

The Farmer, Fox, Goose and Grain

Representation: $[L|R]$ with

- L the items on the left bank and
- R the items on the right bank.

Begin: $[Fa\ Fo\ Go\ Gr\ |]$
 End: $[|Fa\ Fo\ Go\ Gr]$

Rules: $\left\{ \begin{array}{l} [Fa\ \bar{x}|y] \rightarrow [\bar{x}|Fa\ y] \\ [\bar{x}|Fa\ y] \rightarrow [Fa\ \bar{x}|y] \\ [Fa\ z\bar{x}|y] \rightarrow [\bar{x}|Fa\ zy] \\ [\bar{x}|Fa\ zy] \rightarrow [Fa\ z\bar{x}|y] \end{array} \right.$

such that at no point there is combination of either $(Fo\ Go)$ or $(Go\ Gr)$ without the farmer on either bank.

The Farmer, Fox, Goose and Grain: Depth first (2)

8. $\langle [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr] [Fa\ Fo\ Go\ |Gr] [Go\ |Fa\ Fo\ Gr] [Fa\ Go\ |Fo\ Gr] [|Fa\ Fo\ Go\ Gr], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr] [Fa\ Fo\ Go\ |Gr] [Go\ |Fa\ Fo\ Gr] [Fa\ Go\ Gr\ |Fo], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Gr\ |Fa\ Fo\ Go] \rangle$

3

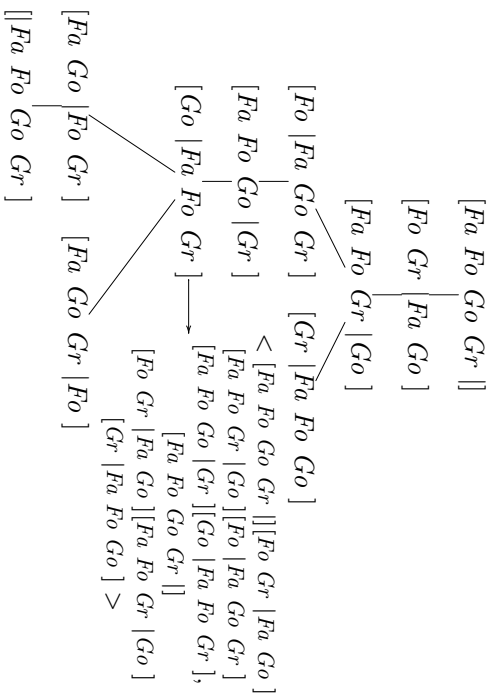
1

The Farmer, Fox, Goose and Grain: Depth first (1)

1. $\langle [Fa\ Fo\ Go\ Gr\ |] \rangle$
2. $\langle [Fa\ Fo\ Go\ Gr\ |][Fo\ Gr\ |Fa\ Go] \rangle$
3. $\langle [Fa\ Fo\ Go\ Gr\ |][Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] \rangle$
4. $\langle [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Gr\ |Fa\ Fo\ Go] \rangle$
5. $\langle [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr] [Fa\ Fo\ Go\ |Gr], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Gr\ |Fa\ Fo\ Go] \rangle$
6. $\langle [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr] [Fa\ Fo\ Go\ |Gr] [Go\ |Fa\ Fo\ Gr], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Gr\ |Fa\ Fo\ Go] \rangle$
7. $\langle [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr] [Fa\ Fo\ Go\ |Gr] [Go\ |Fa\ Fo\ Gr] [Fa\ Go\ |Fo\ Gr], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Fo\ |Fa\ Go\ Gr] [Fa\ Fo\ Go\ |Gr] [Go\ |Fa\ Fo\ Gr] [Fa\ Go\ Gr\ |Fo], [Fa\ Fo\ Go\ Gr\ |] [Fo\ Gr\ |Fa\ Go] [Fa\ Fo\ Gr\ |Go] [Gr\ |Fa\ Fo\ Go] \rangle$

