Rule-Based Systems: Logic Programming

MSc Business Information Systems

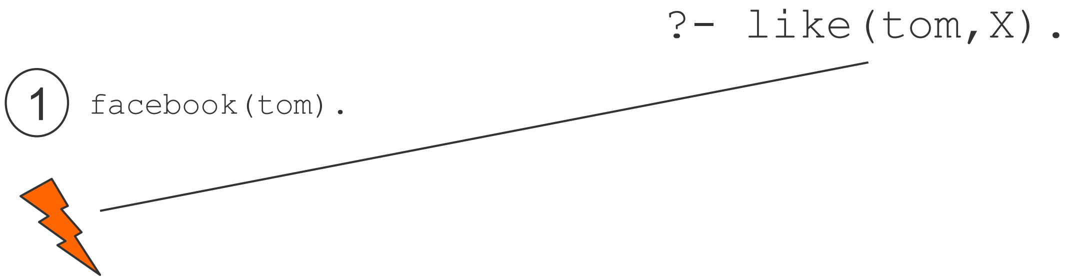
Inference Procedure for Logic Programming

```
Let resolvent be the query ?- $Q_1, \ldots, Q_m$
While resolvent is not empty do
    1. Choose a query literal $Q_i$ from resolvent.
    2. Choose a renamed[1] clause $H$ :- $B_1, \ldots, B_n$ from $P$ such that
       $Q_i$ and $H$ unify with an most general unifier $\sigma$ , i.e. $Q_i\sigma = H\sigma$
    3. If no such $Q_i$ and clause exist, then backtrack
    4. Remove $Q_i$ from the resolvent
    5. Add $B_1, \ldots, B_n$ to the resolvent
    6. Add $\sigma$ to $\sigma_{all}$
    7. Apply substitution $\sigma$ to the resolvent.
If resolvent is empty, return $\sigma_{all}$, else return failure.
```

[1] Renaming means that the variables in the clause get new unique identifiers

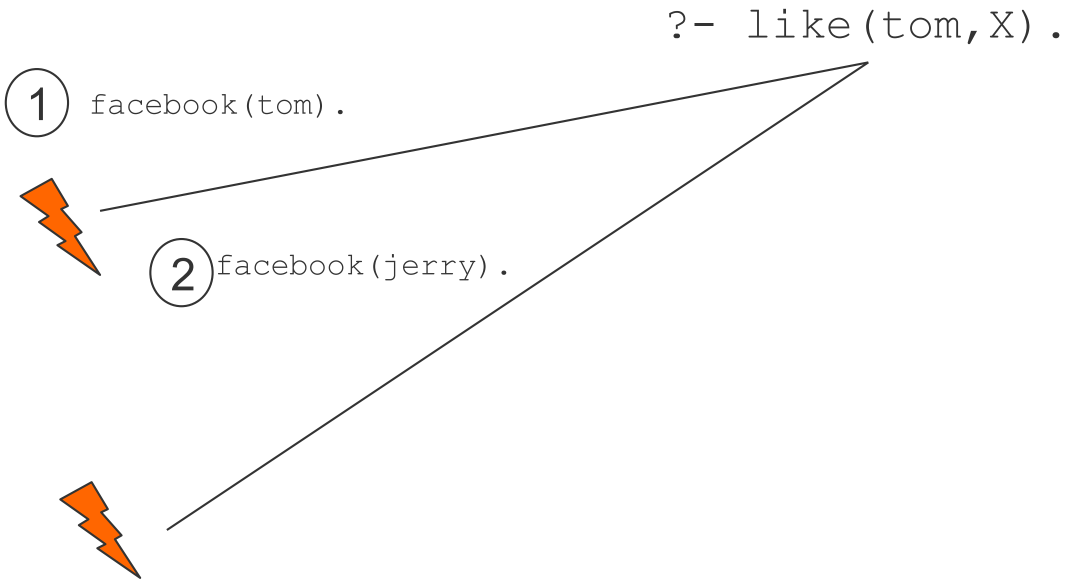
Reasoning Example

```
facebook(tom).
facebook(jerry).
facebook_connect(tom,jerry).


like(peter,john).
like(X,mary).
like(X,Y) :- facebook(X),facebook(Y),facebook_connect(X,Y).


?- like(tom,X).
?- like(tom,X).
```



```
?- like(tom,X).
```

