation((['Alice', 'Bob', 'David', 'Eve'], [('Alice','Bob'),('Bob','Alice')]) should return the boolean value False.")

Dæmi 3 (e. Problem 3) (20%)

Útfærið fall sem tekur inn tvö vensl, R og S og skilar samsettu venslunum $S \circ R$

- **Heiti falls og inntaks:** composite_relations(relation_R, relation_S)
- **Inntak:** Fyrra inntakið (relation_R) er listi af túplum sem táknar venslin R. Seinna inntakið (relation_S) er listi af túplum sem táknar venslin S.
- **Úttak:** Listi af túplum sem tákna samsettu venslin $S \circ R$.
- **Dæmi:**
  - composite_relations($[(2, 1), (3, 1), (3, 2), (4, 2)]$, $[(1, 2), (1, 3), (2, 3), (2, 4), (3, 1)]$) ætti að skila $[(2, 2), (2, 3), (3, 2), (3, 3), (3, 4), (4, 3), (4, 4)]$.
  - composite_relations([('b','c'),('a','a'),('a','c'),('c','b')], [('b','a'),('a','c')]) ætti að skila [('a', 'c'), ('c', 'a')].

English version:

("Implement a function that takes two relations, R and S, and returns the composite relations $S \circ R$

- **Name of function and input:** composite_relations(relation_R, relation_S)
- **Input:** First input (relation_R) is a list of tuples represeting the relation R. Second input (relation_S) is a list of tuples representing the relation S.
- **Output:** List of tuples representing the composite relations $S \circ R$.
- **Example:**
  - composite_relations([(2, 1), (3, 1), (3, 2), (4, 2)], [(1, 2), (1, 3), (2, 3), (2, 4), (3, 1)]) should return [(2, 2), (2, 3), (3, 2), (3, 3), (3, 4), (4, 3), (4, 4)].
  - composite_relations([('b','c'),('a','a'),('a','c'),('c','b')], [('b','a'),('a','c')]) should return [('a', 'c'), ('c', 'a')].")

Dæmi 4 (e. Problem 4) (6%+6%+6%+6%+6%)

Látum R vera vensl á mengið $A = \{1, 2, 3, 4, \ldots, n\}$. Útfærið fimm föll sem taka inn venslin A, sem telja og skila fjölda ása í fylkinu M_R ef R eru venslin

a) $\{(a, b) \mid a = 0\}$

- **Heiti falls og inntaks:** aces_in_relation_a(A)
- **Inntak:** Inntakið (A) er listi sem táknar mengið A.
- **Úttak:** Heiltala (e. integer) sem táknar fjölda ása í fylkinu M_R
- **Dæmi:**
  - aces_in_relation_a([1, 2, 3, 4]), vænt úttak verður ekki gefið upp hér að öðru leiti en að fallið á að skila heiltölu (e. integer).

b) $\{(a, b) \mid a = b + 1\}$

- **Heiti falls og inntaks:** aces_in_relation_b(A)
- **Inntak:** Inntakið (A) er listi sem táknar mengið A.
- **Úttak:** Heiltala (e. integer) sem táknar fjölda ása í fylkinu M_R
- **Dæmi:**
  - aces_in_relation_b([1, 2, 3, 4]), vænt úttak verður ekki gefið upp hér að öðru leiti en að fallið á að skila heiltölu (e. integer).

c) $\{(a, b) \mid a \geq b\}$

- **Heiti falls og inntaks:** aces_in_relation_c(A)
- **Inntak:** Inntakið (A) er listi sem táknar mengið A.
- **Úttak:** Heiltala (e. integer) sem táknar fjölda ása í fylkinu M_R
- **Dæmi:**
  - aces_in_relation_c([1, 2, 3, 4]), vænt úttak verður ekki gefið upp hér að öðru leiti en að fallið á að skila heiltölu (e. integer).

d) $\{(a, b) \mid a + b = 1000\}$

- **Heiti falls og inntaks:** aces_in_relation_d(A)
- **Inntak:** Inntakið (A) er listi sem táknar mengið A..
- **Úttak:** Heiltala (e. integer) sem táknar fjölda ása í fylkinu M_R
- **Dæmi:**
  - aces_in_relation_d([1, 2, 3, 4]), vænt úttak verður ekki gefið upp hér að öðru leiti en að fallið á að skila heiltölu (e. integer).

e) $\{(a, b) \mid a + b \geq 1001\}$

- **Heiti falls og inntaks:** aces_in_relation_e(A)
- **Inntak:** Inntakið (A) er listi sem táknar mengið A.
- **Úttak:** Heiltala (e. integer) sem táknar fjölda ása í fylkinu M_R
- **Dæmi:**
  - aces_in_relation_e([1, 2, 3, 4]), vænt úttak verður ekki gefið upp hér að öðru leiti en að fallið á að skila heiltölu (e. integer).

English version:

Let R be the relation on the set $A = \{1, 2, 3, 4, \ldots, n\}$. Implement five function that take A as an input, counts the number of 1s in the matrix M_R if R is the relation

a) $\{(a, b) \mid a = 0\}$

- **Name of function and input:** aces_in_relation_a(A)
- **Input:** The input (A) is a list representing the set A.
- **Output** Integer representing the count of 1s in M_R
- **Example:**
  - aces_in_relation_a([1, 2, 3, 4]), expected outcome will not provided here other than that the function should return an integer.

b) $\{(a, b) \mid a = b + 1\}$

- **Name of function and input:** aces_in_relation_b(A)
- **Input:** The input (A) is a list representing the set A.
- **Output** Integer representing the count of 1s in M_R
- **Example:**
  - aces_in_relation_b([1, 2, 3, 4]), expected outcome will not provided here other than that the function should return an integer.

c) $\{(a, b) \mid a \geq b\}$

- **Name of function and input:** aces_in_relation_c(A)
- **Input:** The input (A) is a list representing the set A.
- **Output** Integer representing the count of 1s in M_R
- **Example:**

    * aces_in_relation_c([1, 2, 3, 4]), expected outcome will not provided here other than that the function should return an integer.

d) $\{(a, b) \mid a + b = 1000\}$

- **Name of function and input:** aces_in_relation_d(A)
- **Input:** The input (A) is a list representing the set A.
- **Output** Integer representing the count of 1s in M_R
- **Example:**
    * aces_in_relation_d([1, 2, 3, 4]), expected outcome will not provided here other than that the function should return an integer.

e) $\{(a, b) \mid a + b \geq 1001\}$

- **Name of function and input:** aces_in_relation_e(A)
- **Input:** The input (A) is a list representing the set A.
- **Output** Integer representing the count of 1s in M_R
- **Example:**
    * aces_in_relation_e([1, 2, 3, 4]), expected outcome will not provided here other than that the function should return an integer.