Frontier search (manage choices)

frontierSearch([Node|Rest]) :- goal(Node);
    (findall(Next, arc(Node,Next), Children),
     add2frontier(Children, Rest, NewFrontier),
     frontierSearch(NewFrontier)).
Frontier search (manage choices)

frontierSearch([Node|Rest]) :- goal(Node);
    (findall(Next, arc(Node,Next), Children),
     add2frontier(Children, Rest, NewFrontier),
     frontierSearch(NewFrontier)).

Depth first: append(Children, Rest, NewFrontier)
Breadth-first: append(Rest, Children, NewFrontier)
Frontier search (manage choices)

frontierSearch([Node|Rest]) :- goal(Node);
    (findall(Next, arc(Node,Next), Children),
     add2frontier(Children, Rest, NewFrontier),
     frontierSearch(NewFrontier)).

Depth first:  append(Children, Rest, NewFrontier)
Breadth-first: append(Rest, Children, NewFrontier)

For add2frontier(Children, Rest, NewFrontier), require

NewFrontier merges Children and Rest

where a list L is defined to *merge* lists L1 and L2 if
(a) every member of L is a member of L1 or L2
(b) every member of L1 or of L2 is a member of L.
Exercise (Prolog)

Suppose a positive integer Seed links nodes 1, 2, ... in two ways

\[
\text{arc}(N, M, \text{Seed}) :- M \text{ is } N \times \text{Seed}. \\
\text{arc}(N, M, \text{Seed}) :- M \text{ is } N \times \text{Seed} + 1.
\]

e.g. Seed = 3 gives arcs (1, 3), (1, 4), (3, 9), (3, 10) ...
Exercise (Prolog)

Suppose a positive integer Seed links nodes 1, 2, ... in two ways

\[
\text{arc}(N, M, \text{Seed}) :\neg M \text{ is } N \times \text{Seed}.
\]

\[
\text{arc}(N, M, \text{Seed}) :\neg M \text{ is } N \times \text{Seed} + 1.
\]

e.g. Seed=3 gives arcs (1,3), (1,4), (3,9), (3, 10)...

Goal nodes are multiples of a positive integer Target

\[
\text{goal}(N, \text{Target}) :\neg 0 \text{ is } N \mod \text{Target}.
\]

e.g. Target=13 gives goals 13, 26, 39...
Exercise (Prolog)

Suppose a positive integer Seed links nodes 1, 2, ... in two ways

\[
\text{arc}(N, M, \text{Seed}) :- M \text{ is } N \times \text{Seed}.
\]
\[
\text{arc}(N, M, \text{Seed}) :- M \text{ is } N \times \text{Seed} + 1.
\]

e.g. Seed = 3 gives arcs (1, 3), (1, 4), (3, 9), (3, 10) ...

Goal nodes are multiples of a positive integer Target

\[
\text{goal}(N, \text{Target}) :- 0 \text{ is } N \mod \text{Target}.
\]

e.g. Target = 13 gives goals 13, 26, 39 ...

Modify frontier search to define predicates

\[
\text{breadth1st}(+\text{Start}, \ ?\text{Found}, +\text{Seed}, +\text{Target})
\]
\[
\text{depth1st}(+\text{Start}, \ ?\text{Found}, +\text{Seed}, +\text{Target})
\]

that search breadth-first and depth-first respectively for a Target-goal node Found linked to Start by Seed-arcs.
Refining frontier search

For \text{add2frontier}(\text{Children}, \text{Rest}, \text{NewFrontier}), \text{ require }

\text{NewFrontier} \text{ merges } \text{Children} \text{ and } \text{Rest}

\text{and for } \text{NewFrontier} = [\text{Head}|\text{Tail}], \text{ ensure }

\text{Head is “no worse than” any in Tail.}
Refining frontier search

For `add2frontier(Children, Rest, NewFrontier)`, require

NewFrontier merges Children and Rest

and for `NewFrontier = [Head|Tail]`, ensure

Head is "no worse than" any in Tail.

What can it mean for Node1 to be *no worse than* Node2?

