

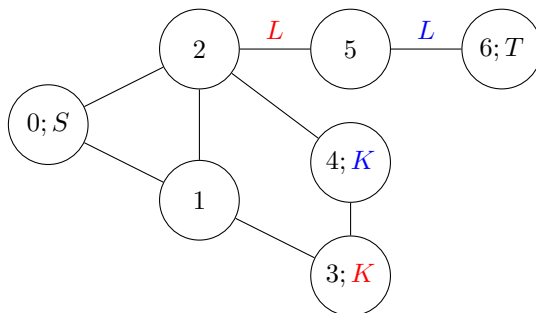
CS 571 Project 5 Instructions

Due Wednesday, Nov. 27

1 Search Problem

10 points

In this problem you will consider an agent whose goal is to get to some treasure. At each time step, the agent can move between rooms using doors arranged according to an undirected graph. However, some of the doors are locked, and the agent must acquire the appropriate key in order to pass through a locked door. For example, consider the scenario depicted in the following graph:



In this scenario, the agent starts in room 0, as indicated by the *S* annotation. The treasure is located in room 6, but the door from room 5 to 6 is locked with a blue lock – as indicated by a blue *L* – and the door from room 2 to 5 is locked with a red lock – as indicated by a red *L*. Thus in order to reach the treasure, the agent must first acquire both the red key – located in room 3 as indicated by the red *K* – and the blue key – located in room 4 as indicated by the blue *K*. Thus, the shortest path the agent can travel to reach the treasure in this scenario is through the rooms 0, 1, 3, 4, 2, 5, 6. This scenario is modeled in Prolog in the file `search/scenario1.pl`.

Your task in this question is to, given a specification of a problem such as the one in `search/scenario1.pl`, write a search procedure in Prolog to find the shortest path for the agent to reach the treasure. You should write – in the file `search/search.pl` – a predicate called `search(X)` that binds *X* to the shortest sequence of moves to lead the agent to the treasure. Examples of the output format are in `search/test1.pl` and `search/test2.pl`.

2 Parsing in Prolog

5 points

Write a parsing algorithm in Prolog for the following grammar, assuming the start nonterminal is *Lines*.

$$\begin{aligned} \textit{Lines} &\rightarrow \textit{Line} ; \textit{Lines} \mid \textit{Line} \\ \textit{Line} &\rightarrow \textit{Num} , \textit{Line} \mid \textit{Num} \\ \textit{Num} &\rightarrow \textit{Digit} \mid \textit{Digit Num} \\ \textit{Digit} &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

Please write your parser as the predicate `parse(X)` that is true if and only if the input list *X* of tokens is in the language specified by the grammar. Write this in the file `parsing/parse.pl`, which also contains examples. For the purpose of this question, you may not use the DCG operator `-->`.