CPSC 322

Introduction to Artificial Intelligence

September 22, 2004

Administrative Stuff

```
galiano:~> pl
Welcome to SWI-Prolog (Multi-threaded, Version 5.3.16)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?- compile('~cs322/cilog/cilog_swi.pl').

CILOG Version 0.13. Copyright 1998-2004, David Poole.
CILOG comes with absolutely no warranty.
All inputs end with a period. Type "help." for help.
cilog:
```

Administrative Stuff

```
Welcome to SWI-Prolog (Multi-threaded, Version 5.2.0)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?- [cilog_swi].

CILOG Version 0.12. Copyright 1998, David Poole.
CILOG comes with absolutely no warranty.
All inputs end with a period. Type "help." for help.
cilog: load 'brother.ci'.
CILOG theory brother.ci loaded.
cilog: listing.
parent(charles, william).
parent(charles, harry).
male(charles).
male(william).
male(harry).
brother(A, B) <- male(A)&parent(C, A)&parent(C, B)&different(A, B).
different(A, B) <- A\=B.
cilog: ask brother(william,harry).
Answer: brother(william, harry).
```

Administrative Stuff

```
Welcome to SWI-Prolog (Multi-threaded, Version 5.2.0)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?- [cilog_swi].

CILOG Version 0.13. Copyright 1998-2004, David Poole.
CILOG comes with absolutely no warranty.
All inputs end with a period. Type "help." for help.
cilog: load 'brother.ci'.
CILOG theory brother.ci loaded.
cilog: listing.
parent(charles, william).
parent(charles, harry).
male(charles).
male(william).
male(harry).
brother(A, B) <- male(A)&parent(C, A)&parent(C, B)&different(A, B).
different(A, B) <- A\=B.
cilog: ask brother(william,harry).
Error: error(existence_error(procedure, dif/2), A)
```

Administrative Stuff

Office hours

Qian Huang	Tu	2:00pm - 4:00pm	308A
Navjot Singh	Th	10:30am - 12:30pm	238
Kurt Eiselt	M W	3:30pm - 4:30pm	247

Administrative Stuff

Homework clarification: some basic relation, like “parent”, doesn’t have to be a rule...sorry for the confusion

Turn in assignments electronically - use `handin`
read `man handin`
our course is `cs322`
the first assignment is `asgn1`

Newsgroup: ubc.courses.cpsc.322

We’re hoping to have WebCT going real soon now....

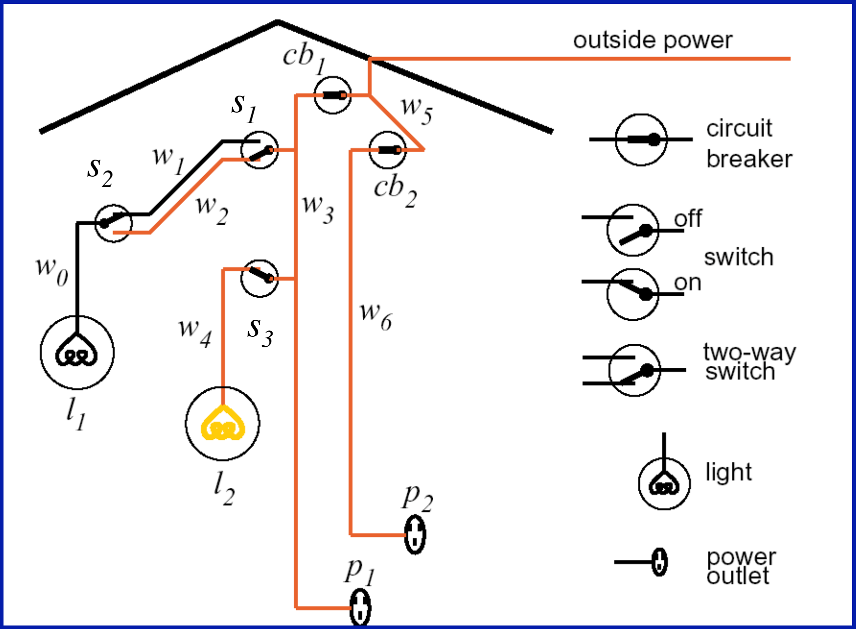
Read Chapter 3

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```



Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: $?a_1 \wedge ... \wedge a_k$.

```
AC := yes <- $a_1 \wedge ... \wedge a_k$.

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w3,w5) ^
connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- connected_to(w3,w5) ^
                 connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- connected_to(w3,w5) ^
             connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- connected_to(w3,w5) ^
             connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- connected_to(w3,w5) ^
             connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- connected_to(w3,w5) ^
             connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- ok(cb1) ^
             connected_to(w2,w3).

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- ok(cb1) ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Note: if the clause C that was chosen is a fact in the knowledge base (no body, just a head) then you know a_i is true

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- TRUE ^ connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Note: if the clause C that was chosen is a fact in the knowledge base (no body, just a head) then you know a_i is true

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Note: if the clause C that was chosen is a fact in the knowledge base (no body, just a head) then you know a_i is true

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- connected_to(w2,w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- down(s1).

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- TRUE.

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- .

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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

AC := yes <- .

