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)

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

**Depth first:** \( \text{append}(\text{Children}, \text{Rest}, \text{NewFrontier}) \)

**Breadth-first:** \( \text{append}(\text{Rest}, \text{Children}, \text{NewFrontier}) \)
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

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

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

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

Goal nodes are multiples of a positive integer Target

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

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.
Exercise (Prolog)

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

\begin{verbatim}
arc(N,M,Seed) :- M is N*Seed.
arc(N,M,Seed) :- M is N*Seed +1.
\end{verbatim}

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

Goal nodes are multiples of a positive integer $Target$

\begin{verbatim}
goal(N,Target) :- 0 is N mod Target.
\end{verbatim}

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 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 ⇝ minimum cost search (= breadth-first if every arc costs 1)

(A2) Node1 is deemed no further from a goal node than Node2 ⇝ best-first search (= depth-first for heuristic ∝ depth − 1)

(A3) some mix of (A1) and (A2) ⇝ A-star (next lecture)
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 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

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

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

\texttt{NewFrontier merges Children and Rest}

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

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

(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 } \text{Children} \text{ and } \text{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) some mix of (A1) and (A2)
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
    \[\leadsto\] minimum cost search (= breadth-first if every arc costs 1)

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

(A3) some mix of (A1) and (A2)
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
    \(\leadsto\) minimum cost search (\(=\) breadth-first if every arc costs 1)

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

(A3) some mix of (A1) and (A2)
    \(\leadsto\) A-star (next lecture)
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).

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

\[
\text{cost}(x_1, x_2, \ldots , x_{k+1}) := k \sum_{i=1}^{k} \text{cost}(x_i, x_{i+1})
\]

\[
\text{cost}(\text{wa,sa,nsw}) = 3 + 5 = 8
\]
Arc costs (space, time, money, ...)

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

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

\[
\text{cost}(x_1, x_2, \ldots, x_{k+1}) := \sum_{i=1}^{k} \text{cost}(x_i, x_{i+1})
\]
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

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

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

**Examples**

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

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

**Examples**

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

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

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

**Examples**
- Fsm accept where node = \([Q, \text{String}]\) and every arc costs 1
  \[ h([Q, \text{String}]) = \text{length(\text{String})} \]
- Prolog search where node = list of propositions to prove, and every arc costs 1
  \[ h(\text{List}) = \text{length(\text{List})} \]
- Node = point on a Euclidean plane, cost = distance between nodes, goal is a point \(G\)
  \[ h(\text{Node}) = \text{straight-line distance to } G \]
Heuristics

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

**Examples**

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

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

- 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

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

Form $\text{NewFrontier} = \{\text{Head}|\text{Tail}\}$ such that

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