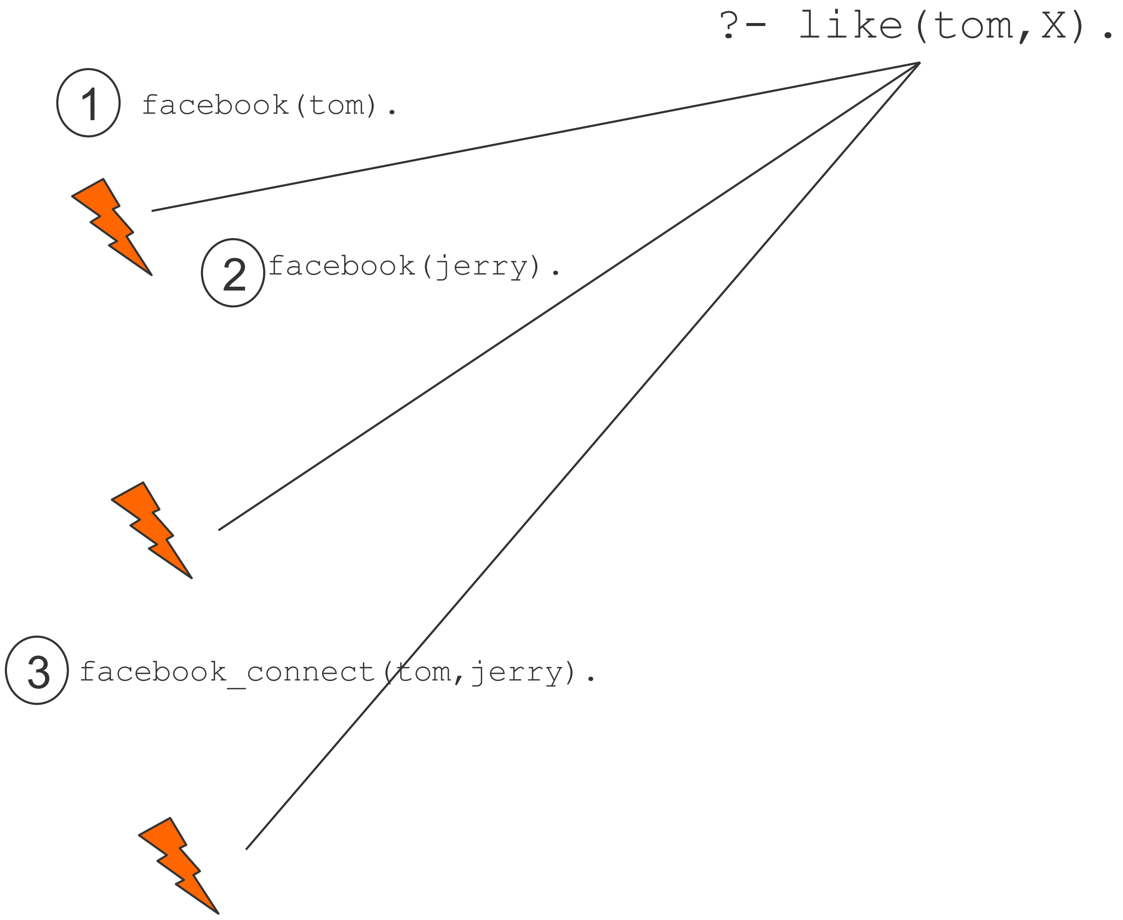
1 `facebook(tom).`



```
?- like(tom,X).
```