```
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
until AC is an answer (i.e., yes <- .)
```

Highlights from last time

So what exactly happens when we ask?

```
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).

ask connected_to(w3,w5) &
    connected_to(w2,w3).
```

prove: ?connected_to(w3,w5) ^ connected_to(w2,w3).

```
AC := yes <- .

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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
ask continuous(l2, w3).
```

```
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
    until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- continuous(l2, w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- continuous(l2, w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- continuous(l2, w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- continuous(l2, w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- continuous(l2, w3).

```
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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- continuous(l2, w3).

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
until AC is an answer (i.e., yes <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- continuous(l2, w3).

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 <- .)
```

Once more, with variables

```
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, w3) <- connected_to(l2, Z) & connected_to(Z, w3).
```

prove: ?continuous(l2, w3).

AC := yes <- continuous(l2, w3).

```
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 <- .)
```

Once more, with variables

```
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, w3) <- connected_to(l2, Z) & connected_to(Z, w3).
```

prove: ?continuous(l2, w3).

```
AC := yes <- connected_to(l2, Z) ^
             connected_to(Z, w3).

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- connected_to(l2, Z) ^
                 connected_to(Z, w3).

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- connected_to(l2, Z) ^
             connected_to(Z, w3).

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

```
prove:  ?continuous(l2, w3).

    AC := yes <- connected_to(l2, Z) ^
                 connected_to(Z, w3).

    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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- connected_to(l2, Z) ^
             connected_to(Z, w3).

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(l2, Z) ^
connected_to(Z, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- connected_to(l2, Z) ^
             connected_to(Z, w3).

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(l2, w4) ^
connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- TRUE ^
             connected_to(w4, w3).

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- connected_to(w4, w3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- up(s3).

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

AC := yes <- TRUE.

```
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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- .

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

prove: ?continuous(l2, w3).

```
AC := yes <- .

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 <- .)
```

Once more, with variables

```
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) & connected_to(Z, Y).
```

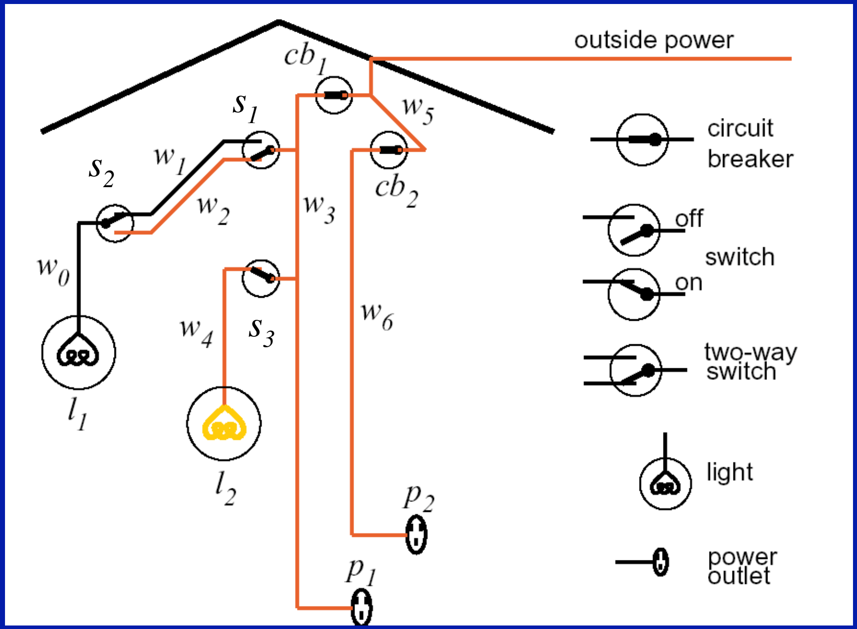
prove: ?continuous(l2, w3).

```
AC := yes <- .

