

Assignment 1

Due: Friday, 29.04.2016, 15:59 via Git

For help, contact alp-staff@lists.iai.uni-bonn.de (staff only) or
alp-course@lists.iai.uni-bonn.de (staff and participants).

Submit your implemented predicates as a file named “assignment01/solutions.pl” in the Git repository of your group.

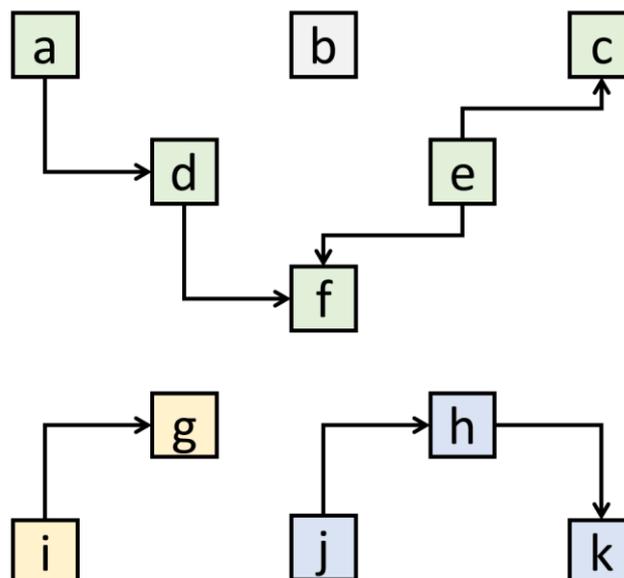
Add a file “assignment01/testRuns.txt” showing the console output of a session in which you test that **each** solution works for the provided input data and some queries that represent sensible test cases.

If no input data is provided in the text of the task, create some sensible input data.

If the input data is represented as facts, include them into the “solutions.pl” file and add suitable comments.

Task 1. Path Search (5 Points)

Write a predicate **path(Start, End)** that succeeds, if there is a path from Start to End in the graph whose arcs are represented by a set of **arc(From,To)** facts. Test your predicate on the following graph:



NOTE: The graph is directed (e.g. there is no path from a to e).

Task 2. *Friends* (9 Points)

Write a Prolog predicate that solves the following logic puzzle:

1. Tick, Trick and Track are friends.
2. One friend is 15, one 17, and one 18 years but we do not know who has which age.
3. One friend's last name is Chang.
4. Miss Yang is three years older than Tick.
5. The person whose last name is Thatcher is 17 years old.

Tip 1: *The condition that X is bigger by 3 than Y is written in Prolog as "X is Y+3".*

Tip 2: *Consider what the puzzle tells you about the three friends and think of a suitable term structure representing a person. Then represent each of the three friends by a person term with variables for the values that are unknown. Represent our little "world" of three friends by a list holding the three (incomplete) persons. Then use 'member(Person, List)' to search the list for a person that fulfills one of the hints given in the second to fifth sentence of the puzzle. If you do this for each hint and also consider tip 1 you have the complete predicate that solves the puzzle.*

Task 3. *Understanding JTransformer PEFs* (6 Points)

Go to https://sewiki.iai.uni-bonn.de/research/jtransformer/api/java/pefs/4.1/java_pef_overview. Read the documentation of the program element facts (PEFs) `compilationUnitT`, `classT`, `fieldT`, `methodT`, `callT`, `getFieldT`, `assignT`. For an explanation of the notation see <http://sewiki.iai.uni-bonn.de/research/jtransformer/api/notation>. For a general introduction to the representation of Java program elements in Prolog you might want to consult <http://sewiki.iai.uni-bonn.de/research/jtransformer/api/java/prologast>.

Then answer the following questions:

- a) Which is the argument position of the "members" argument in a `classT`?
- b) Which is the argument position of the "parent" argument in a `fieldT`?
- c) Is there any common structure that all of the above-mentioned PEFs (`classT`, `fieldT`, `methodT`, `callT`, `compilationUnitT`, `getFieldT`, `assignT`) share?
- d) Is there anything else that `getFieldT`, `assignT` have in common (but not the others)?
- e) Can you anticipate which other elements will share this additional structure?
- f) Can you guess why?