What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
ask continuous(l2, w5).
```

prove: ?a_1 ^ ... ^ a_k.

```
AC := yes <- $a_1$ ^ ... ^ $a_k$.

repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- a_1 ^ ... ^ a_k.

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(l2, w5) <- connected_to(l2, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(l2, w5) <- connected_to(l2, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

OOPS! There’s no good clause in KB.

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

OOPS! There’s no good clause in KB.

backtrack to last choice point

  reset everything to what it was then

  make another choice

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

OOPS! There’s no good clause in KB.
backtrack to last choice point
reset everything to what it was then
make another choice

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

OOPS! There’s no good clause in KB.
backtrack to last choice point
  reset everything to what it was then
  make another choice

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(X, Y) <- connected_to(X, Z) & continuous(Z, Y).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(l2, w5) <- connected_to(l2, Z) & continuous(Z, w5).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(l2, w5).

```
repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(l2, w5) <- connected_to(l2, Z) & continuous(Z, w5).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

```
AC := yes <- connected_to(l2, Z) ^
             continuous(Z, w5).

repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

What if you don’t choose well?

```
light(l1).
light(l2).
down(s1).
up(s2).
up(s3).
ok(l1).
ok(l2).
ok(cb1).
ok(cb2).
connected_to(l1, w0).
connected_to(w0, w1) <- up(s2).
connected_to(w0, w2) <- down(s2).
connected_to(w1, w3) <- up(s1).
connected_to(w2, w3) <- down(s1).
connected_to(l2, w4).
connected_to(w4, w3) <- up(s3).
connected_to(p1, w3).
connected_to(w3, w5) <- ok(cb1).
connected_to(p2, w6).
connected_to(w6, w5) <- ok(cb2).
connected_to(w5, outside).
continuous(l2, w5) <- connected_to(l2, Z) & continuous(Z, w5).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

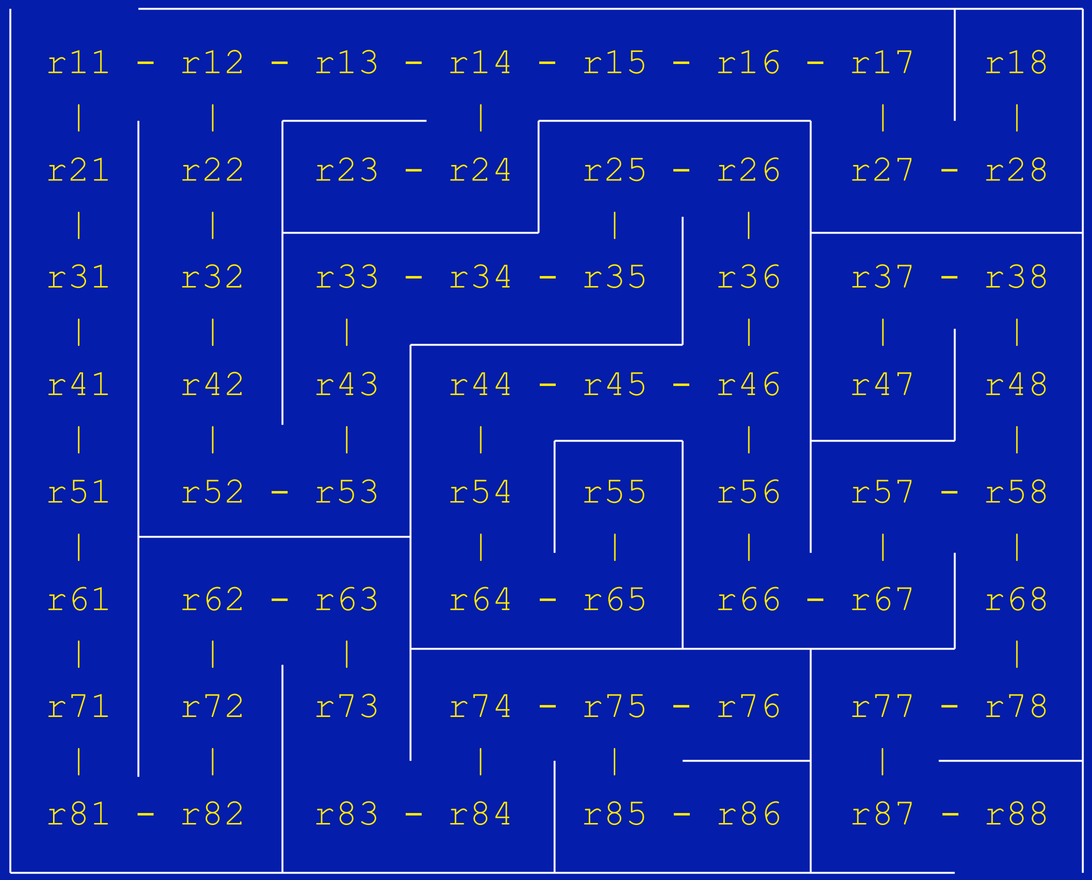
```
AC := yes <- connected_to(l2, Z) ^
             continuous(Z, w5).

repeat
    select a conjunct $a_i$ from the body of AC
    choose clause C from KB with $a_i$ as head
    replace $a_i$ in the body of AC by the body of C
      after making appropriate substitutions
until AC is an answer (i.e., yes <- .)
```

From this point on, it’s just a repeat of the previous example....

The Awesome Power of Recursion

or how getting the representation right makes everything else so easy....

The Maze



```
r11 - r12 - r13 - r14 - r15 - r16 - r17   r18
 |     |           |                 |     |
r21   r22   r23 - r24   r25 - r26   r27 - r28
 |     |                 |     |
r31   r32   r33 - r34 - r35   r36   r37 - r38
 |     |     |                 |     |     |
r41   r42   r43   r44 - r45 - r46   r47   r48
 |     |     |     |           |           |
r51   r52 - r53   r54   r55   r56   r57 - r58
 |                 |     |     |     |     |
r61   r62 - r63   r64 - r65   r66 - r67   r68
 |     |     |                             |
r71   r72   r73   r74 - r75 - r76   r77 - r78
 |     |           |     |           |
r81 - r82   r83 - r84   r85 - r86   r87 - r88
```

It’s time for a CILOG break!

```
cilog: load ‘maze.ci’.
```