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 <- .)
```

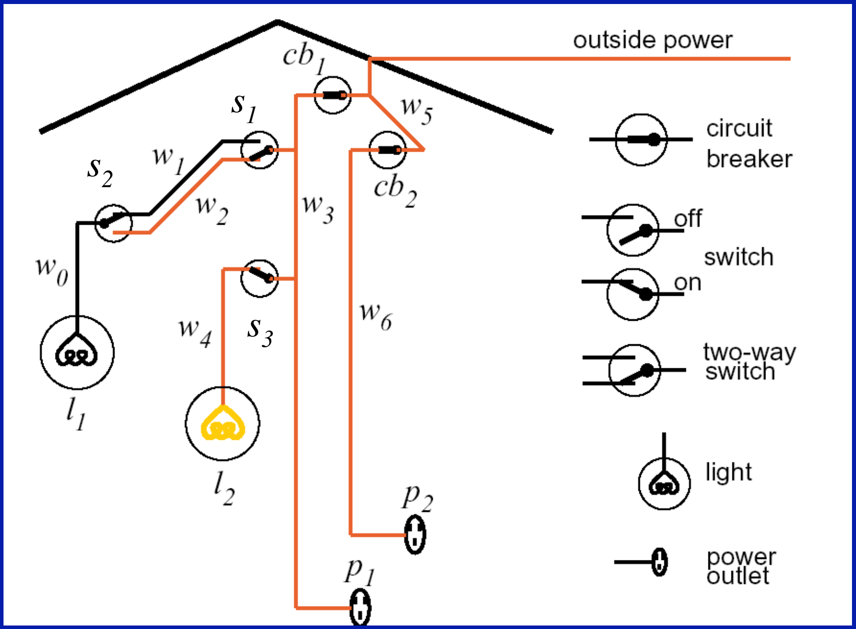
Can we make “continuous” arbitrarily long?

```
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) & connected_to(Z, Y).
```



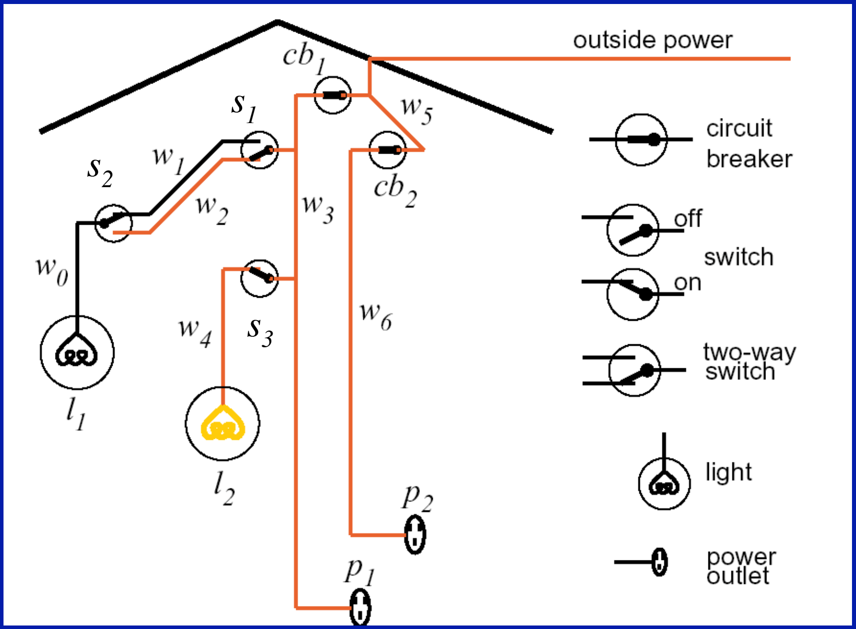
Can we make “continuous” arbitrarily long?

```
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) & connected_to(Z, Y).
continuous(X, Y) <- connected_to(X, Z) & connected_to(Z, A) & connected_to(A, Y).
```



Can we make “continuous” arbitrarily long?

```
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) & connected_to(Z, Y).
continuous(X, Y) <- connected_to(X, Z) & connected_to(Z, A) & connected_to(A, Y).
   ... and so on ...
```



Once more, with recursion

```
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:  ?$a_1$ ^ ... ^ $a_k$.

