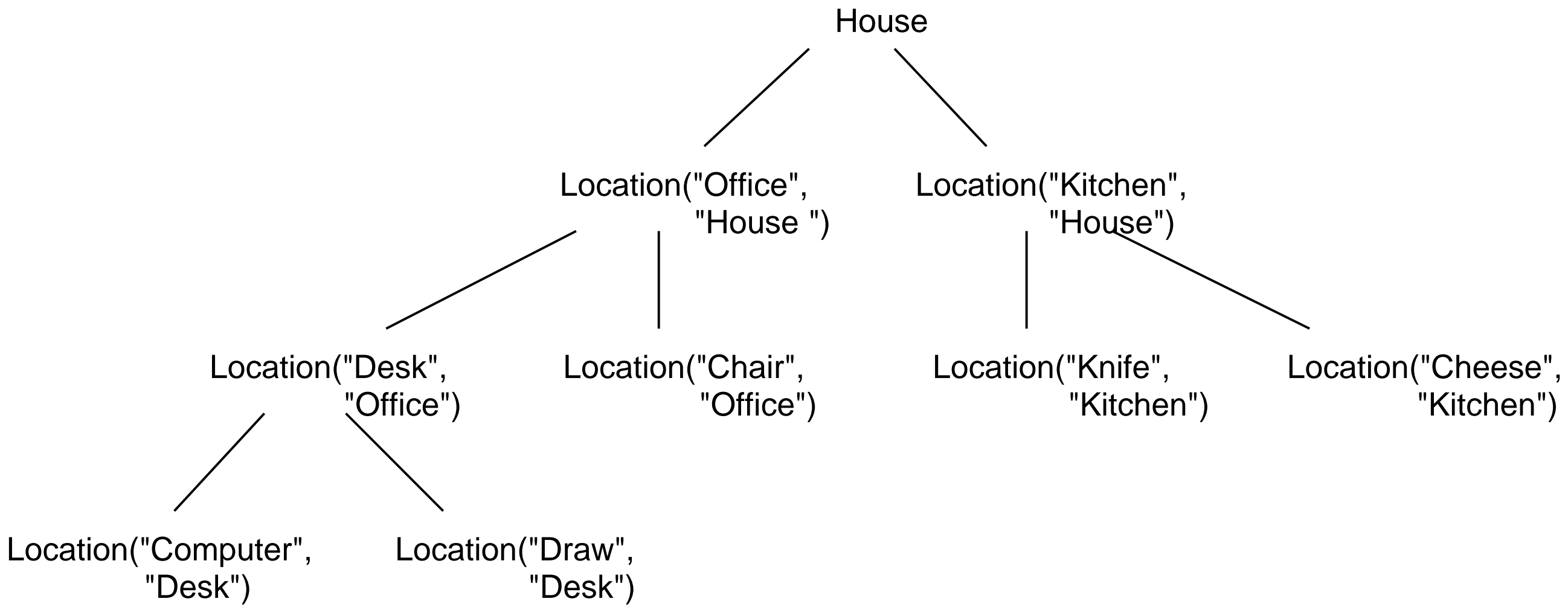


Reasoning with Graphs

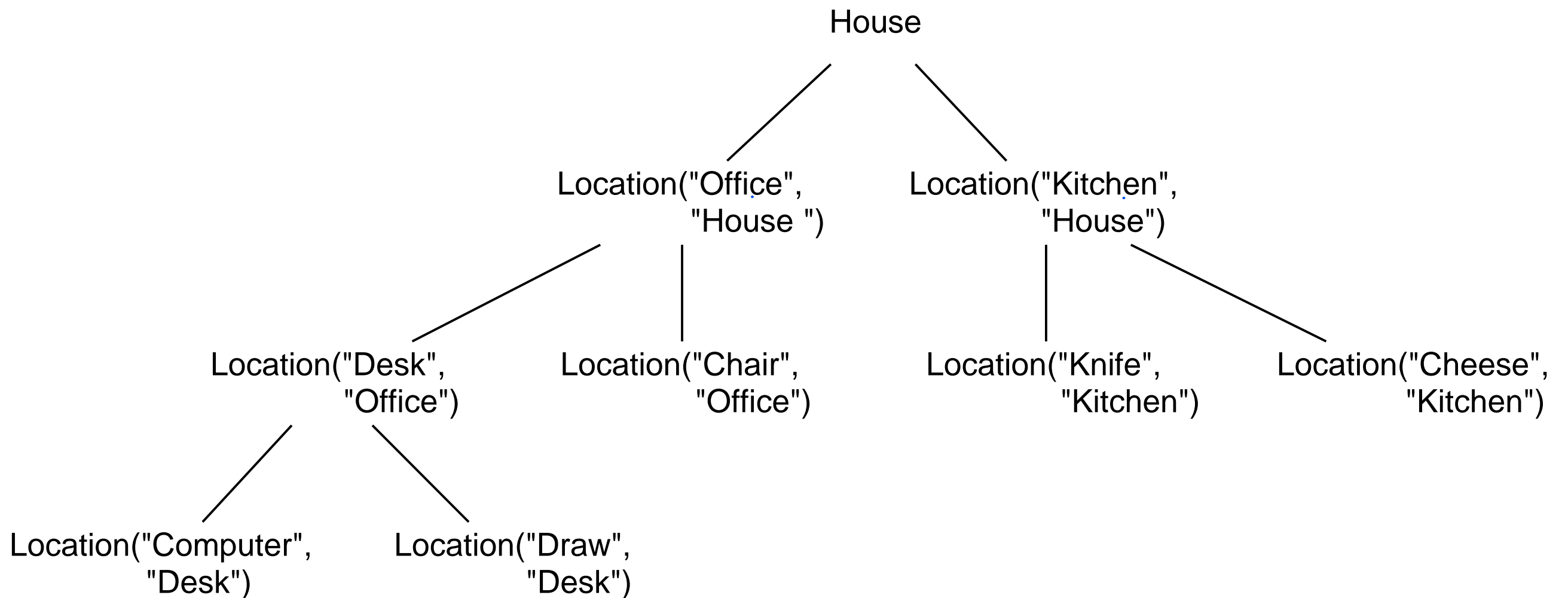


Backward Chaining

```

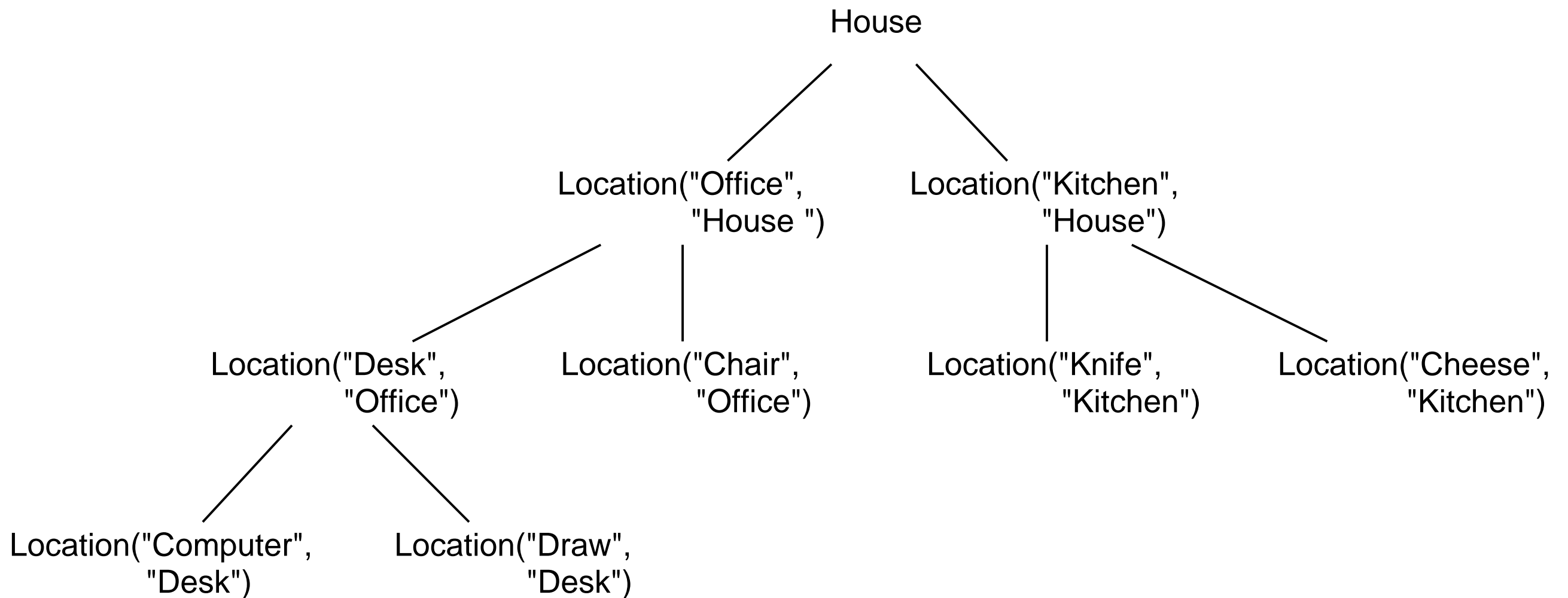
query isContainedIn( String x, String y )
  Location( x, y; )
  or
  ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



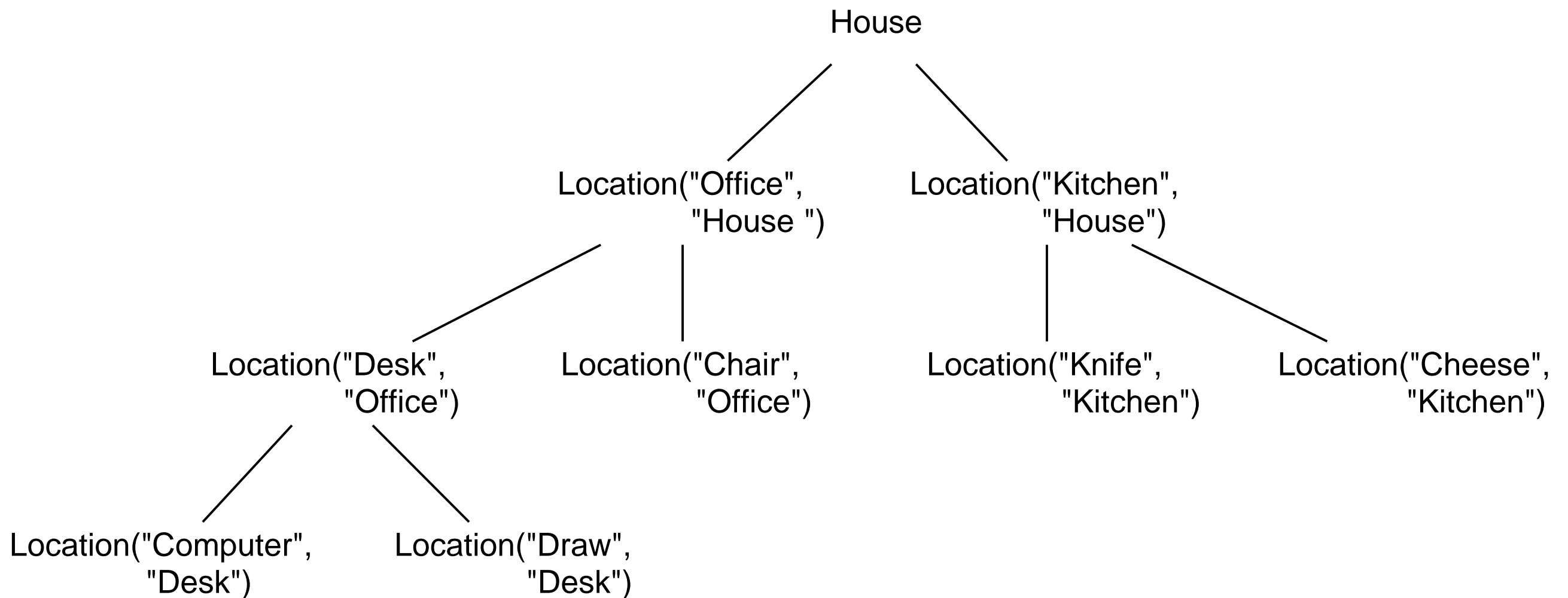
Backward Chaining

```
ksession.insert( new Location("Office", "House") );
ksession.insert( new Location("Kitchen", "House") );
ksession.insert( new Location("Knife", "Kitchen") );
ksession.insert( new Location("Cheese", "Kitchen") );
ksession.insert( new Location("Desk", "Office") );
ksession.insert( new Location("Chair", "Office") );
ksession.insert( new Location("Computer", "Desk") );
ksession.insert( new Location("Draw", "Desk") );
```



Backward Chaining

```
rule "go" salience 10
when
    $s : String( )
then
    System.out.println( $s );
end
```



Backward Chaining