1 `facebook(tom).`

2 `facebook(jerry).`

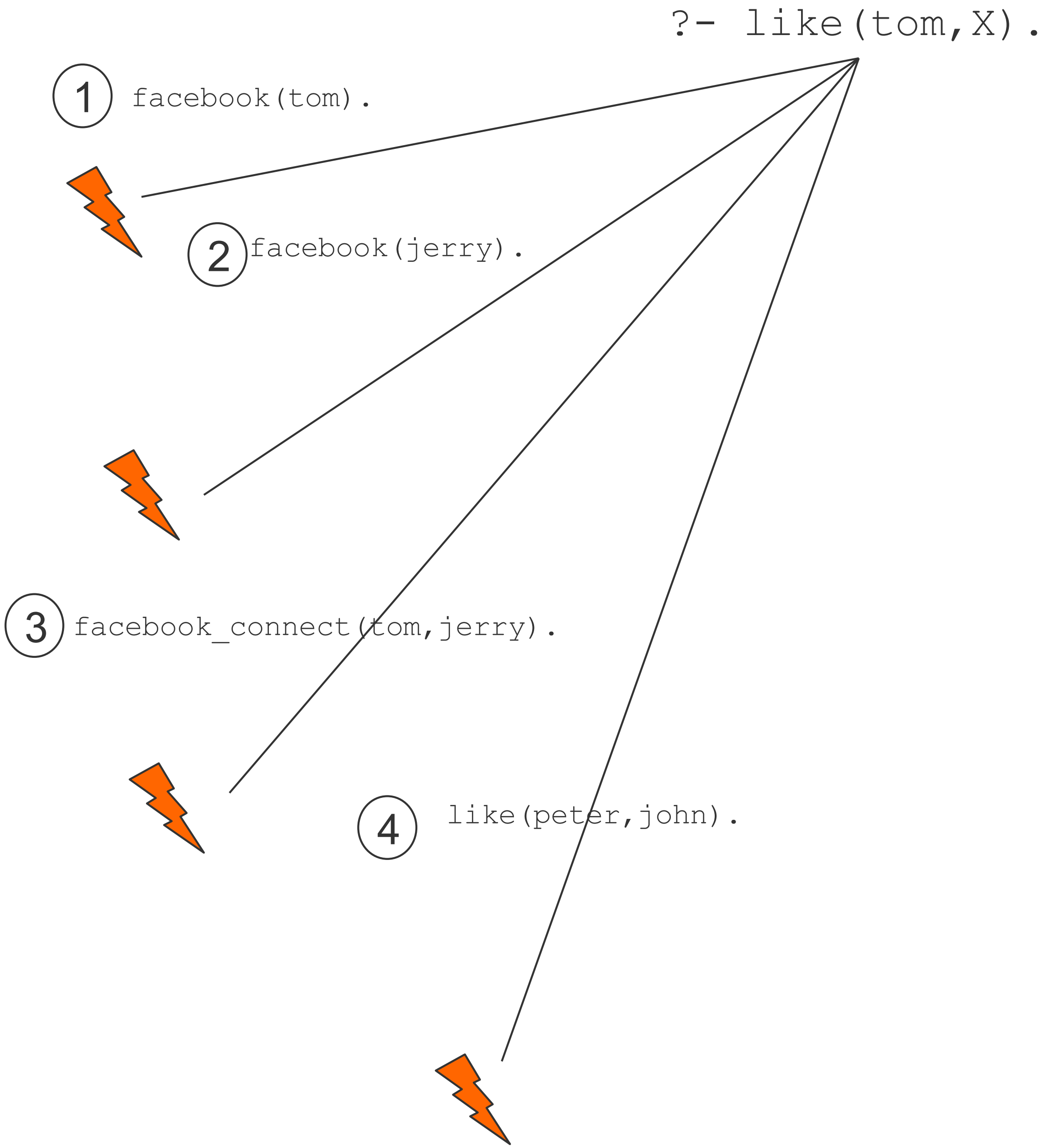


```
?- like(tom,X).
```

1 `facebook(tom).`

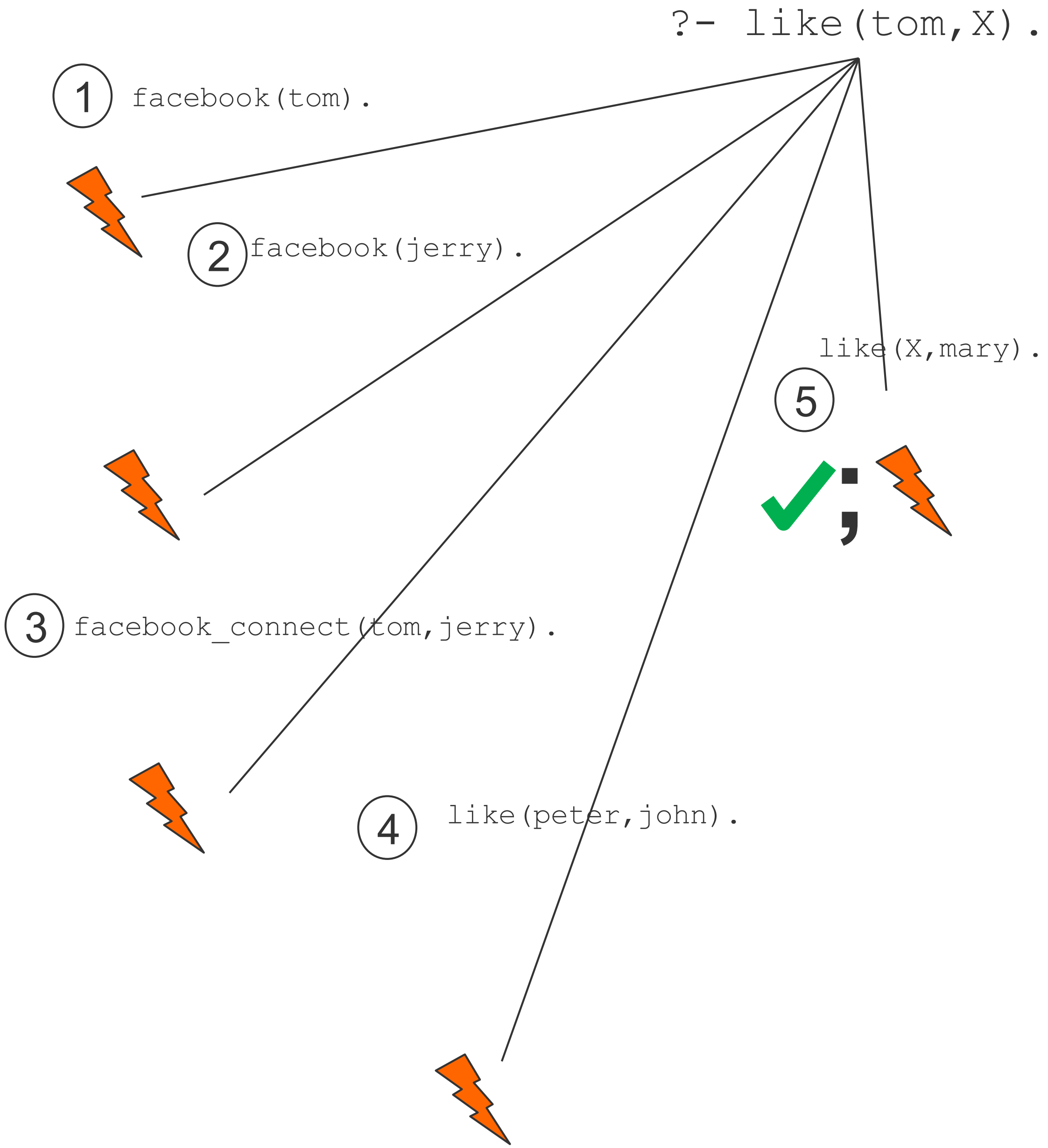
2 `facebook(jerry).`

3 `facebook_connect(tom,jerry).`



```
?- like(tom,X).
```

```
1 facebook(tom).
2 facebook(jerry).
3 facebook_connect(tom,jerry).
4 like(peter,john).
```



```
?- like(tom,X).
```