    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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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 <- .)
```

Once more, with recursion

```
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, 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 <- .)
```

Once more, with recursion

```
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, 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 <- .)
```

Once more, with recursion

```
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, 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 <- .)
```

Once more, with recursion

```
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, 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 <- .)
```

Once more, with recursion

```
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, w4) ^
             continuous(w4, 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 <- .)
```

Once more, with recursion

```
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 <- TRUE ^
                continuous(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, w5) <- connected_to(w4, Z) & continuous(Z, w5).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, w5) <- connected_to(w4, Z) & continuous(Z, w5).
continuous(X, Y) <- connected_to(X, Y).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, 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 <- .)
```

Once more, with recursion

```
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(w4, w3) ^
             continuous(w3, 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 <- .)
```

Once more, with recursion

```
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 <- up(s3) ^
continuous(w3, 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 <- .)
```

Once more, with recursion

```
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 <- up(s3) ^
                      continuous(w3, 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 <- .)
```

Once more, with recursion

```
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 <- up(s3) ^
continuous(w3, 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 <- .)
```

Once more, with recursion

```
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 <- up(s3) ^
                continuous(w3, 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 <- .)
```

Once more, with recursion

```
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 <- up(s3) ^
continuous(w3, 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 <- .)
```

Once more, with recursion

```
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 <- TRUE ^
                   continuous(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- continuous(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- connected_to(w3, 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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- ok(cb1).

```
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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- ok(cb1).

```
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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- ok(cb1).

```
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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- ok(cb1).

```
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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- ok(cb1).

```
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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

AC := yes <- TRUE.

```
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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

```
AC := yes <- .

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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

```
AC := yes <- .

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 <- .)
```

Once more, with recursion

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

```
AC := yes <- .

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 <- .)
```

That’s what happens when we ask.

```
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(w3, w5) <- connected_to(w3, w5).
```

prove: ?continuous(l2, w5).

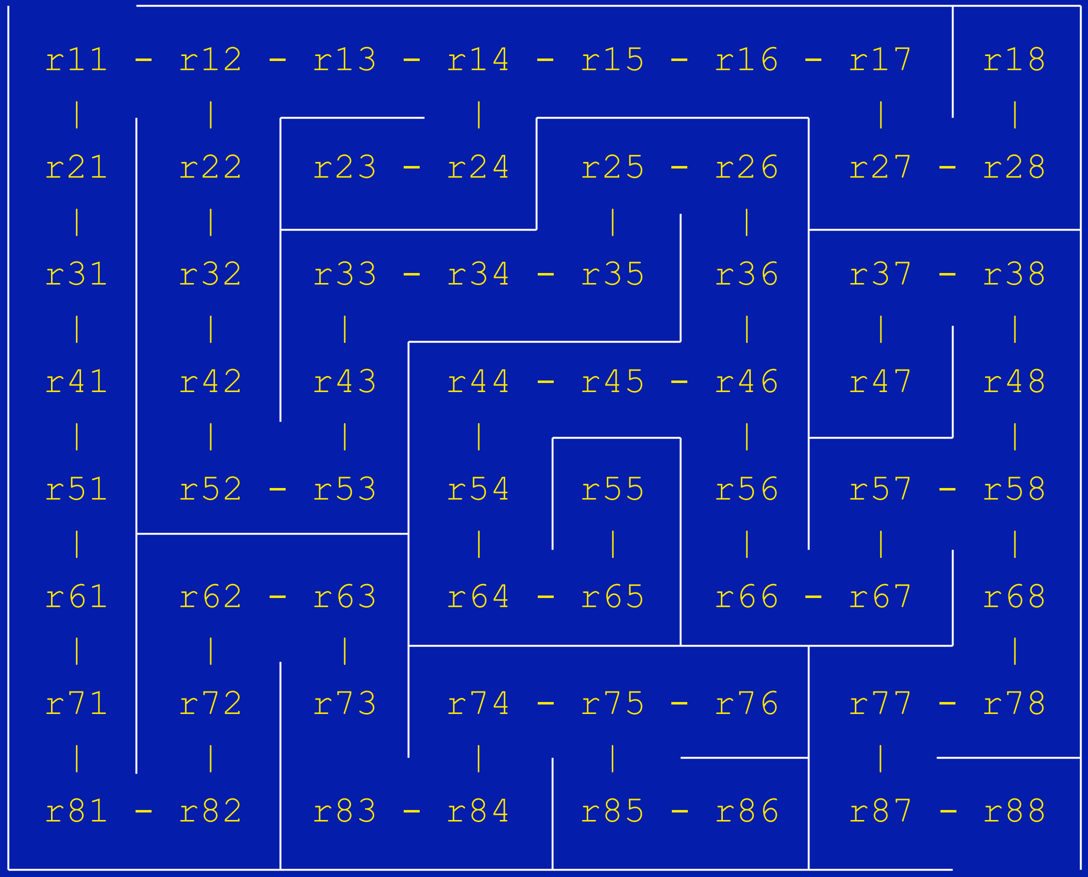
AC := yes <- .

```
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 <- .)
```

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
```