```
rule "go" salience 10
```

```
when
```

```
    $s : String( )
```

```
then
```

```
    System.out.println( $s );
```

```
end
```

```
rule "go1"
```

```
when
```

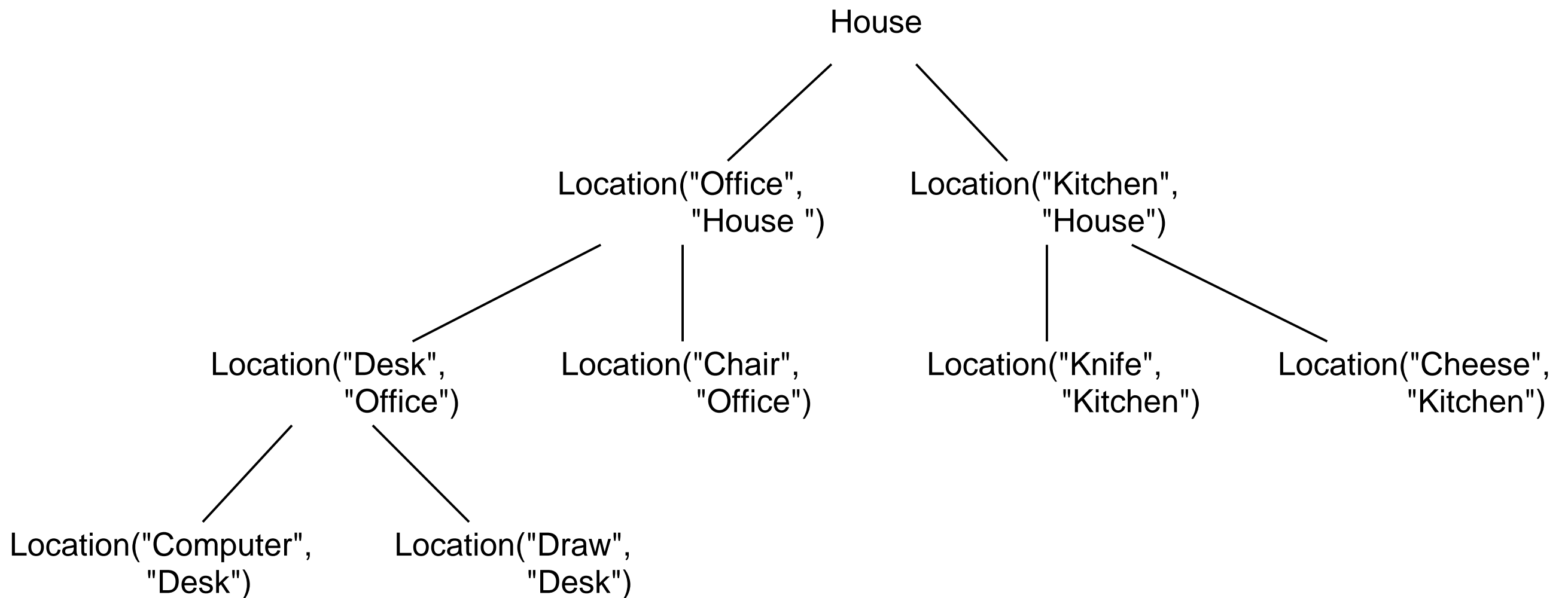
```
    String( this == "go1" )
```

```
    isContainedIn("Office", "House"; )
```

```
then
```

```
    System.out.println( "office is in the house" );
```

```
end
```

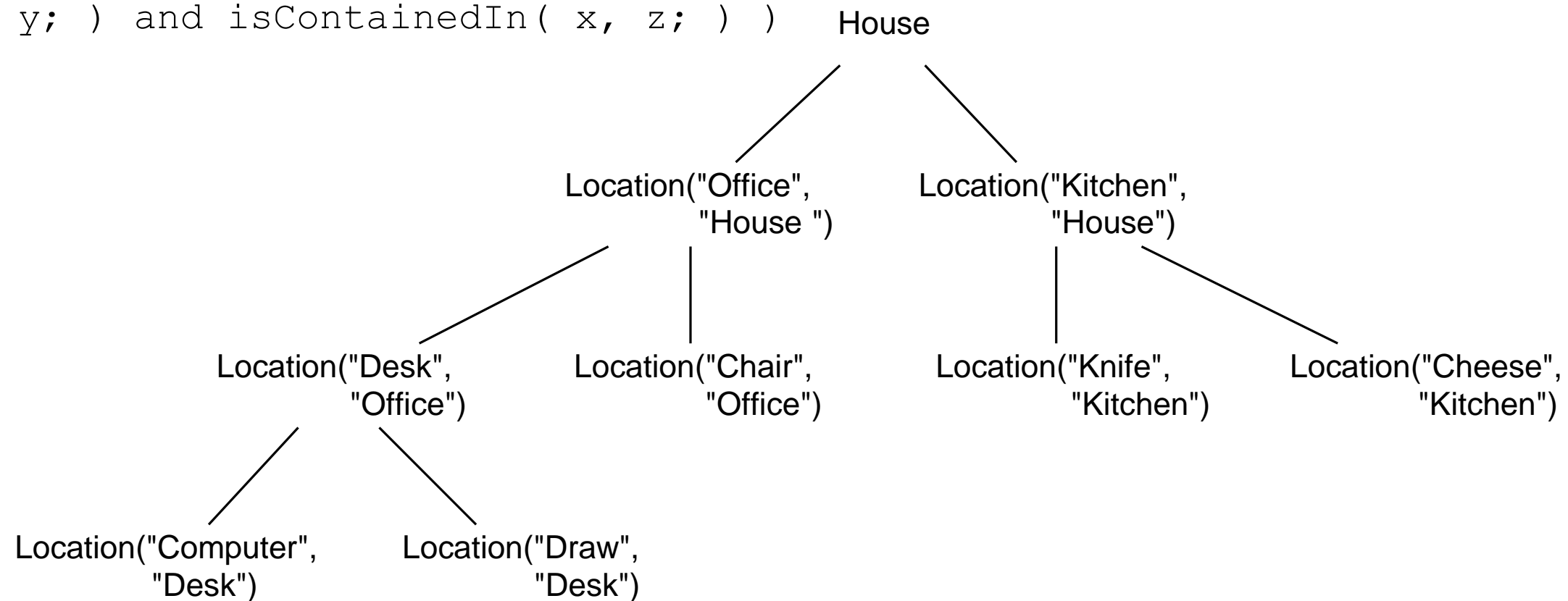


Backward Chaining

```
rule "go" salience 10
when
    $s : String( )
then
    System.out.println( $s );
end
```

```
rule "go1"
when
    String( this == "go1" )
    isContainedIn( "Office", "House"; )
then
    System.out.println( "office is in the house" );
end
```

```
query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end
```



Backward Chaining

```
rule "go" salience 10
when
```

```
    $s : String( )
```

```
then
```

```
    System.out.println( $s );
```

```
end
```

```
query isContainedIn( String x, String y )
```

```
    Location( x, y; )
```

```
    or
```

```
    ( Location( z, y; ) and isContainedIn( x, z; ) )
```

```
end
```

```
rule "go1"
```

```
when
```

```
    String( this == "go1" )
```

```
    isContainedIn( "Office", "House"; )
```

```
then
```

```
    System.out.println( "office is in the house" );
```

```
end
```

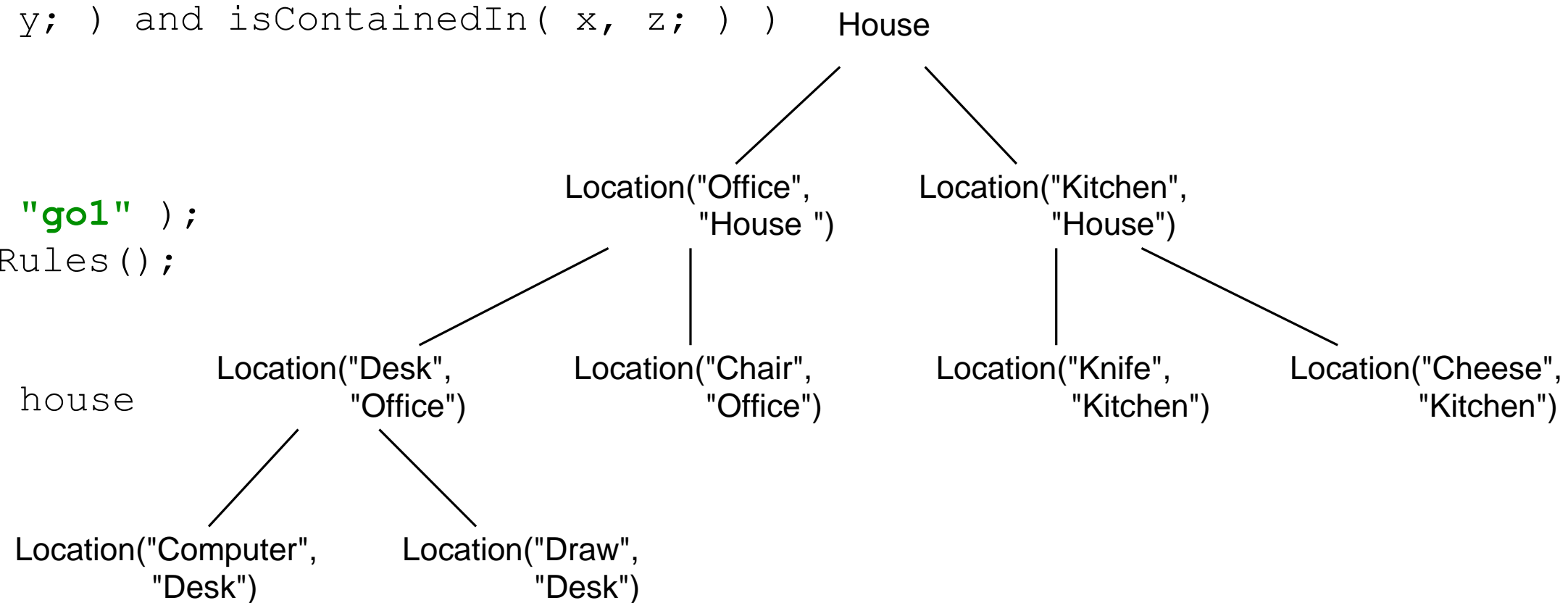
```
ksession.insert( "go1" );
```

```
ksession.fireAllRules();
```

```
---
```

```
go1
```

```
office is in the house
```



Backward Chaining

```
rule "go" salience 10
when
```

```
    $s : String( )
```

```
then
```

```
    System.out.println( $s );
```

```
end
```

```
query isContainedIn( String x, String y )
```

```
    Location( x, y; )
```

```
    or
```

```
    ( Location( z, y; ) and isContainedIn( x, z; ) )
```

```
end
```

```
rule "go1"
```

```
when
```

```
    String( this == "go1" )
```

```
    isContainedIn( "Office", "House"; )
```

```
then
```

```
    System.out.println( "office is in the house" );
```

```
end
```

```
isContainedIn(x==Office, y==House)
```

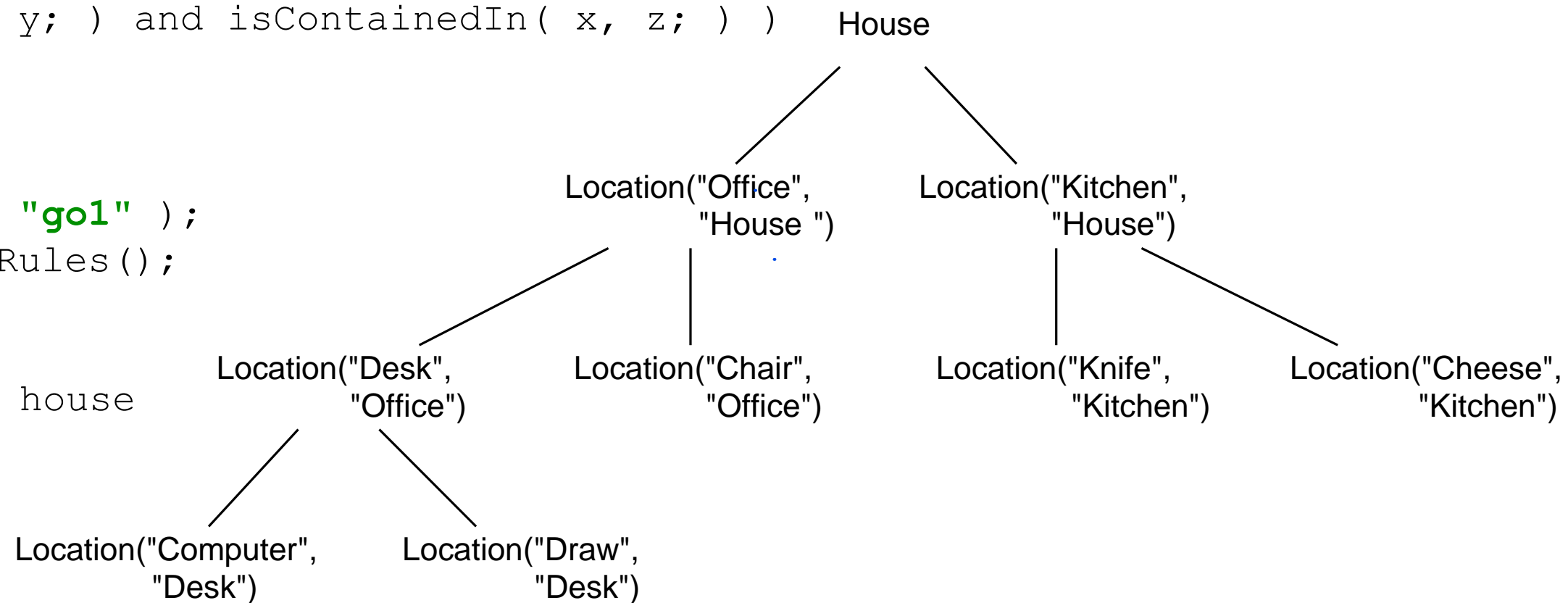
```
ksession.insert( "go1" );
```

