Searching

- Often we are not given an algorithm to solve a problem, but only a specification of what is a solution — we have to search for a solution.

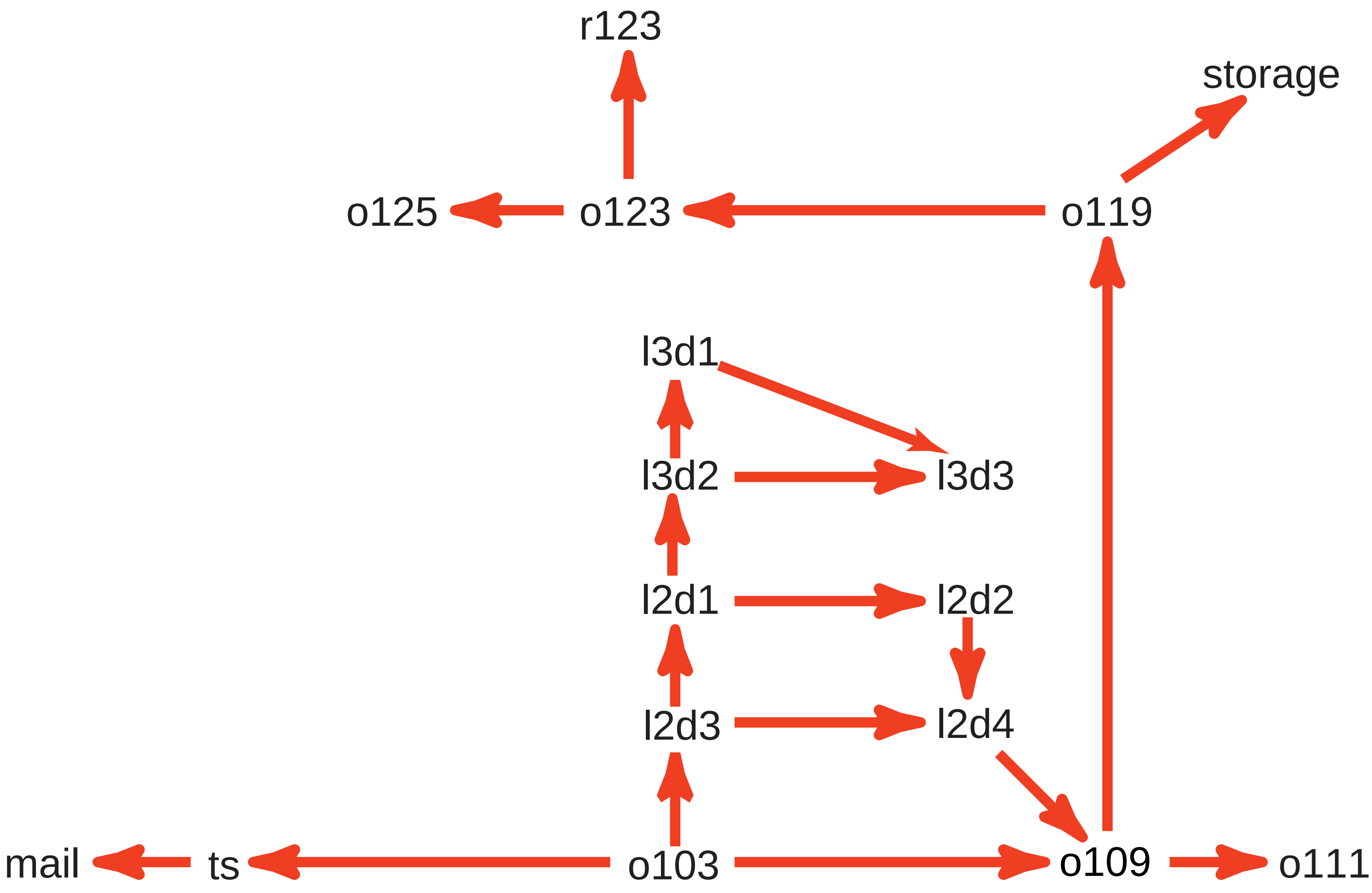
- Search is a way to implement don't know nondeterminism.

- So far we have seen how to convert a semantic problem of finding logical consequence to a search problem of finding derivations.

