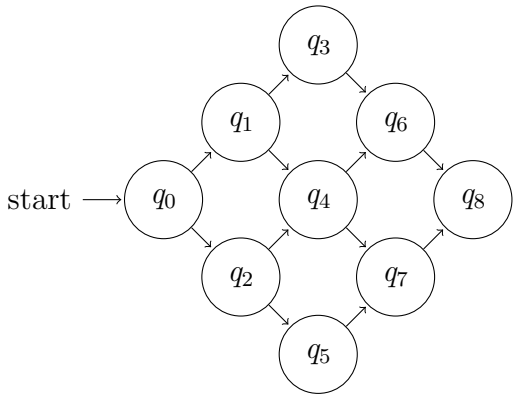
J. Hoenicke
A. Nutz

5.7.2017
Please hand in your solution until
12.7.2017, via email to
nutz@informatik.uni-freiburg.de.

Tutorials for "Formal methods for Java"
Exercise sheet 10

Exercise 1: Graph Search

Consider the following graph and heuristic function.



$$heuristic(v) = \begin{cases} 4 \text{ if } v = q_0 \\ 1 \text{ if } v = q_1 \\ 3 \text{ if } v = q_2 \\ 1 \text{ if } v = q_3 \\ 1 \text{ if } v = q_4 \\ 2 \text{ if } v = q_5 \\ 1 \text{ if } v = q_6 \\ 0 \text{ if } v = q_7 \\ 0 \text{ if } v = q_8 \end{cases}$$

(a) For each search technique covered in the lecture (DFS, BFS, Greedy, and A*) give the order in which the nodes are closed. If multiple decisions are possible at a step, choose one and make this choice explicit (i.e., state the choice you made).

(b) Is the heuristic admissible if node q_8 is the goal state? Justify your claim.

Exercise 2: Compiling into Java Bytecode

To compile a Java statement or expression into Java Bytecode, we can define the function *xlat*. This function takes as input a Java statement or a Java expression and returns a sequence of Java Bytecode instructions. To access local variables, we assume am auxiliary function *slot* that takes the name of a local variable and returns the slot of that variable on the stack. We illustrate these two functions on two simple examples:

$$\begin{aligned} xlat(e_1 \cdot e_2) \quad \equiv \quad & xlat(e_1) \\ & xlat(e_2) \\ & \texttt{"imul"} \end{aligned}$$

That is, we translate a multiplication into the sequence of instructions corresponding to the description First evaluate e_1, then e_2, then execute the instruction `imul`".

The next example simply evaluates the local integer variable `x`:

$$xlat(x) \equiv \texttt{"iload"}\ slot(x)$$

Here, the code produced by *xlat* states Load the value of the variable stored at the slot of `x`."

(a) Give the translation for the statement `x = e;` where `x` is a local integer variable and `e` some integer expression, i. e., specify what $xlat(x = e;)$ expands to.

(b) Give the translation for the statement `e1.f = e2;` where `e1` evaluates to some object of class C, f is an integer field, and `e2` is an integer expression, i. e., specify what $xlat(e_1.f = e_2;)$ expands to.