```
ksession.fireAllRules();
```

```
---
```

```
go1
```

```
office is in the house
```



Backward Chaining

```
rule "go" salience 10
when
```

```
    $s : String( )
```

```
then
```

```
    System.out.println( $s );
```

```
end
```

```
query isContainedIn( String x, String y )
```

```
    Location( x, y; )
```

```
    or
```

```
    ( Location( z, y; ) and isContainedIn( x, z; ) )
```

```
end
```

```
rule "go1"
```

```
when
```

```
    String( this == "go1" )
```

```
    isContainedIn( "Office", "House"; )
```

```
then
```

```
    System.out.println( "office is in the house" );
```

```
end
```

```
isContainedIn(x==Office, y==House)
```

```
Location(x==Office, y==House)
```

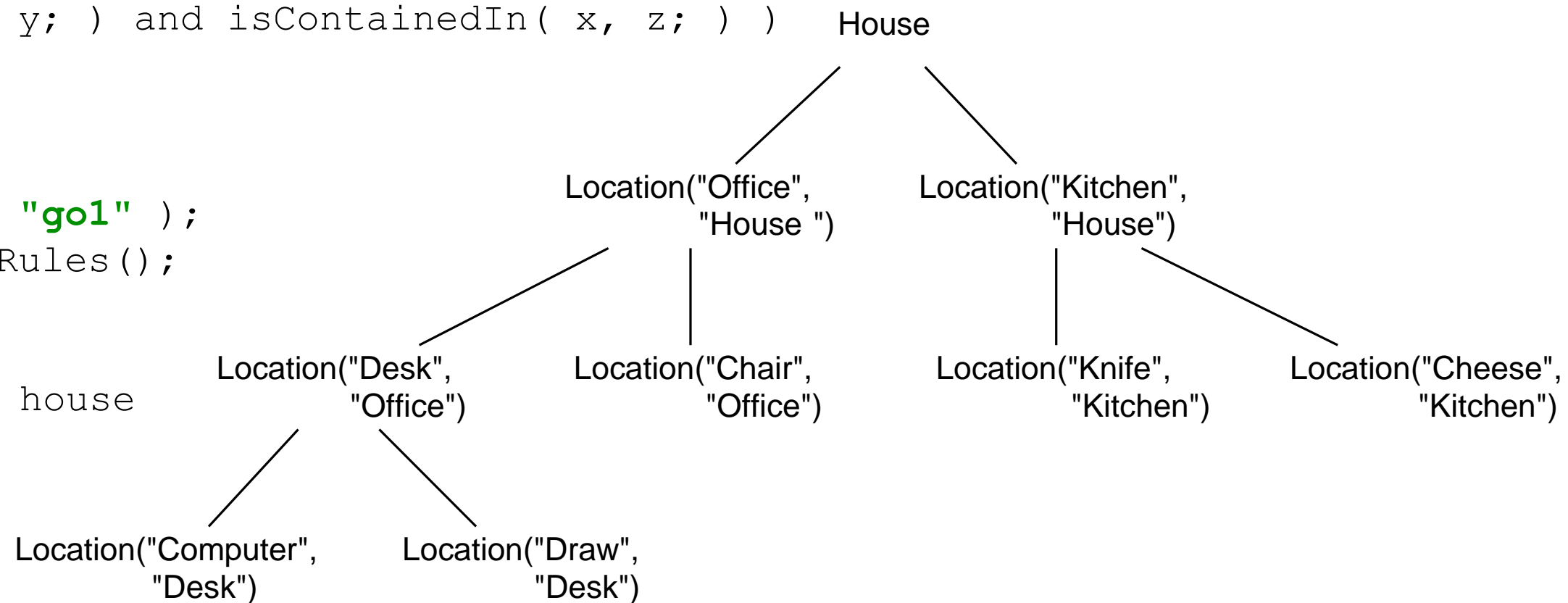
```
ksession.insert( "go1" );
```

```
ksession.fireAllRules();
```

```
---
```

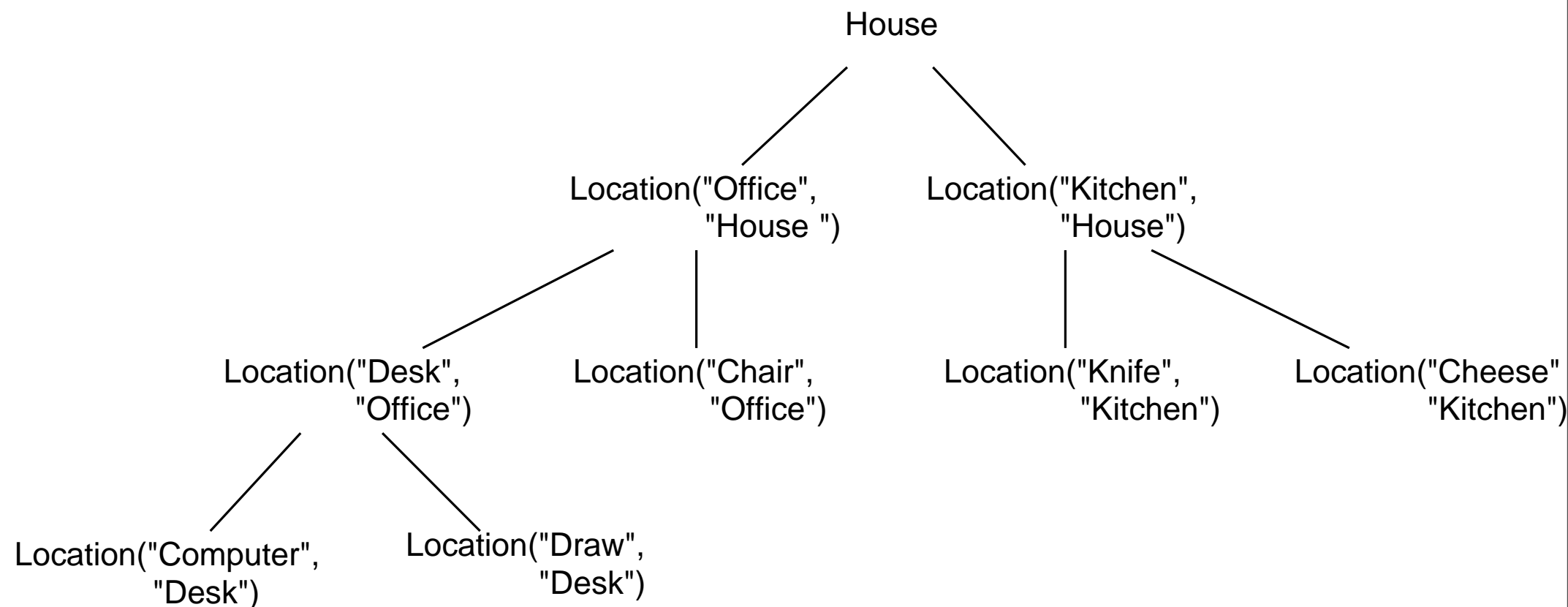
```
go1
```

```
office is in the house
```



Backward Chaining

```
rule "go2"  
when  
    String( this == "go2" )  
    isContainedIn("Draw", "House"; )  
then  
    System.out.println( "Draw in the House" );  
end
```



Backward Chaining

```

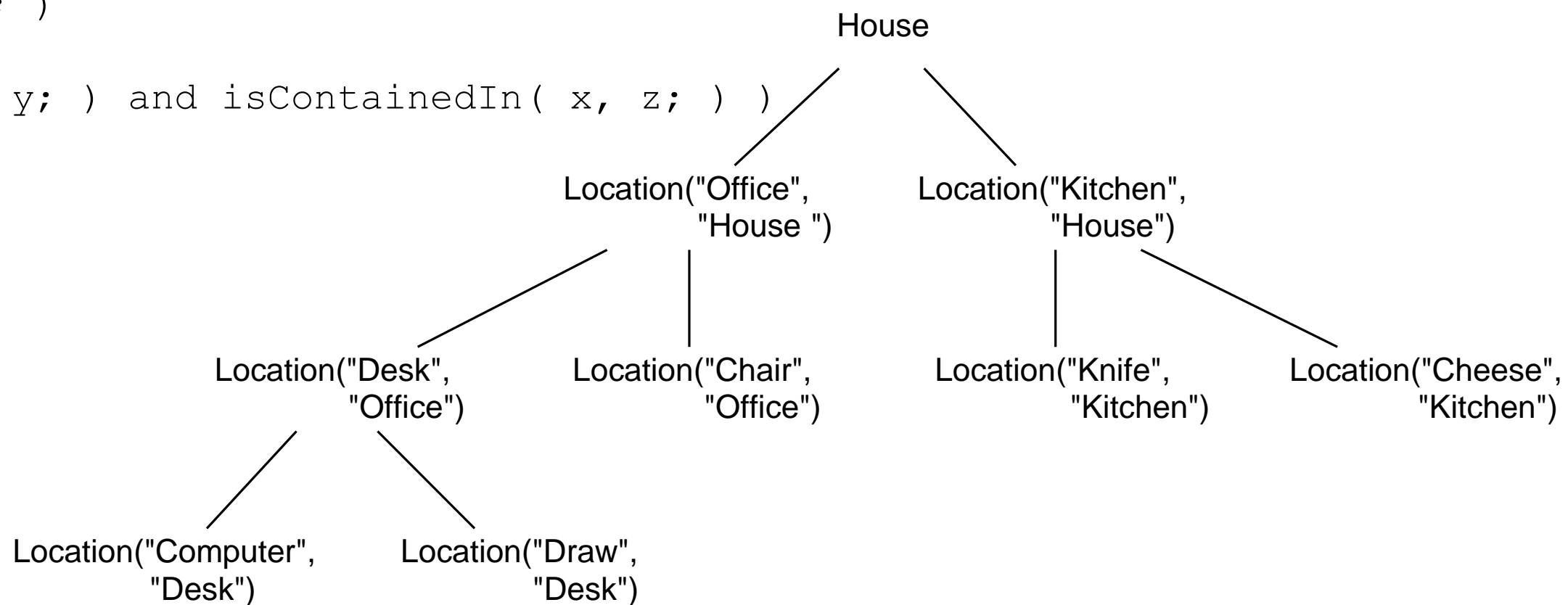
rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

