Пройденный путь будет храниться в виде списка состояний, но в обратном порядке, т.е стартовая вершина будет расположена в конце, а текущая вершина в начале.  
Чтобы продлить путь на один ход, то надо найти какой шаг мы можем сделать, и во избежание зацикливания проверить, чтобы этот шаг не привел к состоянию, в котором мы уже побывали.

| **Prolog**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2 | prolong([Temp|Tail],[New,Temp|Tail]):-      move(Temp,New),not(member(New,[Temp|Tail])). | |  |

Теперь сам предикат поиска в глубину

| **Prolog**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4  5  6 | dpth([Finish|Tail],Finish,[Finish|Tail]). *%если текущая вершина*  *%совпадает с конечной, то путь найден*  dpth(TempWay,Finish,Way):-      prolong(TempWay,NewWay), *%пробуем сделать шаг*      dpth(NewWay,Finish,Way).*%продолжаем поиск уже*  *%с учетом сделанного шага* | |  |

Вспомогательный предикат для удобства пользователя

| **Prolog**[Выделить код](http://www.cyberforum.ru/) | |
| --- | --- |
| |  |  | | --- | --- | | 1  2  3  4 | search\_dpth(Start,Finish):-      dpth([Start],Finish,Way),*%вызываем поиск в глубину,*  *%считая, что пока путь состоит только из начальной вершины*      show\_answer(Way).*%выводим путь на экран в наглядном виде* | |  |