(A1) Node1 costs no more than Node2
Refining frontier search

For \texttt{add2frontier(Children, Rest, NewFrontier)}, require

\begin{align*}
\text{NewFrontier} & \text{ merges } \text{Children} \text{ and } \text{Rest} \\
\text{and for } \text{NewFrontier} & = [\text{Head}|\text{Tail}], \text{ ensure} \\
\text{Head is “no worse than” any in Tail.}
\end{align*}

What can it mean for \texttt{Node1} to be \textit{no worse than} \texttt{Node2} ?

(A1) \texttt{Node1} costs no more than \texttt{Node2}

(A2) \texttt{Node1} is deemed no further from a goal node than \texttt{Node2}
Refining frontier search

For \texttt{add2frontier(Children, Rest, NewFrontier)}, require

\begin{verbatim}
NewFrontier merges Children and Rest
\end{verbatim}

and for \texttt{NewFrontier = [Head|Tail]}, ensure

\begin{verbatim}
Head is “no worse than” any in Tail.
\end{verbatim}

What can it mean for \texttt{Node1} to be \textit{no worse than} \texttt{Node2}?

(A1) \texttt{Node1} costs no more than \texttt{Node2}

(A2) \texttt{Node1} is deemed no further from a goal node than \texttt{Node2}

(A3) some mix of (A1) and (A2)
Refining frontier search

For \text{add2frontier}(\text{Children}, \text{Rest}, \text{NewFrontier})$, require

\text{NewFrontier} \text{ merges Children and Rest}

and for \text{NewFrontier} = [\text{Head}|\text{Tail}]$, ensure

\text{Head is “no worse than” any in Tail.}

What can it mean for \text{Node1} to be \textit{no worse than} \text{Node2} ?

(A1) \text{Node1 costs no more than Node2}

\sim \text{ minimum cost search (=} \text{breadth-first if every arc costs 1})

(A2) \text{Node1 is deemed no further from a goal node than Node2}

(A3) \text{some mix of (A1) and (A2)}
Refining frontier search

For \texttt{add2frontier(Children, Rest, NewFrontier)}, require

\begin{quote}
NewFrontier merges Children and Rest
\end{quote}

and for NewFrontier = [Head|Tail], ensure

\begin{quote}
Head is “no worse than” any in Tail.
\end{quote}

What can it mean for Node1 to be \textit{no worse than} Node2?

(A1) Node1 costs no more than Node2
\begin{quote}
\leadsto \text{minimum cost search (} = \text{breadth-first if every arc costs 1)}
\end{quote}

(A2) Node1 is deemed no further from a goal node than Node2
\begin{quote}
\leadsto \text{best-first search (} = \text{depth-first for heuristic } \propto \text{ depth}^{-1}\text{)}
\end{quote}

(A3) some mix of (A1) and (A2)
For \texttt{add2frontier(Children, Rest, NewFrontier)}, require

\texttt{NewFrontier} merges \texttt{Children} and \texttt{Rest}

and for \texttt{NewFrontier} = \texttt{[Head|Tail]}, ensure

\texttt{Head} is “no worse than” any in \texttt{Tail}.

What can it mean for \texttt{Node1} to be \textit{no worse than} \texttt{Node2}?

(A1) \texttt{Node1} costs no more than \texttt{Node2}
\quad \leadsto \text{minimum cost search} (= \text{breadth-first if every arc costs 1})

(A2) \texttt{Node1} is deemed no further from a goal node than \texttt{Node2}
\quad \leadsto \text{best-first search} (= \text{depth-first for heuristic } \propto \text{depth}^{-1})

(A3) some mix of (A1) and (A2)
\quad \leadsto \text{A-star (next week)}
Arc costs (space, time, money, . . .)

\[ \text{cost}(w_a, n_t, q, n_s) = 1 + 2 + 2 = 5 \]

\[ \text{cost}(w_a, s_a, n_s) = 3 + 5 = 8 \]

\text{arc}(w_a, n_t, 1). \quad \text{arc}(n_t, q, 2).
\text{arc}(q, n_s, 2). \quad \text{arc}(w_a, s_a, 3).
\text{arc}(n_t, s_a, 2). \quad \text{arc}(s_a, q, 3).
\text{arc}(s_a, n_s, 5). \quad \text{arc}(s_a, v, 1).
\text{arc}(v, n_s, 1).
Arc costs (space, time, money, . . .)