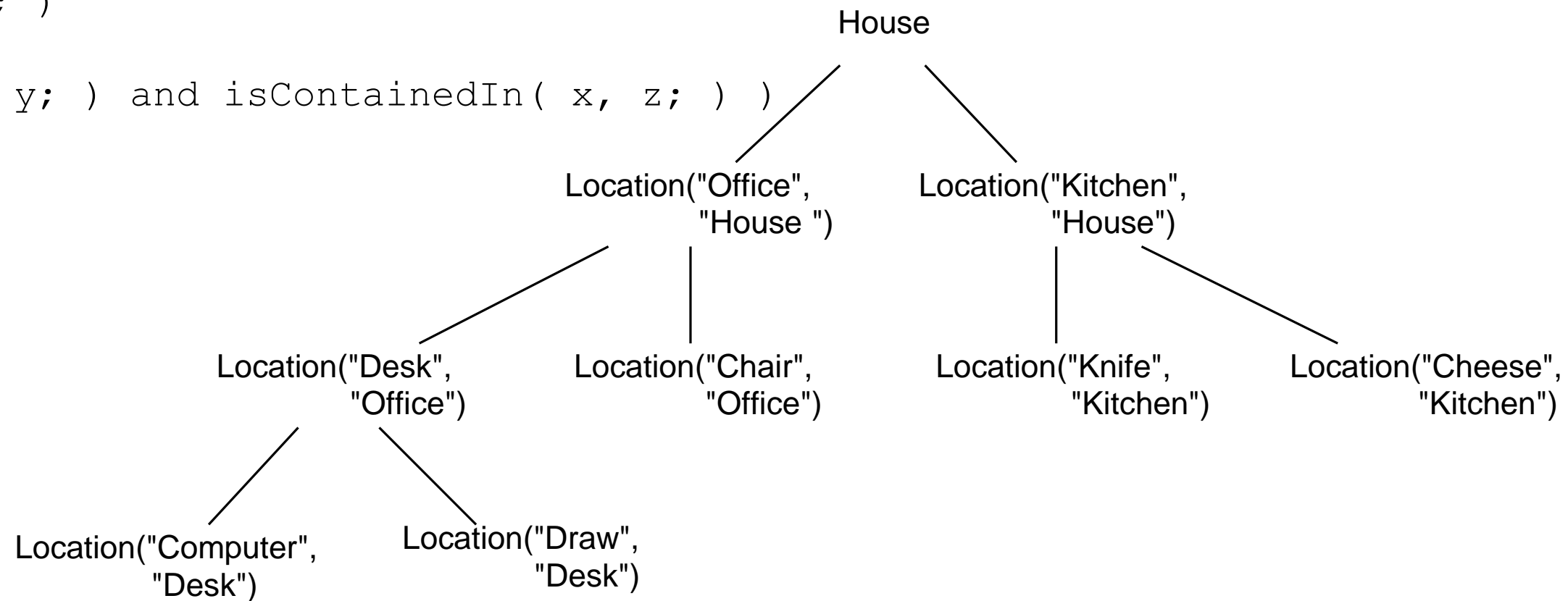
ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();

```

go2

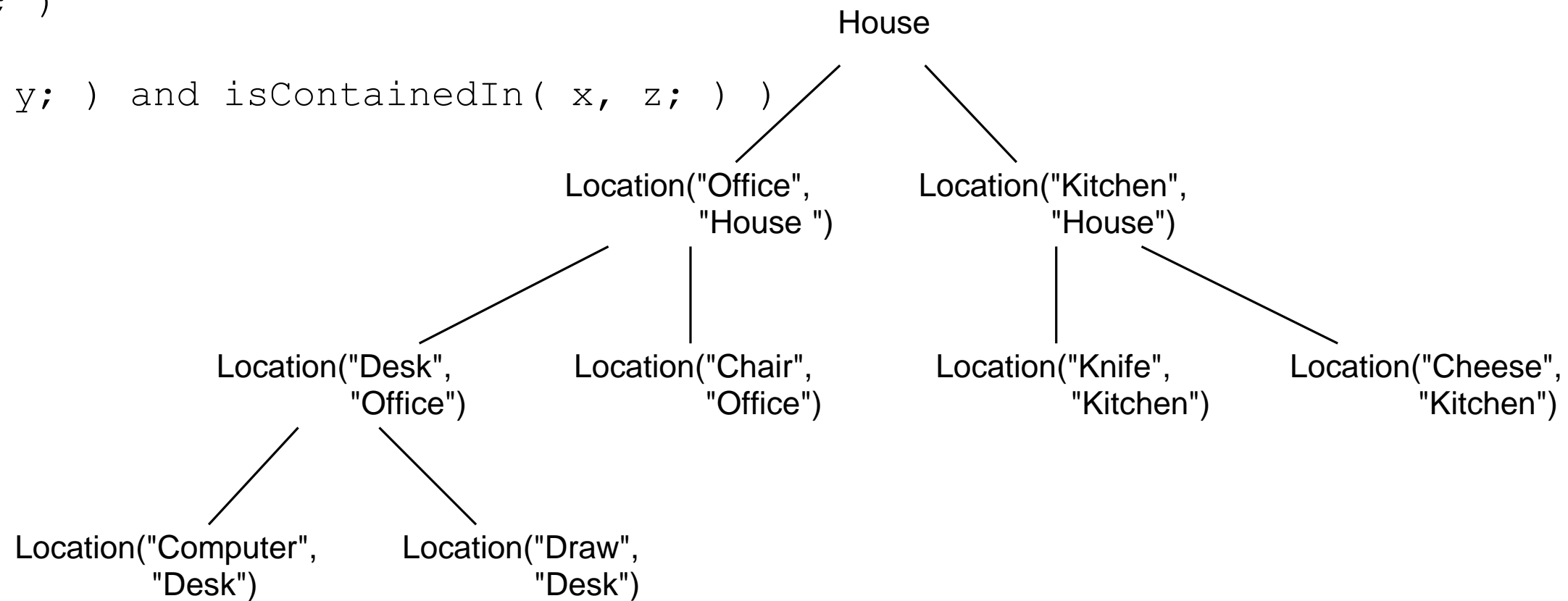
Draw in the House

`isContainedIn(x==Draw, y==House)`

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();

```

go2

Draw in the House

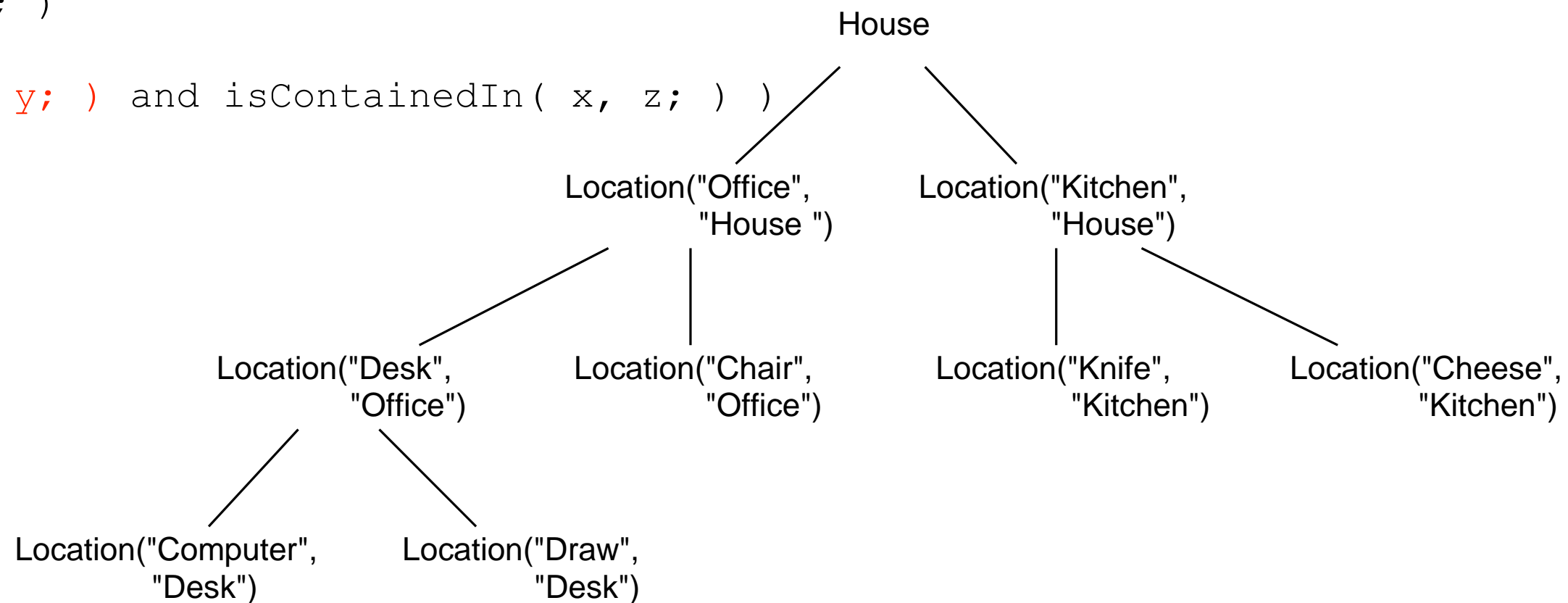
`isContainedIn(x==Draw, y==House)`

`Location(z==Office, y==House)`

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();

```

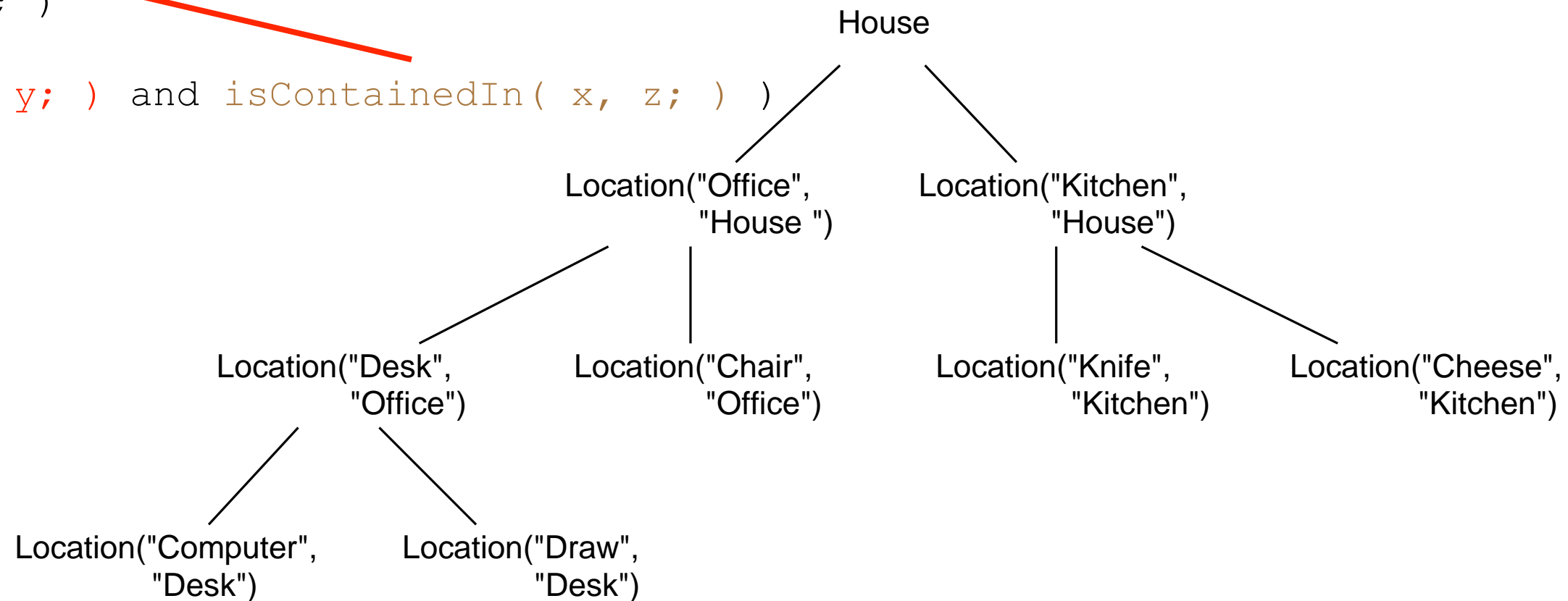
go2

Draw in the House

`isContainedIn(x==Draw, y==House)`

`Location(z==Office, y==House)`

`isContainedIn(x==Draw, z==Office)`



Backward Chaining

```
rule "go2"
```

```
when
```

```
String( this == "go2" )
isContainedIn("Draw", "House"; )
```

```
then
```

```
System.out.println( "Draw in the House" );
```

```
end
```

```
query isContainedIn( String x, String y )
```

```
Location( x, y; )
```

```
or
```

```
( Location( z, y; ) and isContainedIn( x, z; ) ) isContainedIn( x==Draw, z==Kitchen)
```

```
end
```

```
ksession.insert( "go2" );
```

```
ksession.fireAllRules();
```

```
---
```

```
go2
```

```
Draw in the House
```

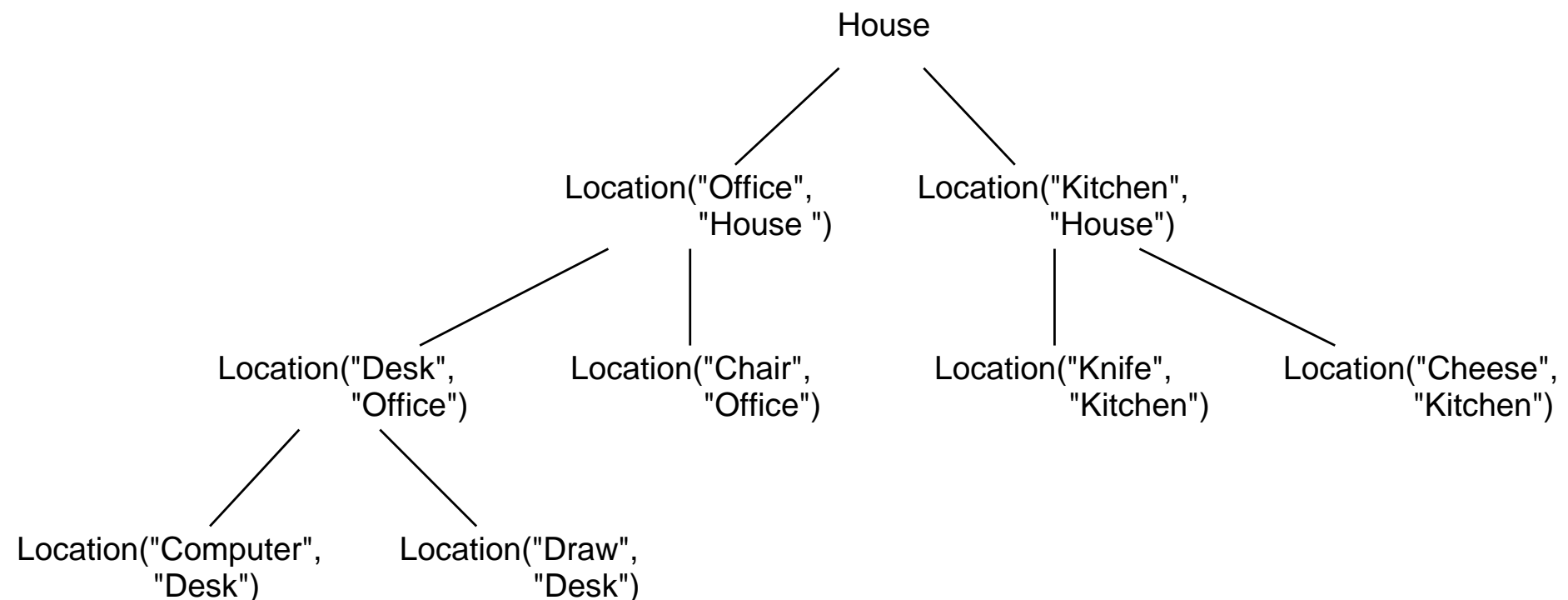
```
isContainedIn(x==Draw, y==House)
```