1 `facebook(tom).`

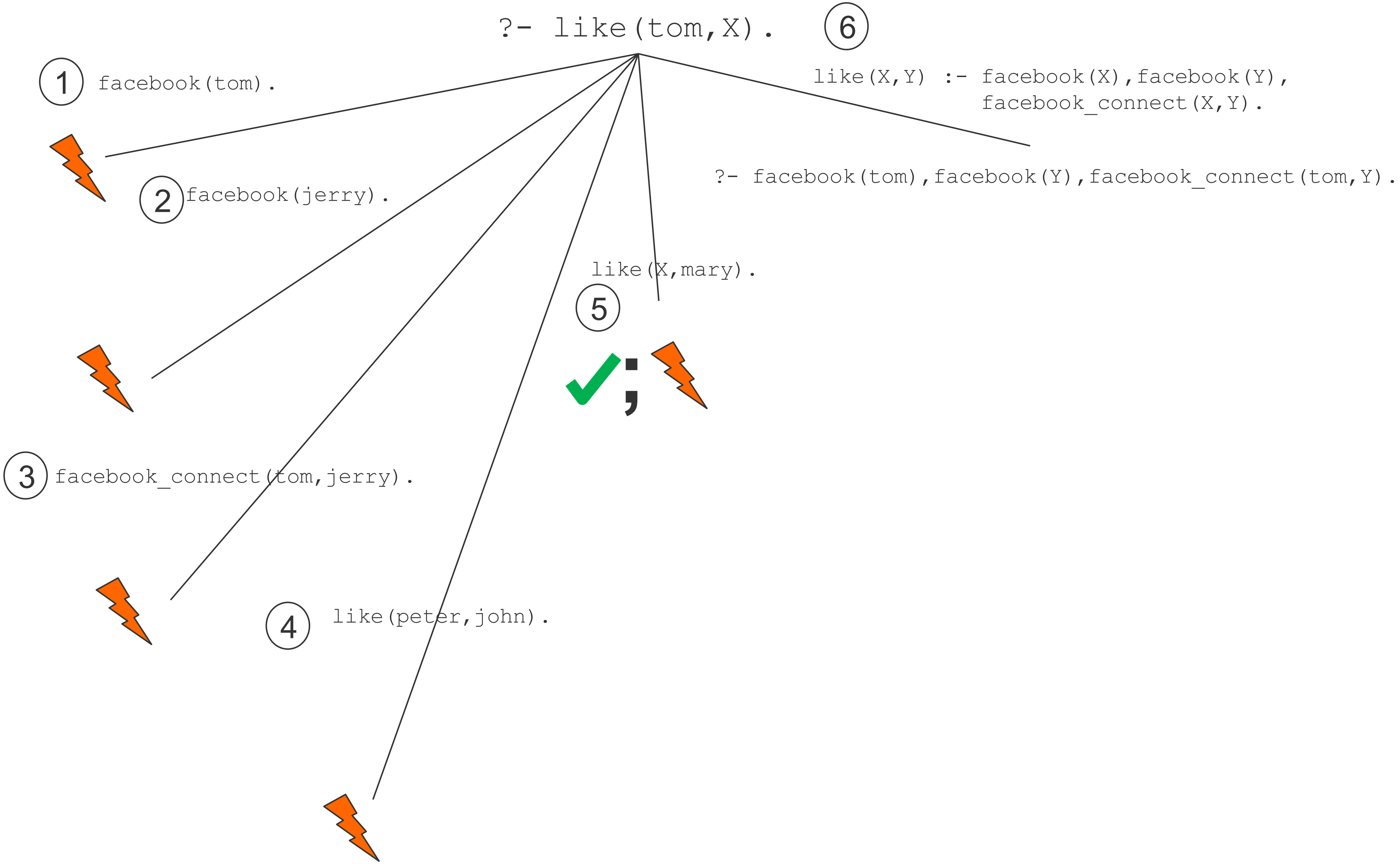
2 `facebook(jerry).`

3 `facebook_connect(tom,jerry).`

4 `like(peter,john).`

5 `like(X,mary).`

✓;



```
?- like(tom,X).
```

1 `facebook(tom).`

2 `facebook(jerry).`

3 `facebook_connect(tom,jerry).`

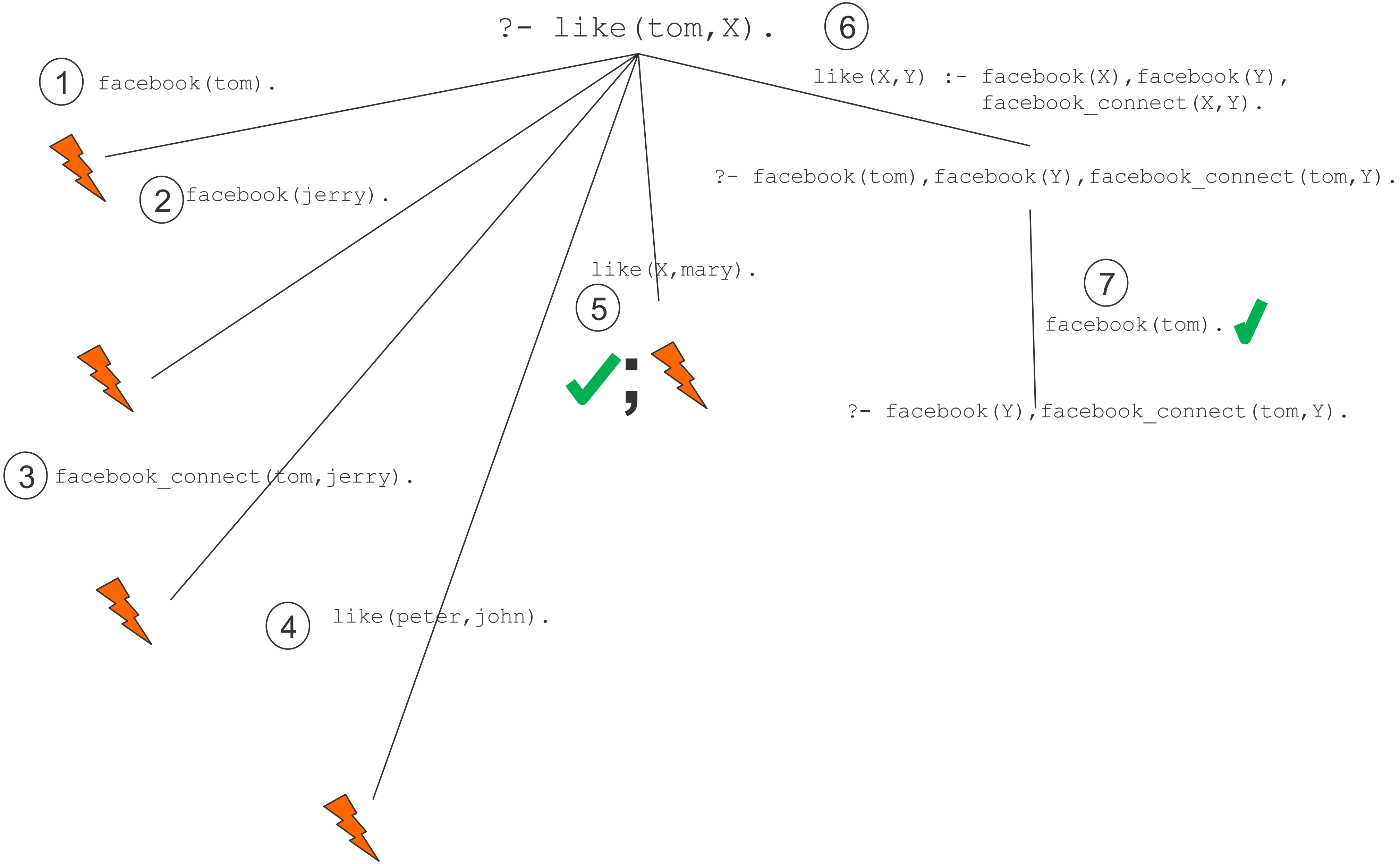
4 `like(peter,john).`

5 `like(X,mary).`

✓;

6
```
like(X,Y) :- facebook(X),facebook(Y),
             facebook_connect(X,Y).
```

```
?- facebook(tom),facebook(Y),facebook_connect(tom,Y).
```



?- like(tom,X).

1 facebook(tom).

2 facebook(jerry).

3 facebook_connect(tom,jerry).

4 like(peter,john).

5 like(X,mary).