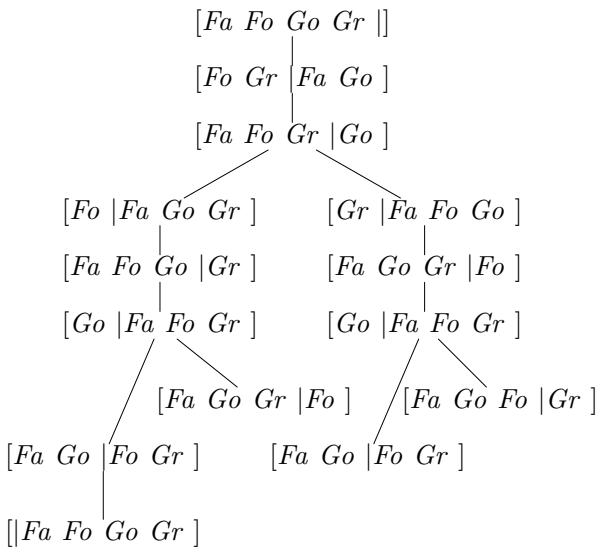
2

The Farmer, Fox, Goose and Grain: Depth first (3)



4

The Farmer, Fox, Goose and Grain: Breadth first



5

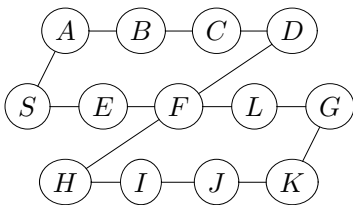
Bidirectional search

Each combination with at least one complete algorithm, is complete, but not each combination benefits from searching at both ends.

When only checking for identical end nodes, paths can cross each other unnoticed.

6

Bidirectional search (2)



Example 1: Depth first (2x)

- | | |
|----------------|----------------|
| < S > | < G > |
| < SA, SE > | < GK, GL > |
| < SAB, SE > | < GKJ, GL > |
| < SABC, SE > | < GKJI, GL > |
| < SABCD, SE > | < GKJIH, GL > |
| < SABCDF, SE > | < GKJIHF, GL > |

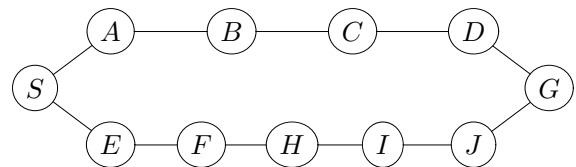
Example 2: Breadth first (2x)

- | | |
|--------------|--------------|
| < S > | < G > |
| < SA, SE > | < GK, GL > |
| < SE, SAB > | < GL, GKJ > |
| < SAB, SEF > | < GKJ, GLF > |

7

Bidirectional search (3)

Example 3: Breadth first / Depth first

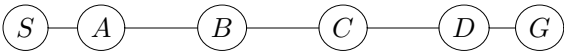


- | | |
|--------------|---------------|
| < S > | < G > |
| < SA, SE > | < GJ, GD > |
| < SE, SAB > | < GJI, GD > |
| < SAB, SEF > | < GJIH, GD > |
| < SEF, SAB > | < GJIHF, GD > |

8

Bidirectional search (4)

Example 4: Only check for identical end nodes



| | |
|------------|------------|
| < S > | < G > |
| < SA > | < GD > |
| < SAB > | < GDC > |
| < SABC > | < GDCB > |
| < SABCD > | < GDCBA > |
| < SABCDG > | < GDCBAS > |

Beam search

```

QUEUE ← path only containing root
WIDTH ← <number>

WHILE QUEUE is not empty
    AND goal is not reached
DO remove all paths from the QUEUE
    create new paths (to all children)
    reject the new paths with loops
    sort the new paths (by heuristic)
    add the WIDTH best paths to QUEUE

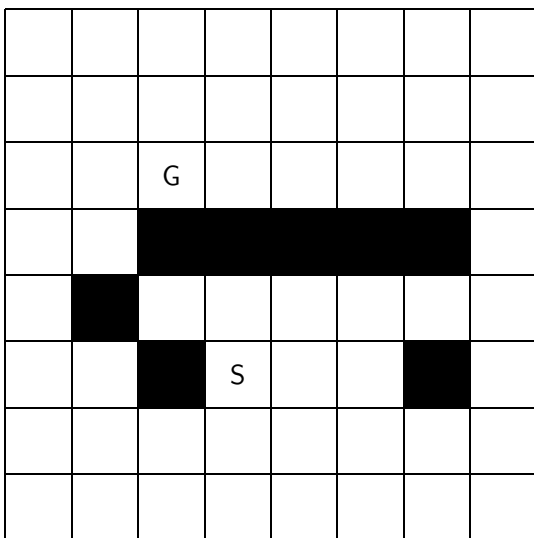
IF goal reached
    THEN success
    ELSE failure
  