```
Location(z==Office, y==House)
```

```
isContainedIn(x==Draw, z==Office)
```

```
Location(z==Kitchen, y==House)
```

```
isContainedIn(x==Draw, z==Kitchen)
```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

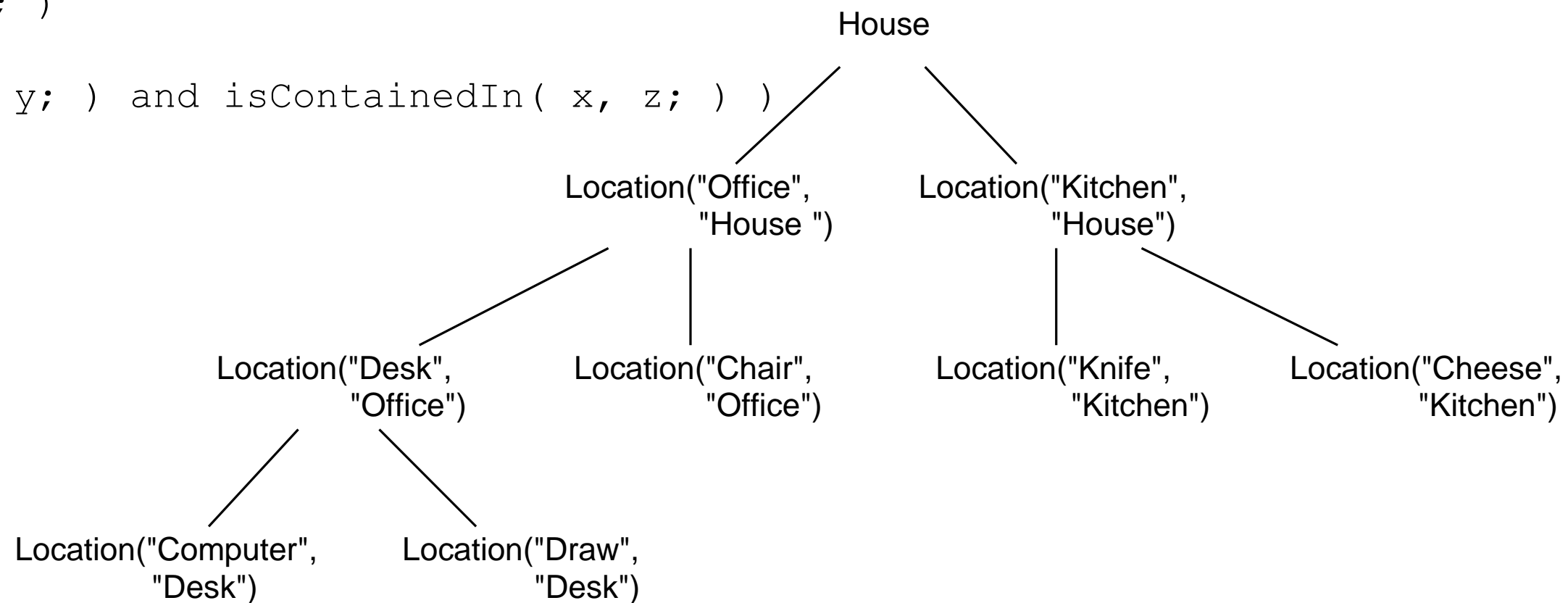
```

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

isContainedIn(x==Draw, y==Office)



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

```

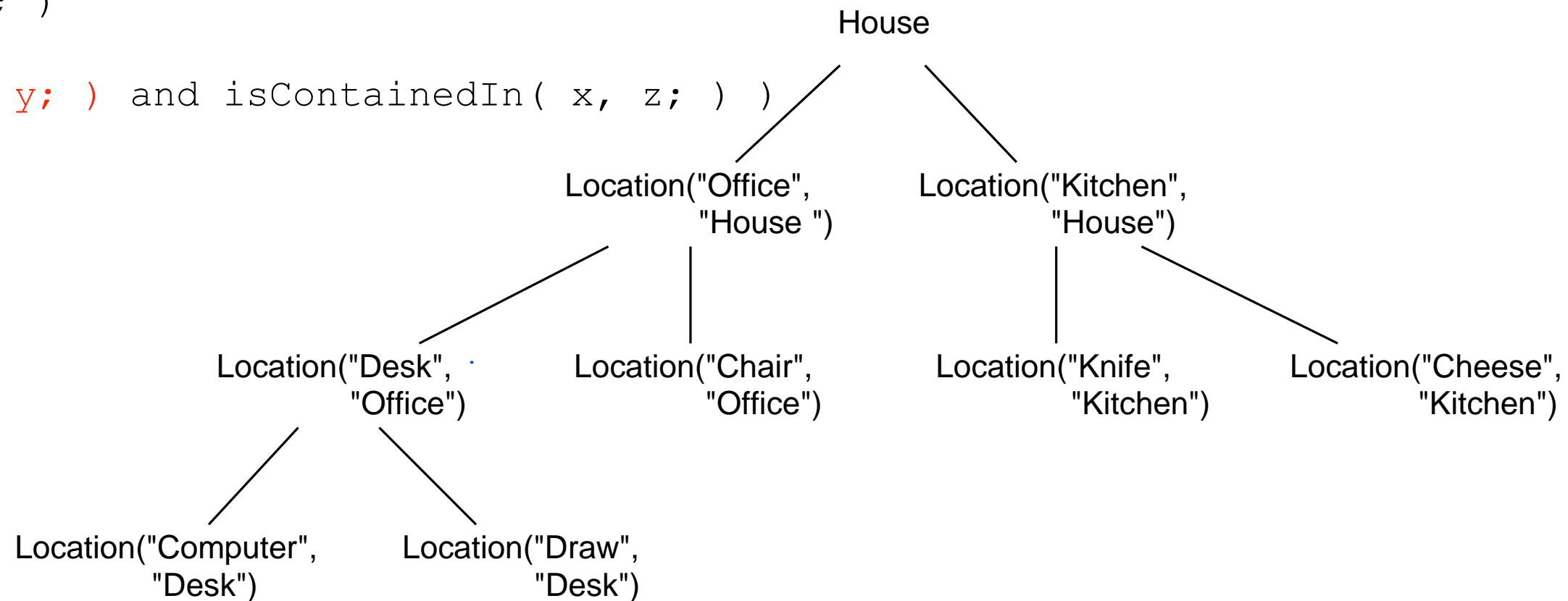
query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

isContainedIn(x==Draw, y==Office)
Location(z==Desk, y==Office)

```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

```

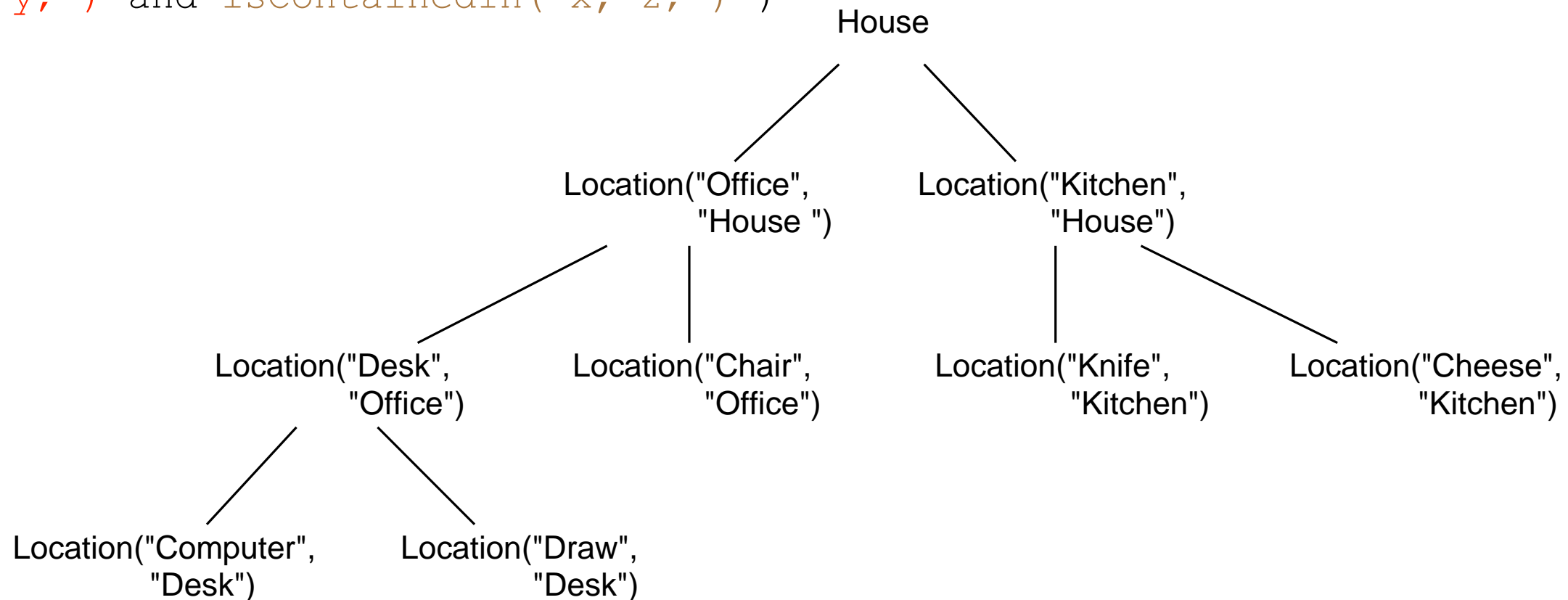
query isContainedIn( String x, String y )
    Location( x, y; )
    or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

isContainedIn(x==Draw, y==Office)
Location(z==Desk, y==Office)
isContainedIn(x==Draw, z==Desk)

```



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

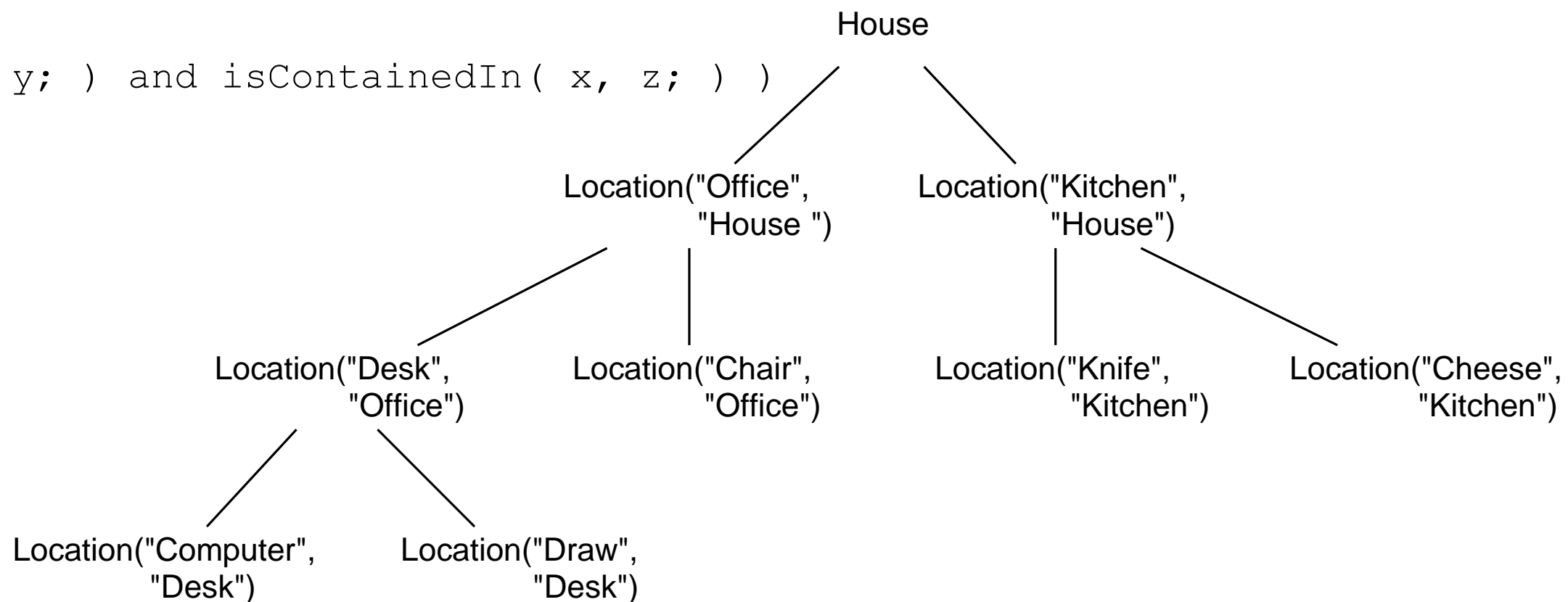
```

```

query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

isContainedIn(x==Draw, y==Desk)