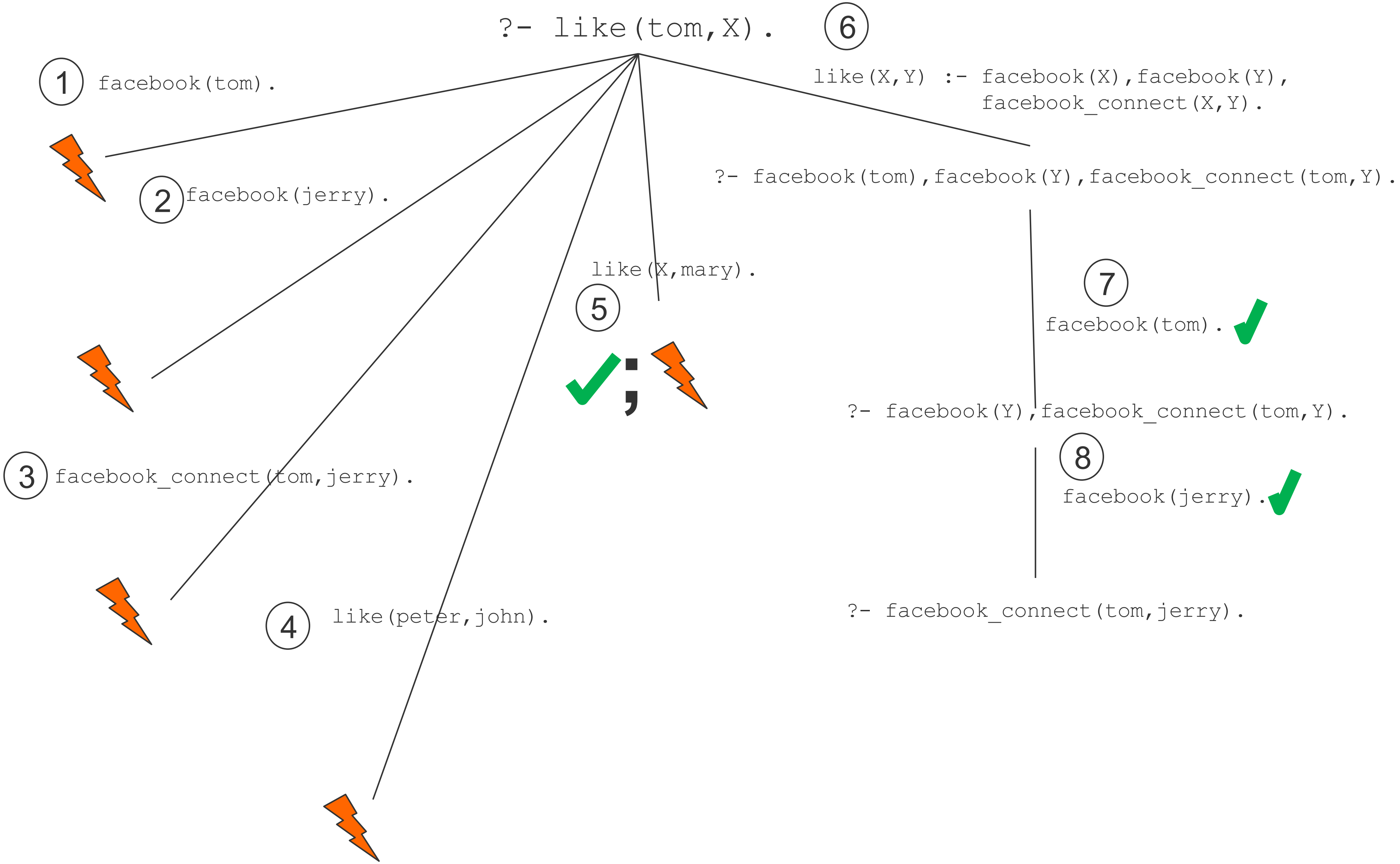
✓; ⚡

6 like(X,Y) :- facebook(X),facebook(Y),
facebook_connect(X,Y).

?- facebook(tom),facebook(Y),facebook_connect(tom,Y).

7 facebook(tom). ✓

?- facebook(Y),facebook_connect(tom,Y).



?- like(tom,X).

1 facebook(tom). ✗

2 facebook(jerry). ✗

3 facebook_connect(tom,jerry). ✗

4 like(peter,john). ✗

5 like(X,mary). ✓ ; ✗

6 like(X,Y) :- facebook(X),facebook(Y),
facebook_connect(X,Y).

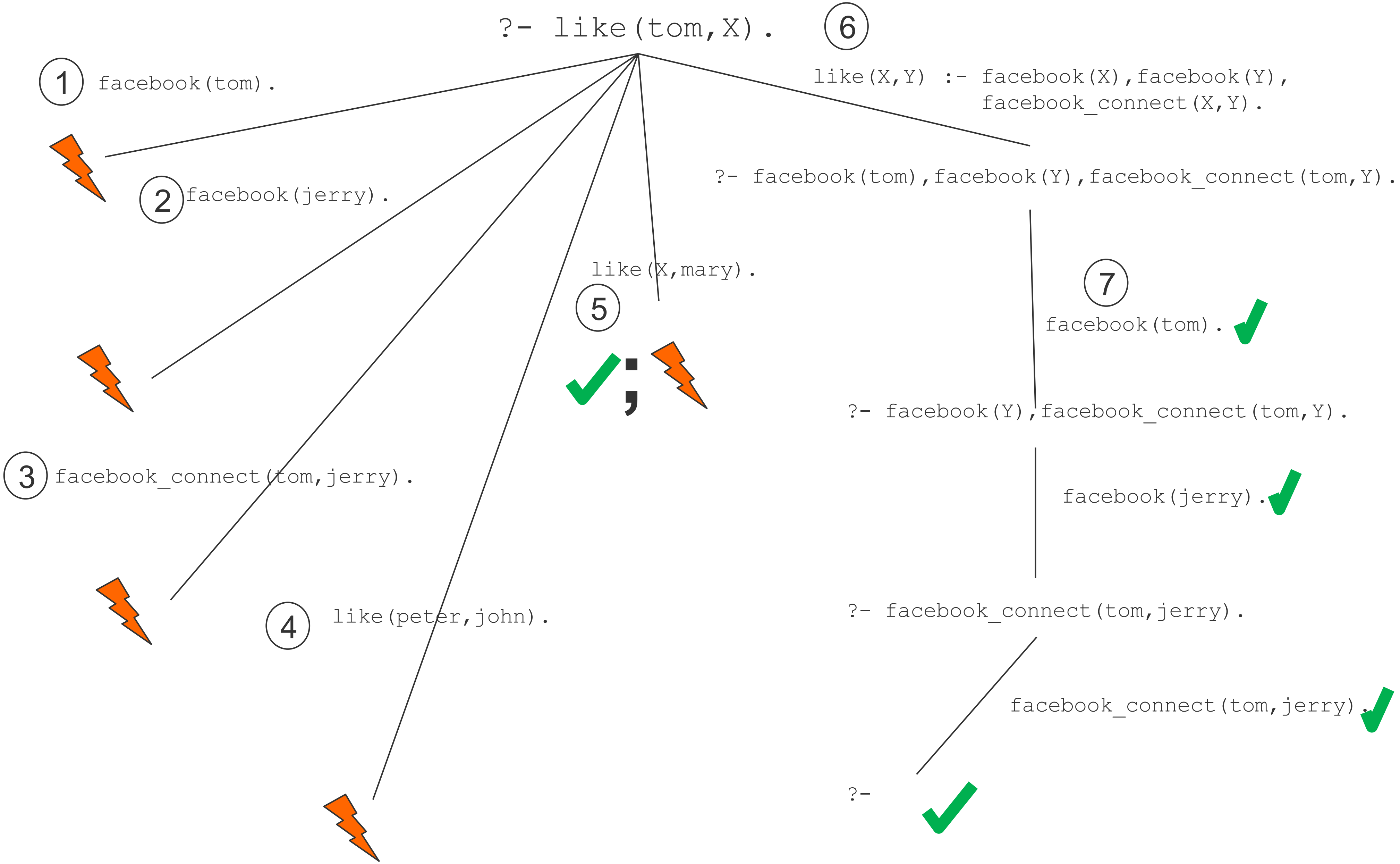
?- facebook(tom),facebook(Y),facebook_connect(tom,Y).