Search Graphs

- A graph consists of a set N of nodes and a set A of ordered pairs of nodes, called arcs.
- Node n_2 is a neighbor of n_1 if there is an arc from n_1 to n_2. That is, if $\langle n_1, n_2 \rangle \in A$.
- A path is a sequence of nodes $n_0, n_1, \ldots, n_k$ such that $\langle n_{i-1}, n_i \rangle \in A$.
- Given a set of start nodes and goal nodes, a solution is a path from a start node to a goal node.

Example Graph for the Delivery Robot



Search Graph for SLD Resolution

$a \leftarrow b \wedge c.$ $\quad a \leftarrow g.$

$a \leftarrow h.$ $\quad b \leftarrow j.$

$b \leftarrow k.$ $\quad d \leftarrow m.$

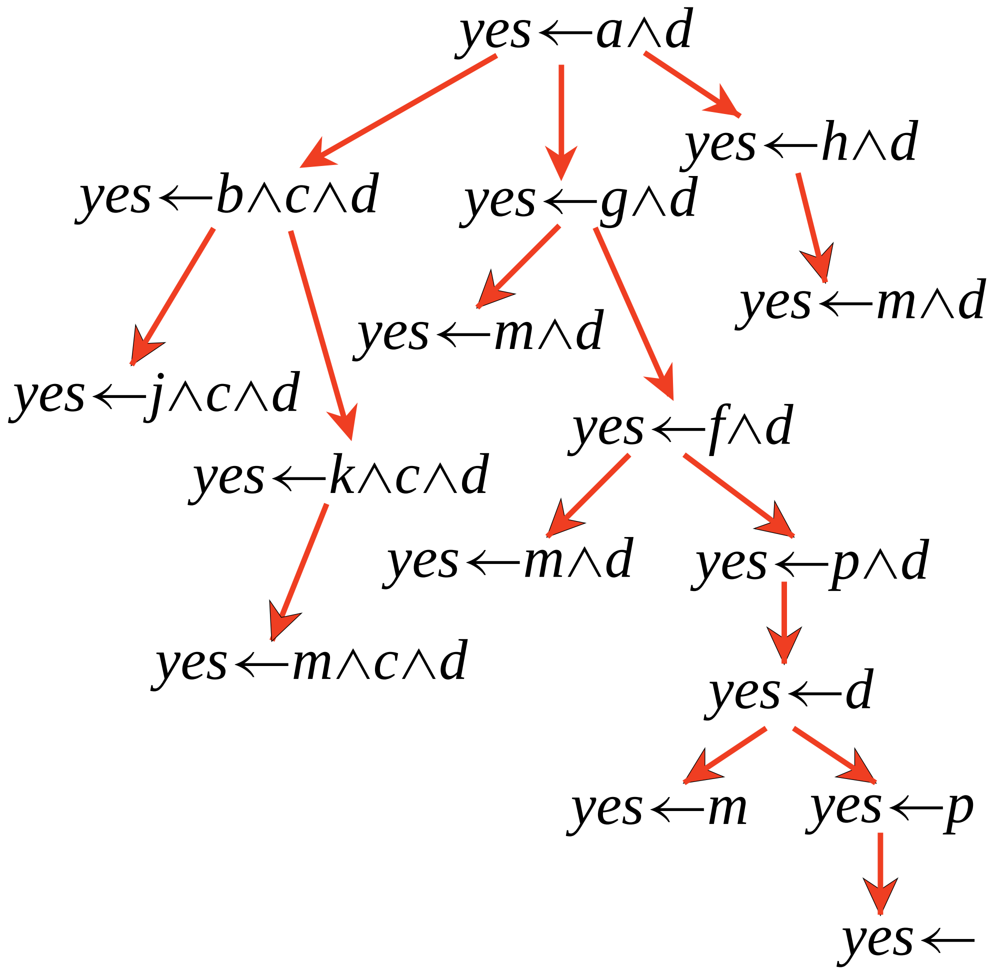
$d \leftarrow p.$ $\quad f \leftarrow m.$