Backward Chaining

```

rule "go2"
when
    String( this == "go2" )
    isContainedIn("Draw", "House"; )
then
    System.out.println( "Draw in the House" );
end

```

```

ksession.insert( "go2" );
ksession.fireAllRules();
---
go2
Draw in the House

```

```

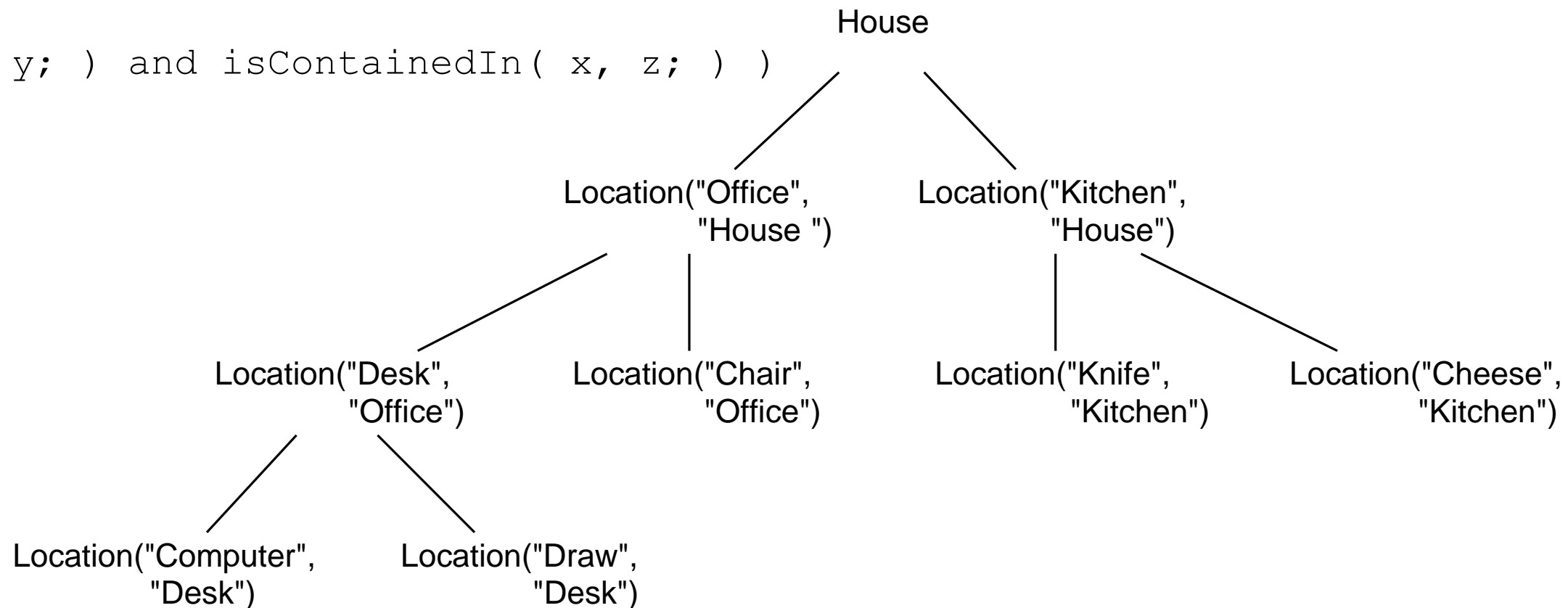
query isContainedIn( String x, String y )
    Location( x, y; )
or
    ( Location( z, y; ) and isContainedIn( x, z; ) )
end

```

```

isContainedIn(x==Draw, y==Desk)
Location(x==Draw, y==Desk)

```

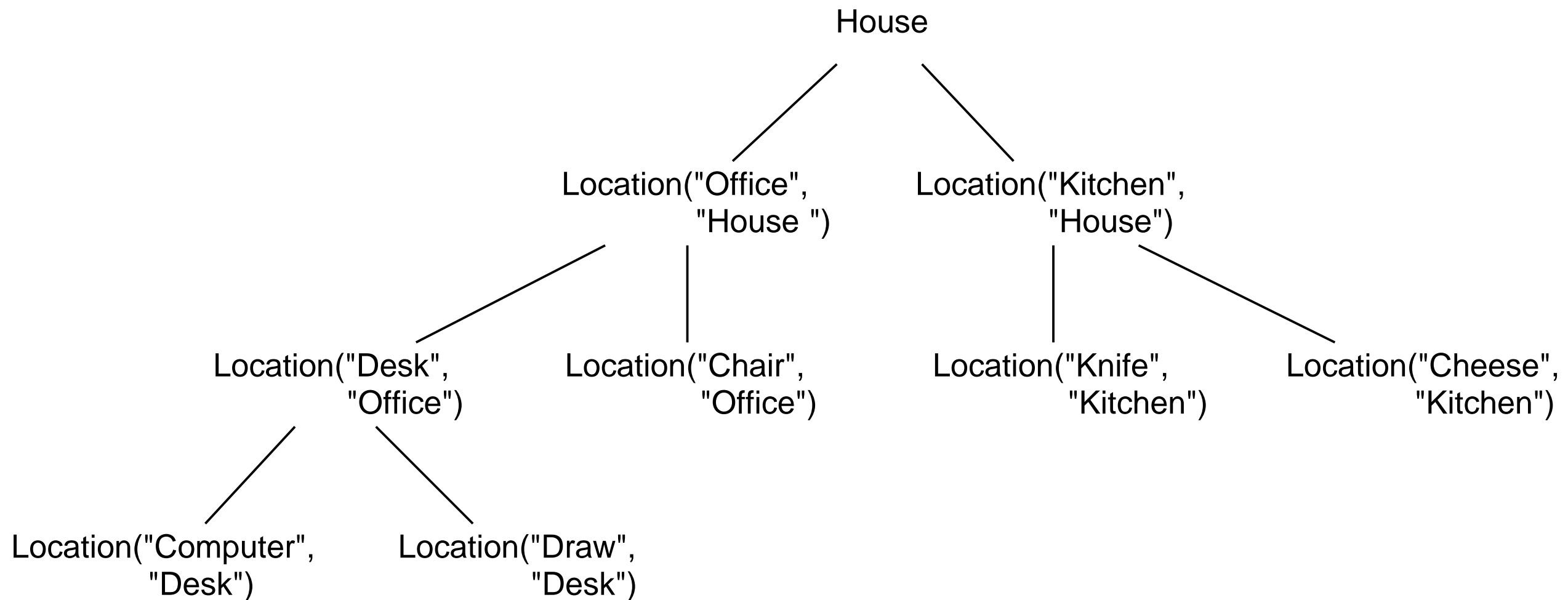


Backward Chaining

```

rule "go3"
when
    String( this == "go3" )
    isContainedIn("Key", "Office"; )
then
    System.out.println( "Key in the Office" );
end

```



Backward Chaining

```

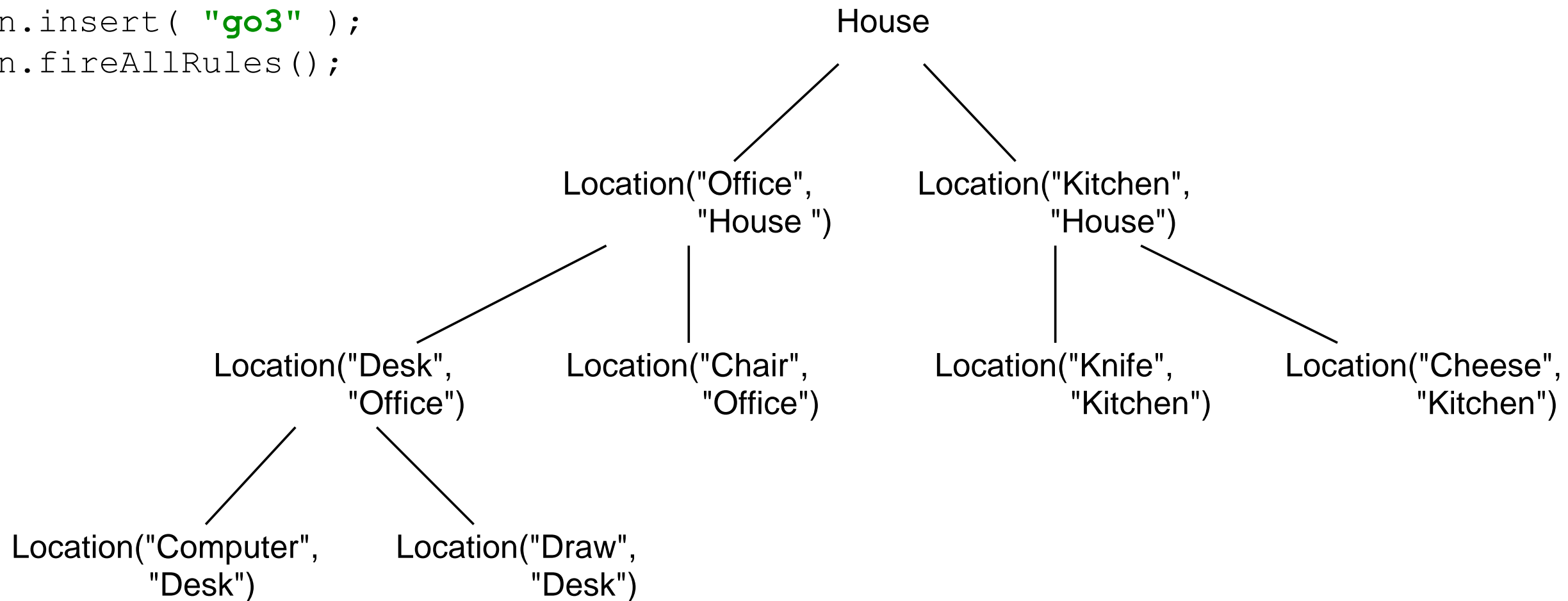
rule "go3"
when
    String( this == "go3" )
    isContainedIn("Key", "Office"; )
then
    System.out.println( "Key in the Office" );
end

```

```

ksession.insert( "go3" );
ksession.fireAllRules();
---
go3

```

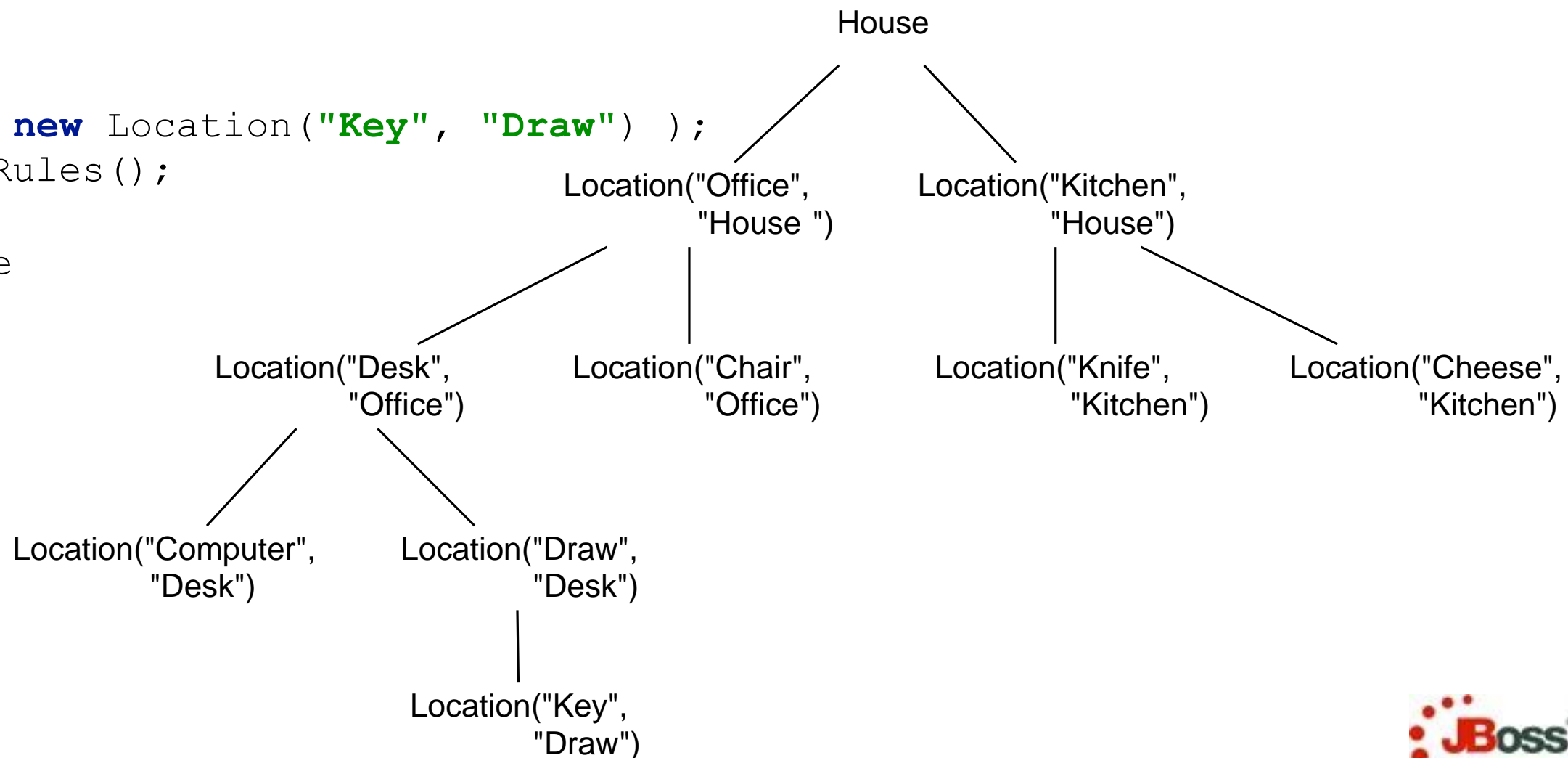


Backward Chaining

```
rule "go3"
when
    String( this == "go3" )
    isContainedIn("Key", "Office"; )
then
    System.out.println( "Key in the Office" );
end
```

```
ksession.insert( "go3" );
ksession.fireAllRules();
---
go3
```

```
ksession.insert( new Location("Key", "Draw") );
ksession.fireAllRules();
---
Key in the Office
```

