The program (P):

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y).
```

resolvent =

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent =

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

#### The program (P):

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent =

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)
```

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no

#### The program (P):

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,
```

else output no

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':-B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

The program (P):

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a ground instance of a clause

A':- B1,...,Bn from program P

such that A and A' are identical

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

OOPS! Time to modify things a bit.

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a clause

A':-B1,...,Bn from program P

such that A and A' unify

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a clause

A':-B1,...,Bn from program P

such that A and A' unify

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).

resolvent = grandmother(betty,roxy)

Initialize resolvent to goal G (the query)

while resolvent not empty do

choose a goal A from the resolvent

choose a clause

A':-B1,...,Bn from program P

such that A and A' unify

(if no such goal and clause exist, exit

the while loop)

replace A by B1,...,Bn in the resolvent

If the resolvent is empty, then output yes,

else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = grandmother(betty,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
   choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

parent(X,Y) := mother(X,Y).

parent(X,Y) := father(X,Y).

The program (P):

?- grandmother(betty,roxy). father(fred, pebbles). father(bamm-bamm, roxy). resolvent = grandmother(betty,roxy) father(barney,bamm-bamm). father(bamm-bamm,chip). mother(pebbles, roxy). Initialize resolvent to goal G (the query) mother(pebbles,chip). while resolvent not empty do mother(wilma, pebbles). choose a goal A from the resolvent mother(betty,bamm-bamm). choose a clause grandmother(betty,roxy) :-A' := B1,...,Bn from program Pmother(betty, Z), parent(Z, roxy). such that A and A'unify

the while loop)
replace A by B1,...,Bn in the resolvent
(after performing the appropriate
substitutions)

(if no such goal and clause exist, exit

If the resolvent is empty, then output yes, else output no

The program (P):

father(fred,pebbles).

```
father(fred,pebbles).
father(bamm-bamm,roxy). resolvent = mother(betty,Z),
father(barney,bamm-bamm). parent(Z,roxy)
father(bamm-bamm,chip).
mother(pebbles,roxy). Initialize resolvent to goal G (
mother(pebbles,chip). while resolvent not empty do
mother(wilma,pebbles). choose a goal A from the re
mother(betty,bamm-bamm). choose a clause
grandmother(betty,roxy):- A':-B1,...,Bn from prog
mother(betty,Z),parent(Z,roxy). such that A and A' unify
parent(X,Y):- mother(X,Y). (if no such goal and clause)
parent(X,Y):- father(X,Y).
```

else output no

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty, Z),
            parent(Z,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
   choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty, Z),
            parent(Z,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
   choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty, Z),
            parent(Z,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
   choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty, Z),
            parent(Z,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty, Z),
            parent(Z,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty, Z),
            parent(Z,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = mother(betty,bamm-bamm),
            parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
mother(X,Y):-
father(bamm-bamm,roxy):-
```

```
?- grandmother(betty,roxy).
resolvent = parent(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
mother(X,Y):-
father(bamm-bamm,roxy):-
father(bamm-bamm,roxy).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent = father(bamm-bamm,roxy)
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent =
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
  replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent =
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
resolvent =
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

```
father(fred,pebbles).
father(bamm-bamm,roxy).
father(barney,bamm-bamm).
father(bamm-bamm,chip).
mother(pebbles,roxy).
mother(pebbles,chip).
mother(wilma,pebbles).
mother(betty,bamm-bamm).
grandmother(X,Y):-
mother(X,Z),parent(Z,Y).
parent(X,Y):-
father(X,Y).
```

```
?- grandmother(betty,roxy).
ves
resolvent =
Initialize resolvent to goal G (the query)
while resolvent not empty do
   choose a goal A from the resolvent
  choose a clause
     A' := B1,...,Bn from program P
     such that A and A' unify
     (if no such goal and clause exist, exit
     the while loop)
   replace A by B1,...,Bn in the resolvent
     (after performing the appropriate
     substitutions)
If the resolvent is empty, then output yes,
  else output no
```

# Questions?