$f \leftarrow p.$ $\quad g \leftarrow m.$

$g \leftarrow f.$ $\quad k \leftarrow m.$

$h \leftarrow m.$ $\quad p.$

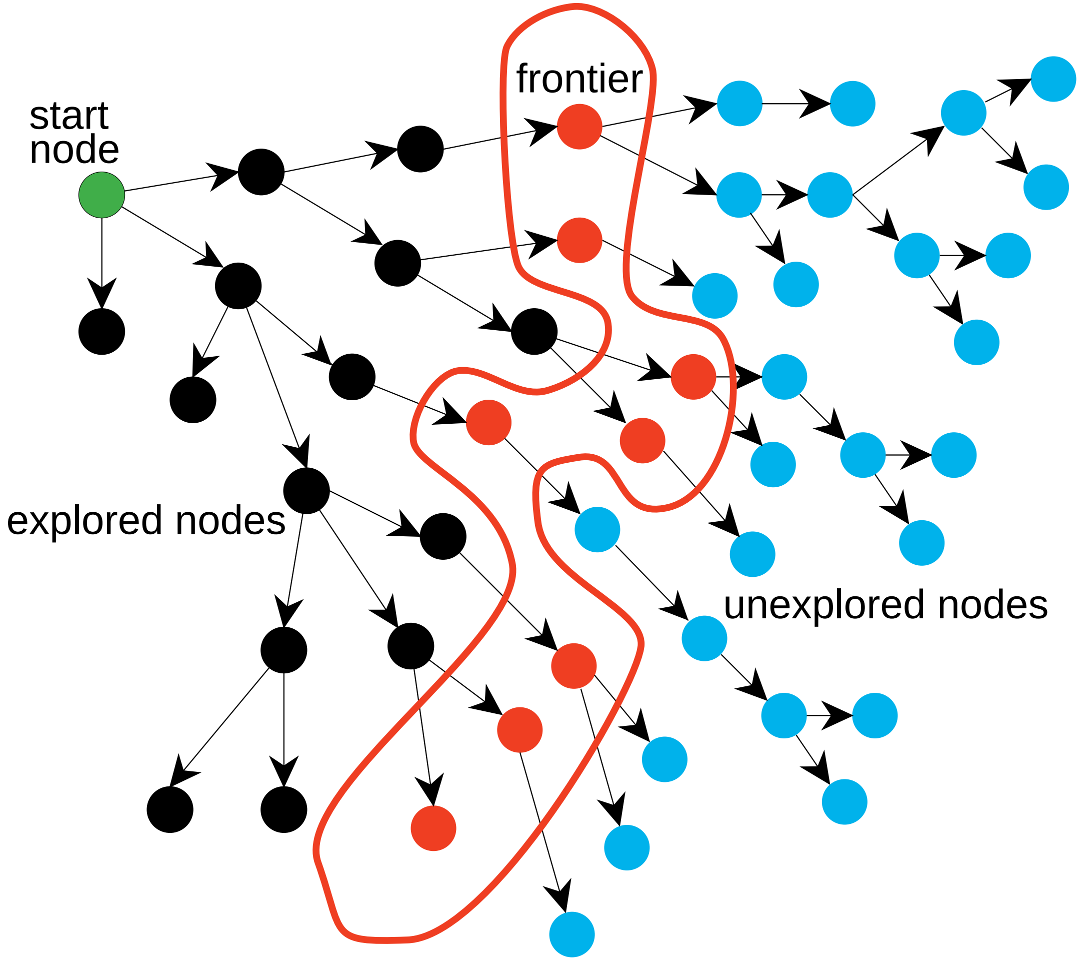
$?a \wedge d$



Graph Searching

- Generic search algorithm: given a graph, start nodes, and goal nodes, incrementally explore paths from the start nodes.
- Maintain a frontier of paths from the start node that have been explored.
- As search proceeds, the frontier expands into the unexplored nodes until a goal node is encountered.
- The way in which the frontier is expanded defines the search strategy.

Problem Solving by Graph Searching



Generic Graph Search Algorithm

$$
\begin{aligned}
&search(F_0) \leftarrow \\
&\qquad select(Node, F_0, F_1) \wedge \\
&\qquad is\_goal(Node). \\
&search(F_0) \leftarrow \\
&\qquad select(Node, F_0, F_1) \wedge \\
&\qquad neighbors(Node, NN) \wedge \\
&\qquad add\_to\_frontier(NN, F_1, F_2) \wedge \\
&\qquad search(F_2).
\end{aligned}
$$

- *search*(*Frontier*) is true if there is a path from one element of the *Frontier* to a goal node.
- *is_goal*(N) is true if N is a goal node.
- *neighbors*(N, NN) means NN is list of neighbors of N.
- *select*(N, F_0, F_1) means $N \in F_0$ and $F_1 = F_0 - \{N\}$. Fails if F_0 is empty.
- *add_to_frontier*(NN, F_1, F_2) means that $F_2 = F_1 \cup NN$.

select and *add_to_frontier* define the search strategy.

neighbors defines the graph

is_goal defines what is a solution.