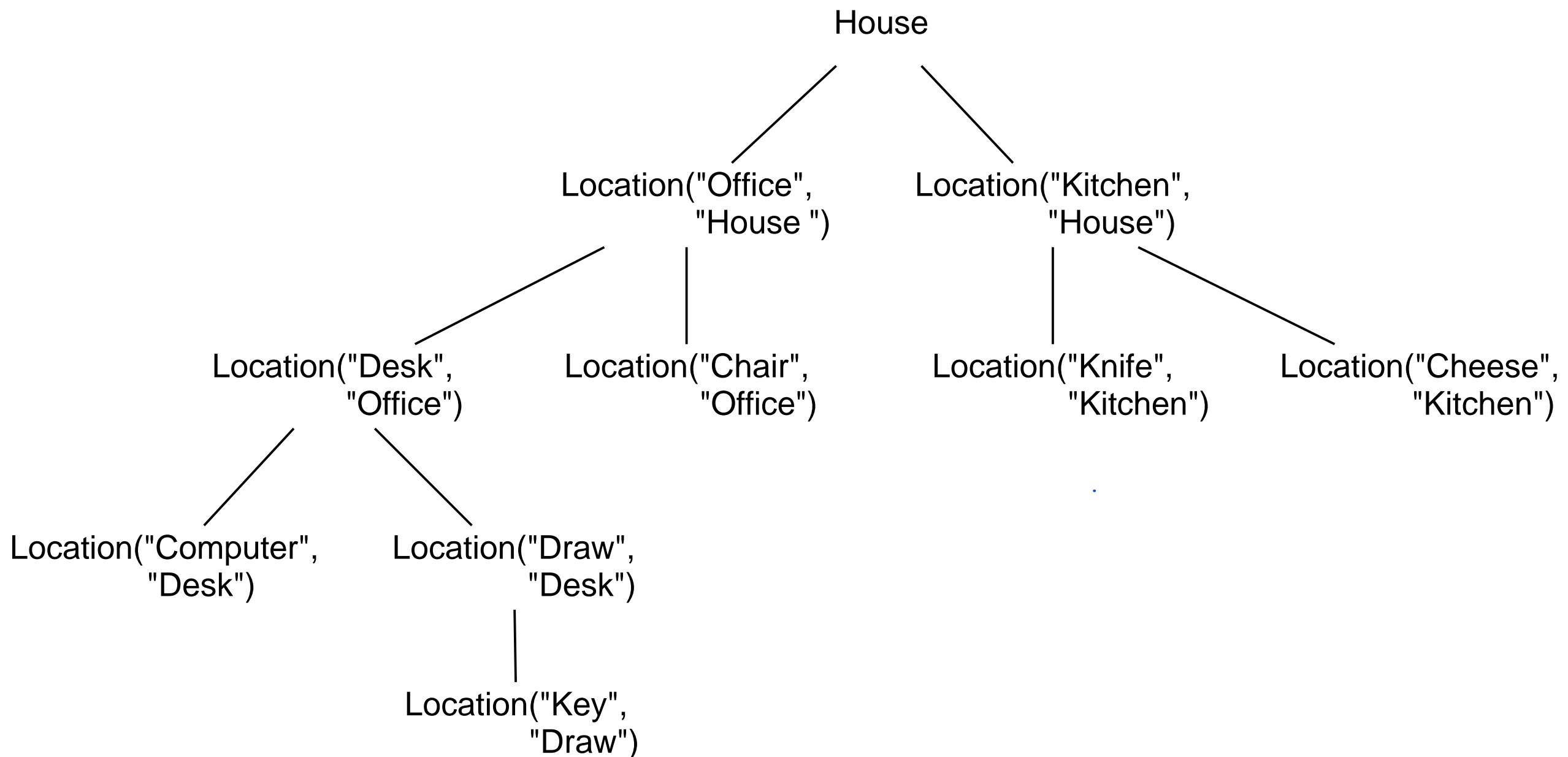


Backward Chaining

```

rule "go4"
when
    String( this == "go4" )
    isContainedIn(thing, "Office"; )
then
    System.out.println( "thing " + thing + " is in the Office" );
end

```



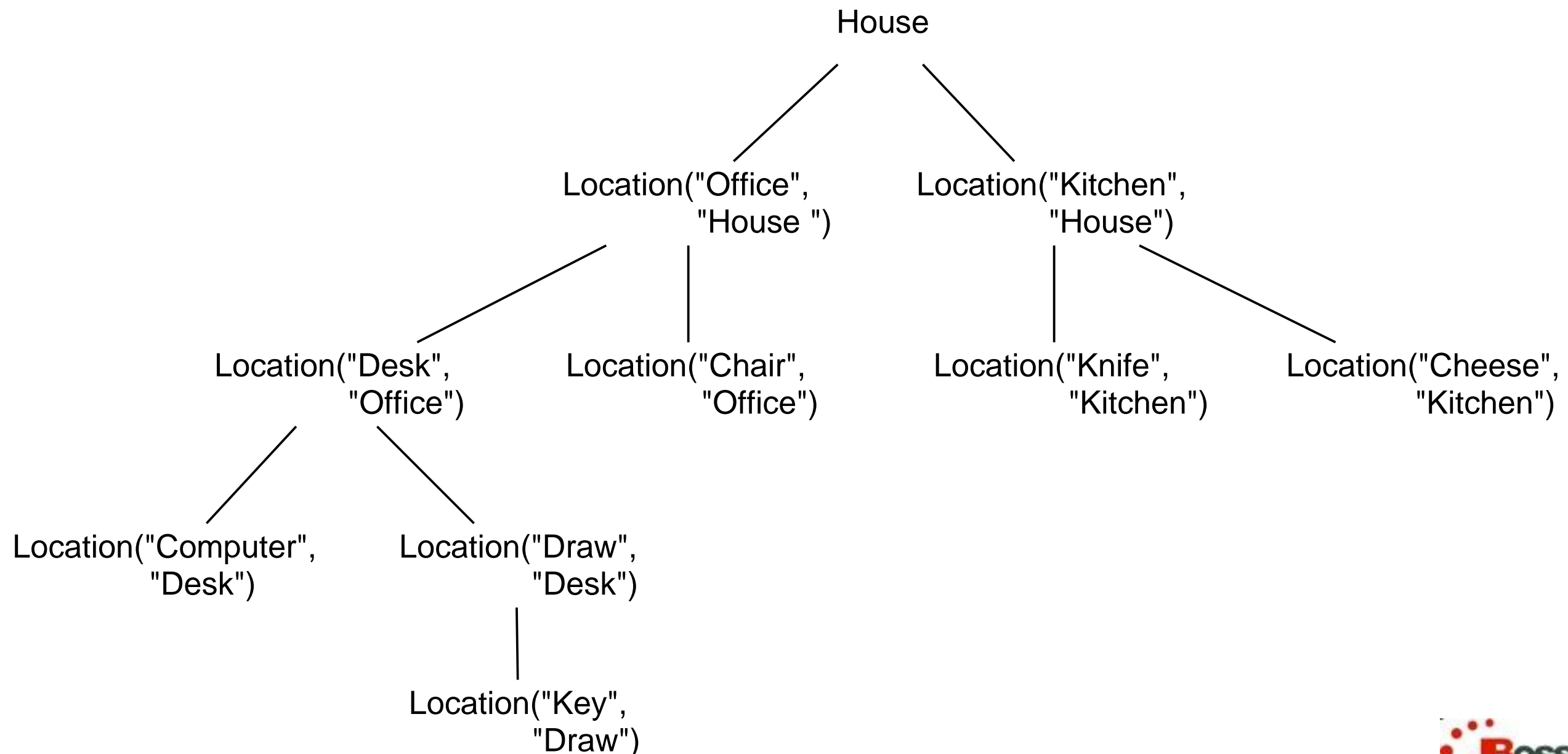
Backward Chaining

```

rule "go4"
when
    String( this == "go4" )
    isContainedIn(thing, "Office"; )
then
    System.out.println( "thing " + thing + " is in the Office" );
end

```

Out Var
(unbound)



Backward Chaining

```

rule "go4"
when
    String( this == "go4" )
    isContainedIn(thing, "Office"; )
then
    System.out.println( "thing " + thing + " is in the Office" );
end

```

Out Var
(unbound)

```

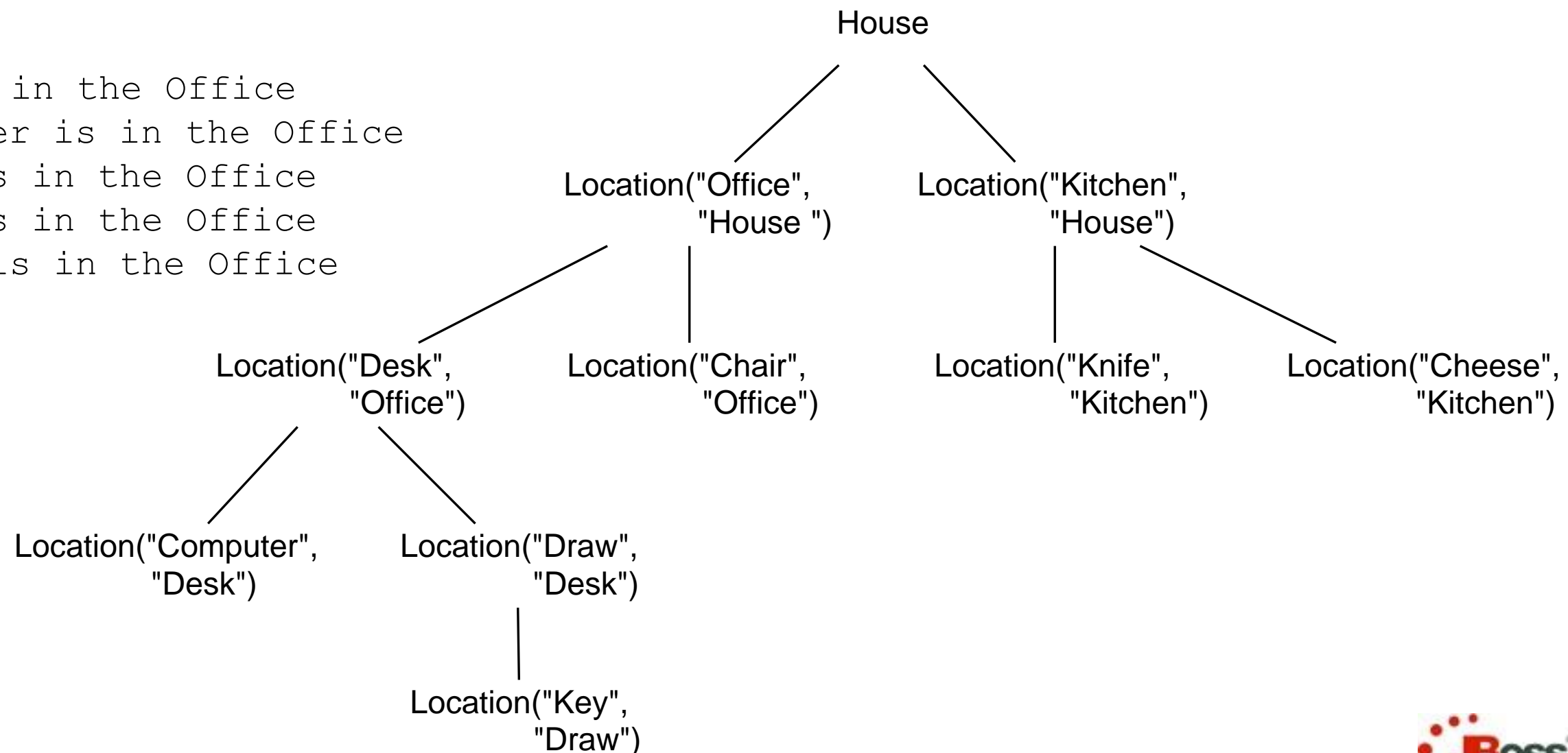
ksession.insert( "go4" );
ksession.fireAllRules();