7 facebook(tom). ✓

?- facebook(Y),facebook_connect(tom,Y).

8 facebook(jerry). ✓

?- facebook_connect(tom,jerry).



?- like(tom,X).

1 facebook(tom).

2 facebook(jerry).

3 facebook_connect(tom,jerry).

4 like(peter,john).

5 like(X,mary).

✓ ; ⚡

6 like(X,Y) :- facebook(X),facebook(Y),
facebook_connect(X,Y).

?- facebook(tom),facebook(Y),facebook_connect(tom,Y).

7 facebook(tom). ✓

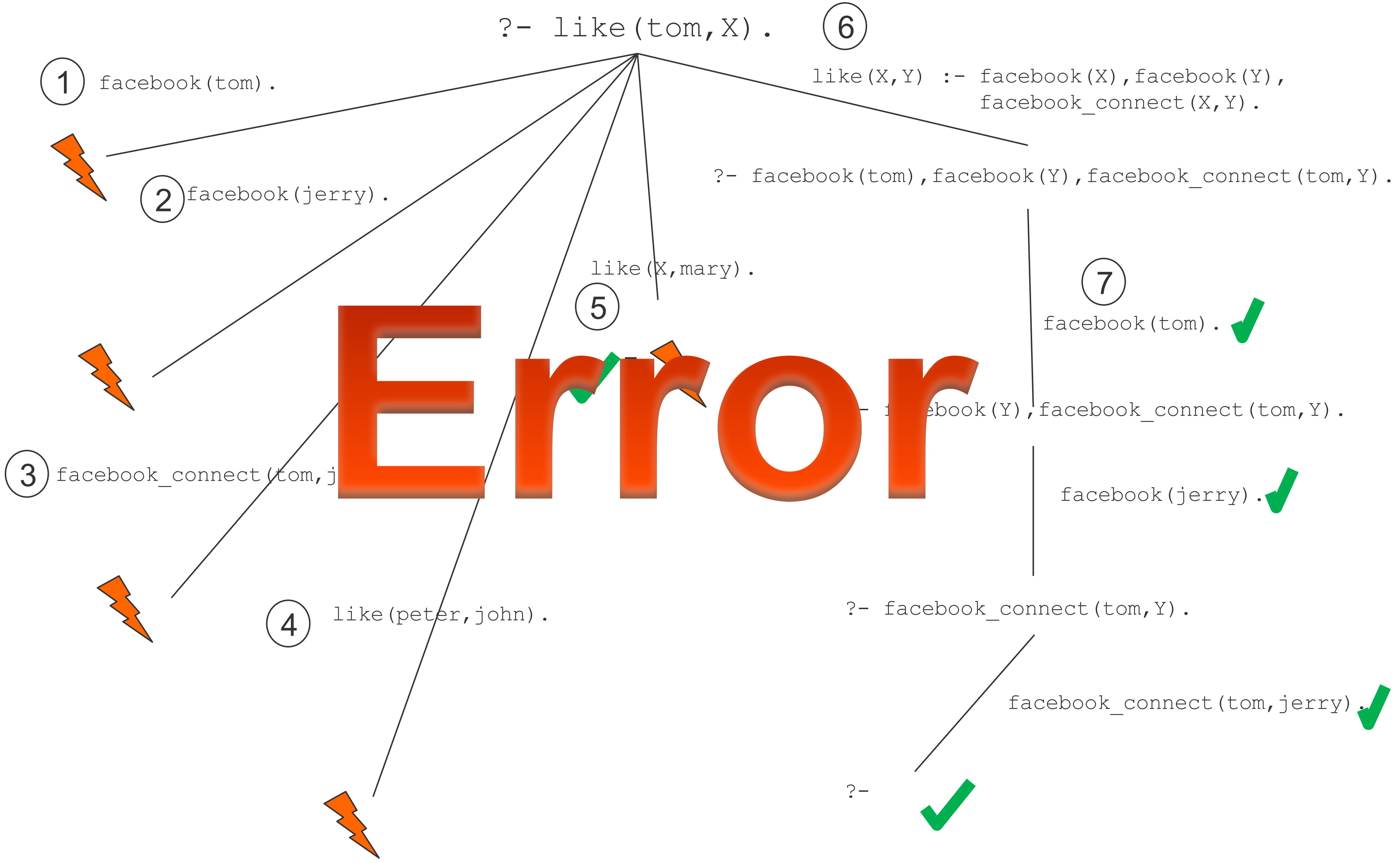
?- facebook(Y),facebook_connect(tom,Y).

facebook(jerry). ✓

?- facebook_connect(tom,jerry).

facebook_connect(tom,jerry). ✓

?- ✓



```
?- like(tom,X).
```

1 facebook(tom).