arc(wa,nt,1). arc(nt,q,2).
arc(q,nsw,2). arc(wa,sa,3).
arc(nt,sa,2). arc(sa,q,3).
arc(sa,nsw,5). arc(sa,v,1).
arc(v,nsw,1).

\[
\text{cost}(wa,nt,q,nsw) = 1 + 2 + 2 = 5
\]

\[
\text{cost}(x_1, x_2, \ldots, x_{k+1}) := \sum_{i=1}^{k} \text{cost}(x_i, x_{i+1})
\]
Arc costs (space, time, money, . . .)

\[
\begin{align*}
\text{arc(wa,nt,1).} & \quad \text{arc(nt,q,2).} \\
\text{arc(q,nsw,2).} & \quad \text{arc(wa,sa,3).} \\
\text{arc(nt,sa,2).} & \quad \text{arc(sa,q,3).} \\
\text{arc(sa,nsw,5).} & \quad \text{arc(sa,v,1).} \\
\text{arc(v,nsw,1).} & \\
\end{align*}
\]

\[
\begin{align*}
\text{cost(wa,nt,q,nsw)} & = 1 + 2 + 2 = 5 \\
\text{cost}(x_1, x_2, \ldots, x_{k+1}) & := \sum_{i=1}^{k} \text{cost}(x_i, x_{i+1}) \\
\text{cost(wa,sa,nsw)} & = 3 + 5 = 8
\end{align*}
\]
Heuristics

\[ h(\text{Node}) = \text{estimate the minimum cost of} \]
\[ \text{a path from Node to a goal node} \]
Heuristics

\[ h(\text{Node}) = \text{estimate the minimum cost of} \]
\[ \text{a path from Node to a goal node} \]

**Examples**

- Fsm accept where node = [Q, String] and every arc costs 1
  \[ h([Q, \text{String}]) = \text{length(\text{String})} \]
Heuristics

\[ h(\text{Node}) = \text{estimate the minimum cost of} \]
\[ \text{a path from Node to a goal node} \]

**Examples**

- **Fsm accept** where node = \([Q, \text{String}]\) and every arc costs 1
  \[ h([Q, \text{String}]) = \text{length(String)} \]

- **Prolog search** where node = list of propositions to prove, and every arc costs 1
  \[ h(\text{List}) = \text{length(List)} \]
Heuristics

\[ h(\text{Node}) = \text{estimate the minimum cost of} \]
\[ \text{a path from Node to a goal node} \]

Examples

- FSM accept where node = \([Q,\text{String}]\) and every arc costs 1
  \[ h([Q,\text{String}]) = \text{length(}\text{String}\text{)} \]

- Prolog search where node = list of propositions to prove, and every arc costs 1
  \[ h(\text{List}) = \text{length(}\text{List}\text{)} \]

- Node = point on a Euclidean plane, cost = distance between nodes, goal is a point \(G\)
  \[ h(\text{Node}) = \text{straight-line distance to} \ G \]
Heuristics

\[ h(\text{Node}) = \text{estimate the minimum cost of} \]
\[ \text{a path from Node to a goal node} \]

**Examples**

- Fsm accept where node = \([Q,\text{String}]\) and every arc costs 1
  \[ h([Q,\text{String}]) = \text{length(String)} \]
- Prolog search where node = list of propositions to prove, and every arc costs 1
  \[ h(\text{List}) = \text{length(List)} \]
- Node = point on a Euclidean plane, cost = distance between nodes, goal is a point \(G\)
  \[ h(\text{Node}) = \text{straight-line distance to } G \]
- estimate assuming lots of arcs (simplifying the problem)
Best-first search

Form NewFrontier = [Head|Tail] such that

\[ h(\text{Head}) \leq h(\text{Node}) \] for every Node in Tail
Best-first search

Form NewFrontier = [Head|Tail] such that

\[ h(\text{Head}) \leq h(\text{Node}) \text{ for every Node in Tail} \]