```

Optimisation : before the last DO-line :
 "remove the paths with no successors "

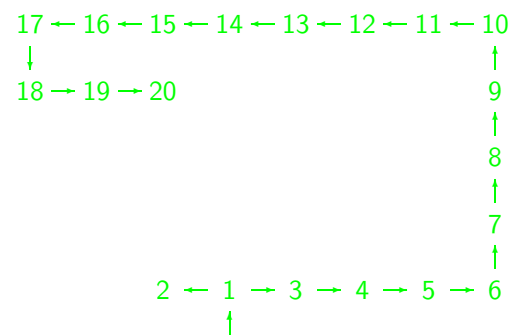
9

10

Path Search



Path Search: Depth first



11

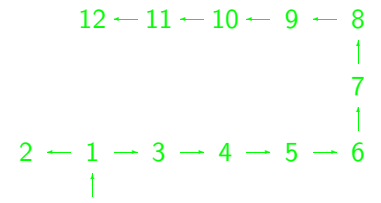
12

Path Search: Manhattan Distance

| | | | | | | | | | |
|--|---|---|---|---|---|---|---|----|---|
| | 4 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | |
| | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | |
| | 2 | 1 | G | 0 | 1 | 2 | 3 | 4 | 5 |
| | 3 | 2 | | | | | | 6 | |
| | 4 | | | 2 | 3 | 4 | 5 | 6 | 7 |
| | 5 | 4 | | | S | 4 | 5 | 6 | 8 |
| | 6 | 5 | 4 | 5 | 6 | 7 | 8 | 9 | |
| | 7 | 6 | 5 | 6 | 7 | 8 | 9 | 10 | |

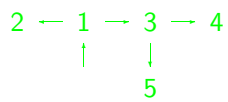
13

Path Search: Hill climbing 1



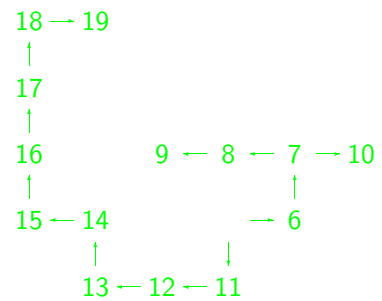
14

Path Search: Greedy search (1)



15

Path Search: Greedy search (2)



16

Water Jugs

Representation: $[x, y]$ with

- x contents of 4l jug
- y contents of 3l jug

Begin: $[0, 0]$
End: $[2, 0]$

Rules:

$$\left\{ \begin{array}{l} [x, y], x < 4 \rightarrow [4, y] \\ [x, y], y < 3 \rightarrow [x, 3] \\ [x, y], x > 0 \rightarrow [0, y] \\ [x, y], y > 0 \rightarrow [x, 0] \\ [x, y], x + y > 4 \ \& \ y > 0 \rightarrow [4, y - (4 - x)] \\ [x, y], x + y \leq 4 \ \& \ y > 0 \rightarrow [x + y, 0] \\ [x, y], x + y > 3 \ \& \ x > 0 \rightarrow [x - (3 - y), 3] \\ [x, y], x + y \leq 3 \ \& \ x > 0 \rightarrow [0, x + y] \end{array} \right.$$

17

Water Jugs: Heuristic

$$h([x, y]) = f(x) + f(y)$$

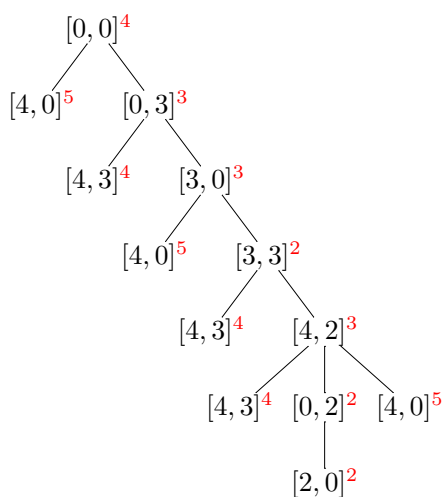
with

| | | | | | |
|--------|---|---|---|---|---|
| x | 0 | 1 | 2 | 3 | 4 |
| $f(x)$ | 2 | 1 | 0 | 1 | 3 |

“To get a 2, you need a 1 or a 3 and 0 is better than 4.”

18

Water Jugs: Hill Climbing



19