2 facebook(jerry).

3 facebook_connect(tom,j

4 like(peter,john).

5 like(X,mary).

6 like(X,Y) :- facebook(X),facebook(Y),
facebook_connect(X,Y).

?- facebook(tom),facebook(Y),facebook_connect(tom,Y).

7 facebook(tom). ✓

– f ebook(Y),facebook_connect(tom,Y).

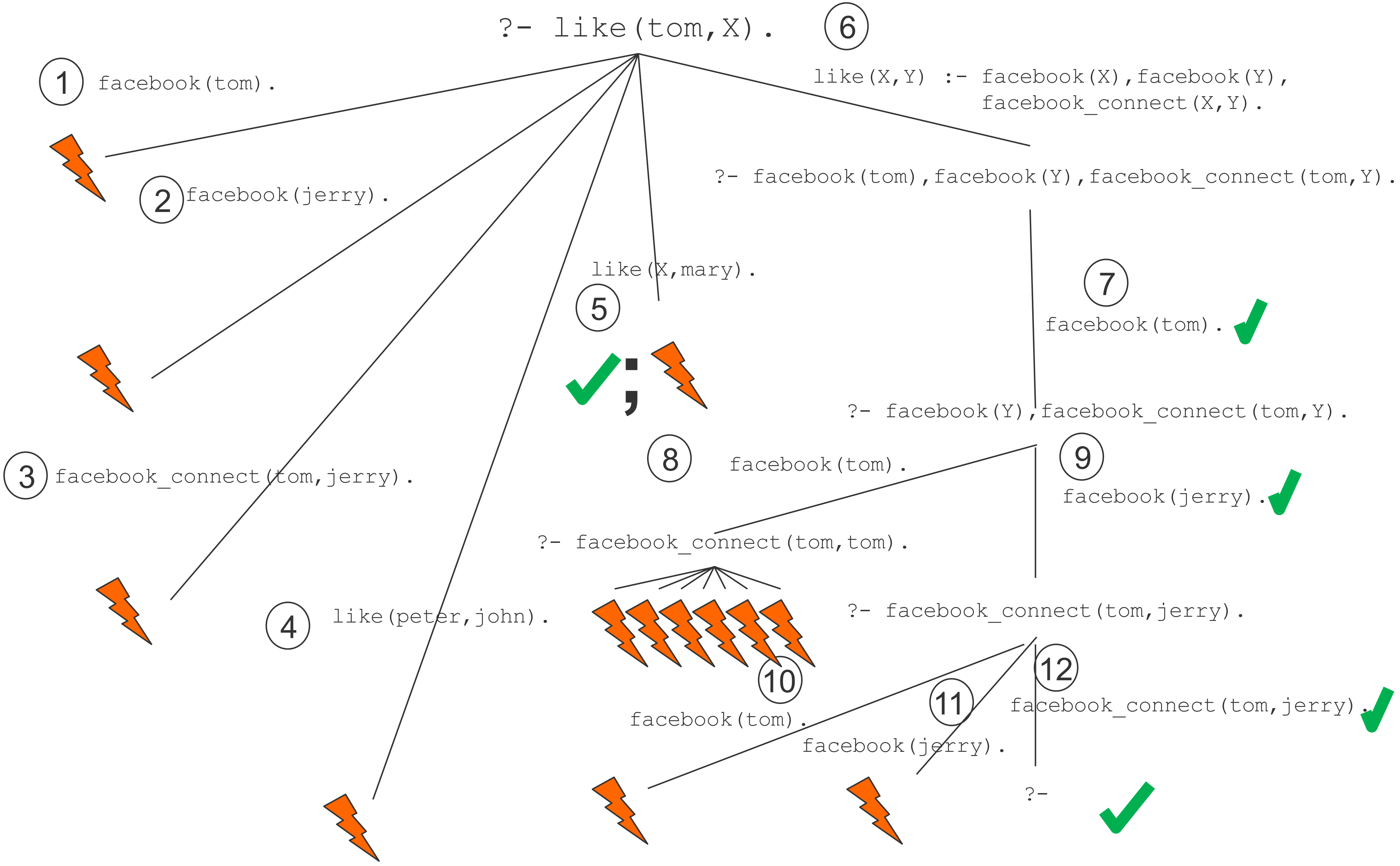
facebook(jerry). ✓

?- facebook_connect(tom,Y).

facebook_connect(tom,jerry). ✓

?- ✓

Error



```
?- like(tom,X).
```

1 `facebook(tom).`

2 `facebook(jerry).`

3 `facebook_connect(tom,jerry).`

4 `like(peter,john).`

5 `like(X,mary).`

✓ ;

6 `like(X,Y) :- facebook(X),facebook(Y),`
`facebook_connect(X,Y).`

```
?- facebook(tom),facebook(Y),facebook_connect(tom,Y).
```

7 `facebook(tom).` ✓

```
?- facebook(Y),facebook_connect(tom,Y).
```

8 `facebook(tom).`

9 `facebook(jerry).` ✓

```
?- facebook_connect(tom,tom).
```

```
?- facebook_connect(tom,jerry).
```

10 `facebook(tom).`

11 `facebook(jerry).`

12 `facebook_connect(tom,jerry).` ✓

```
?-
```

✓