```

```

go4
thing Key is in the Office
thing Computer is in the Office
thing Draw is in the Office
thing Desk is in the Office
thing Chair is in the Office

```

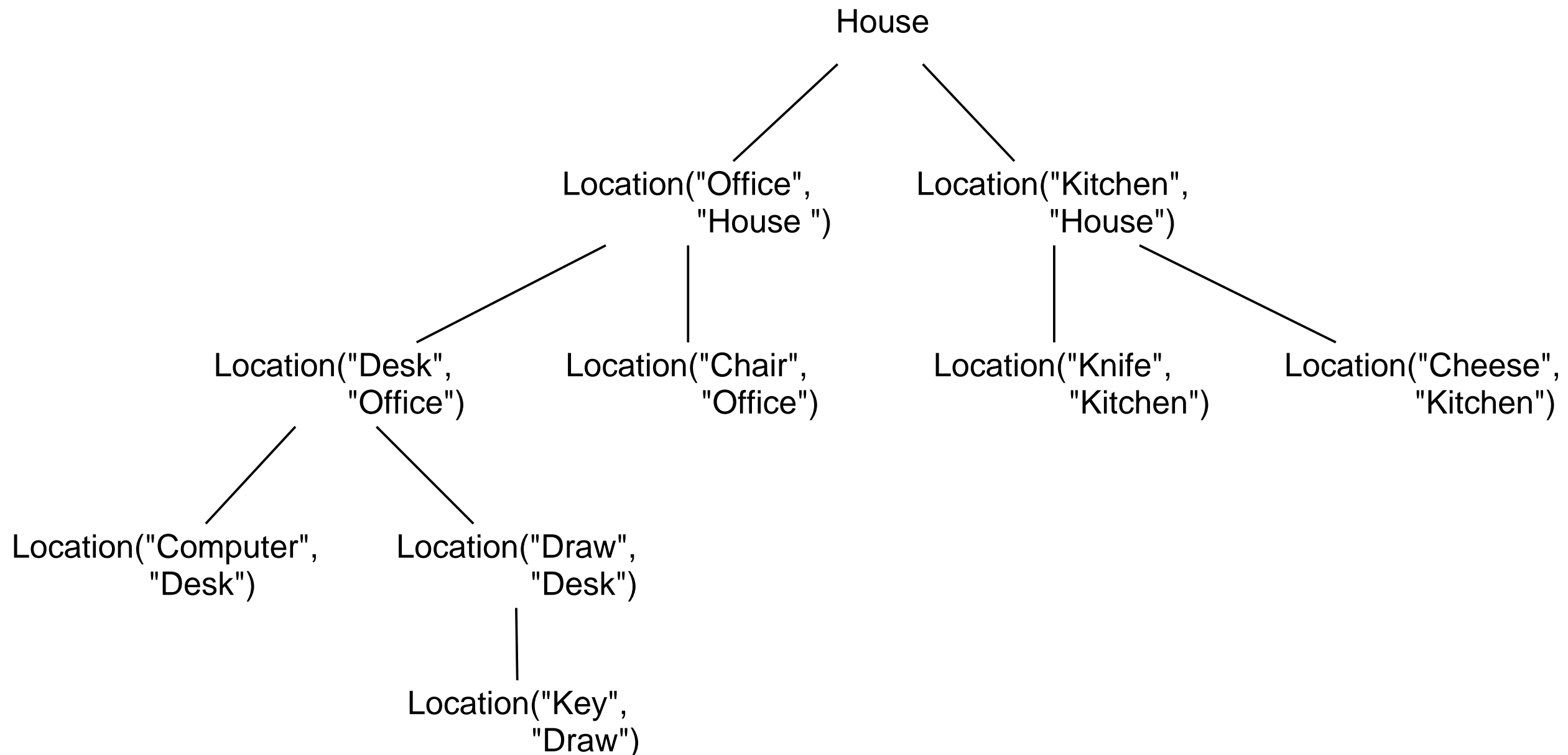


Backward Chaining

```

rule "go5"
when
    String( this == "go5" )
    isContainedIn(thing, location; )
then
    System.out.println( "thing " + thing + " is in " + location );
end

```



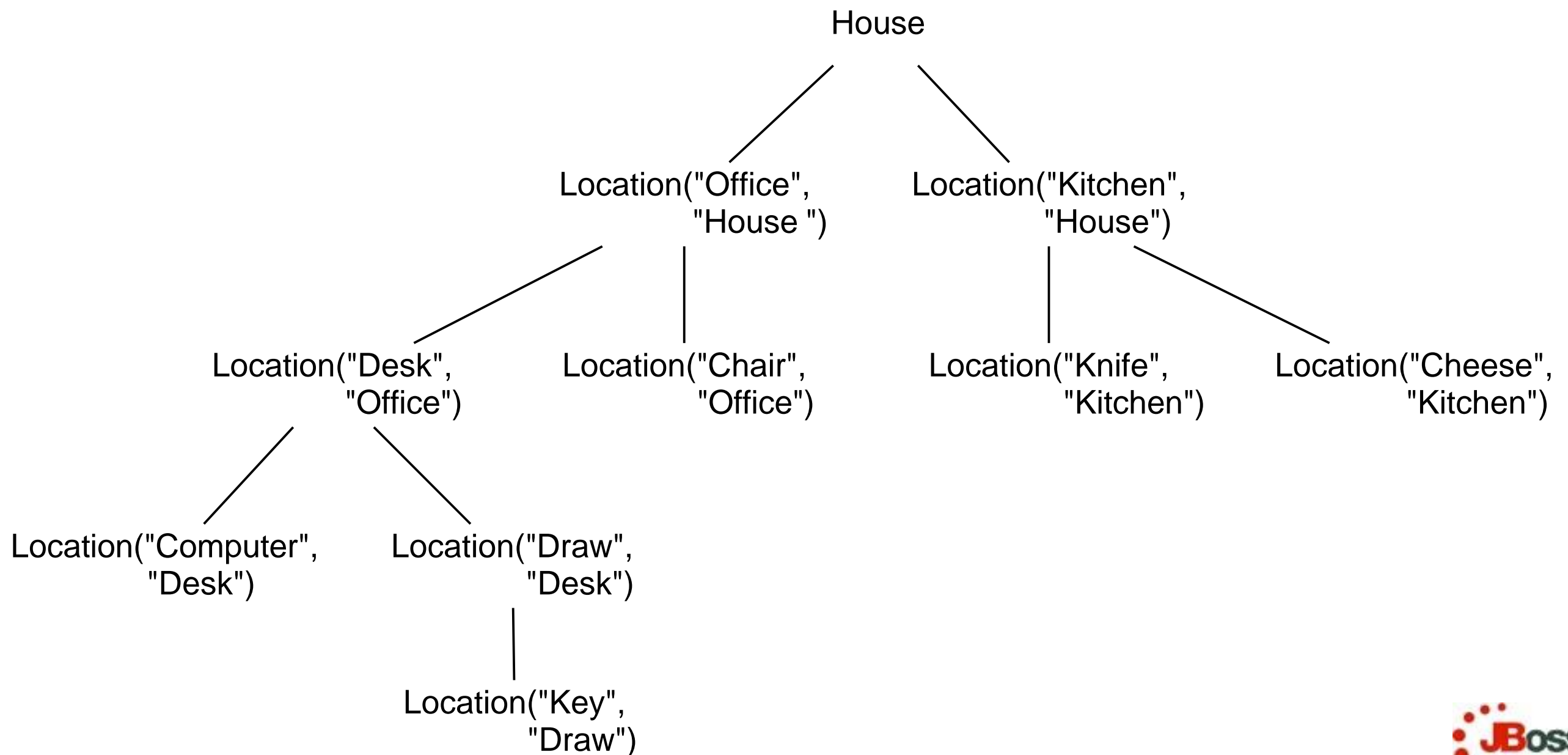
Backward Chaining

```

rule "go5"
when
    String( this == "go5" )
    isContainedIn(thing, location; )
then
    System.out.println( "thing " + thing + " is in " + location );
end

```

Out Var
(unbound)



Backward Chaining

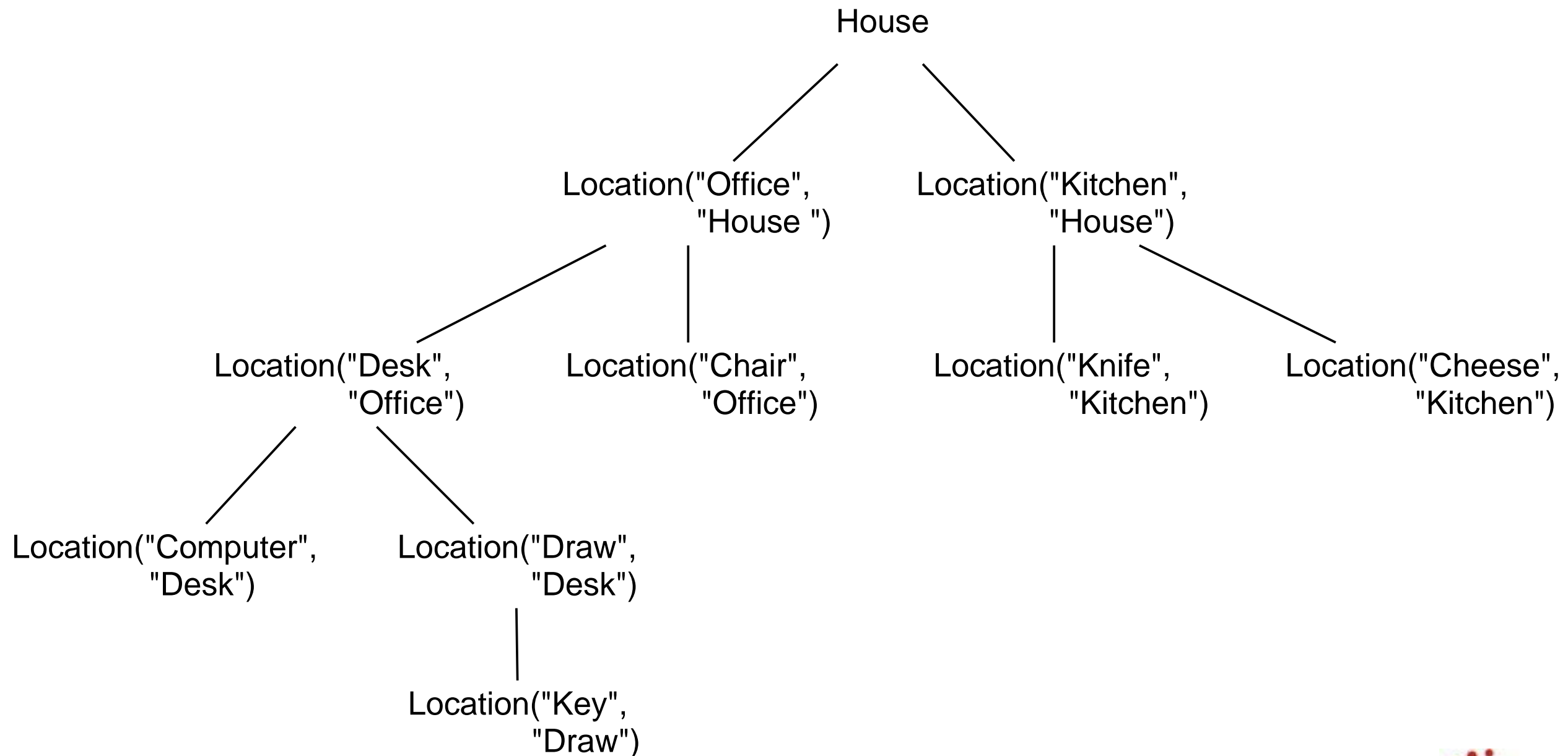
```

rule "go5"
when
    String( this == "go5" )
    isContainedIn(thing, location; )
then
    System.out.println( "thing " + thing + " is in " + location );
end

```

Out Var
(unbound)

Out Var
(unbound)



Backward Chaining

```

rule "go5"
when
    String( this == "go5" )
    isContainedIn(thing, location; )
then
    System.out.println( "thing " + thing + " is in " + location );
end

```

Out Var
(unbound)

Out Var
(unbound)

```

ksession.insert( "go5" );
ksession.fireAllRules();

```

go5

```

thing Knife is in House
thing Cheese is in House
thing Key is in House
thing Computer is in House
thing Draw is in House
thing Desk is in House
thing Chair is in House
thing Key is in Office
thing Computer is in Office
thing Draw is in Office
thing Key is in Desk
thing Office is in House

```

```

thing Computer is in Desk
thing Knife is in Kitchen
thing Cheese is in Kitchen
thing Kitchen is in House
thing Key is in Draw
thing Draw is in Desk
thing Desk is in Office
thing Chair is in Office

```

