Experiment No. 9

Aim: To implement family tree using recursion in Prolog.

Requirements: Compatible version of Prolog.

Theory:

Recursion

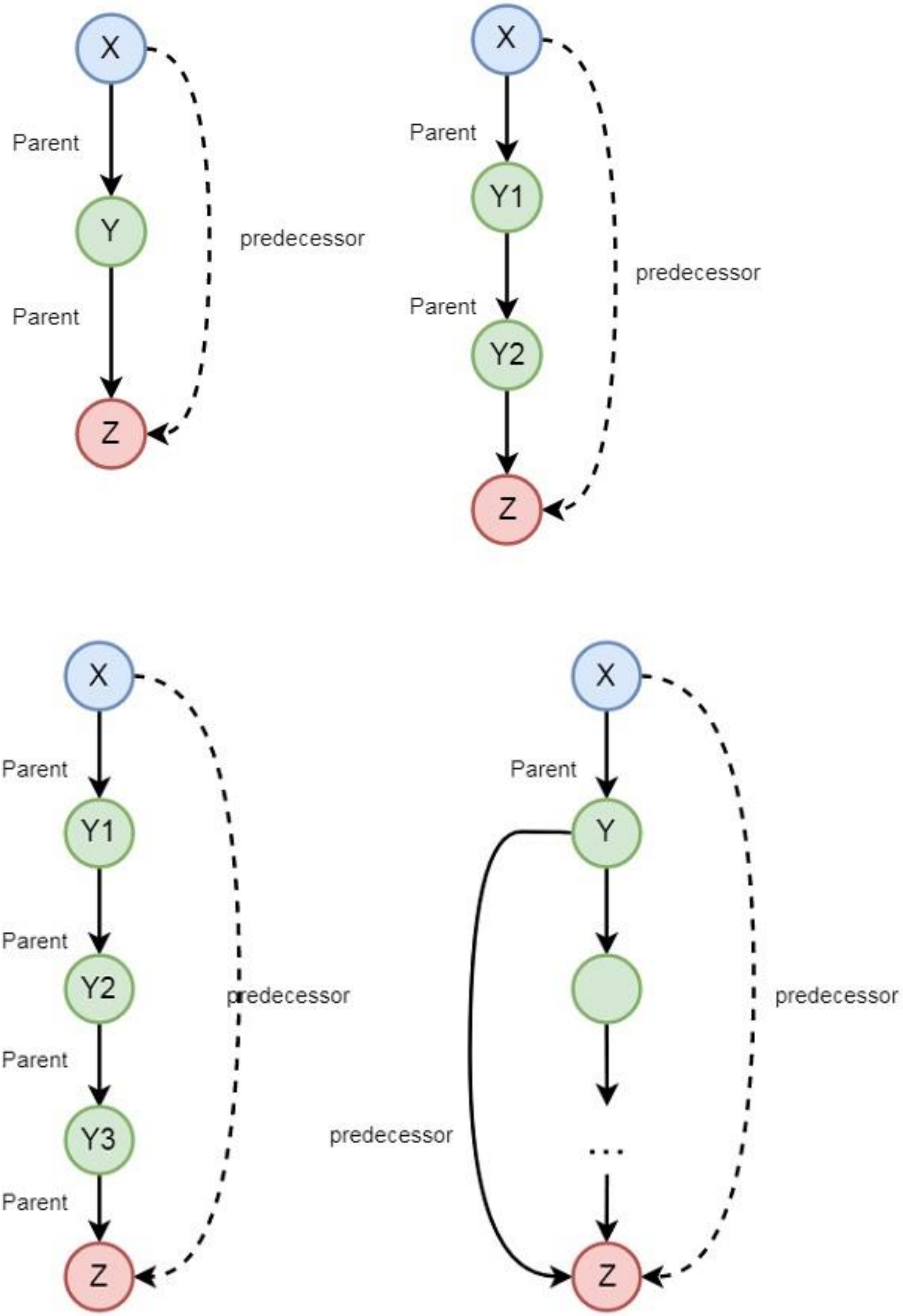
Recursion is a technique in which one predicate uses itself (may be with some other predicates) to find the truth value.

Let us understand this definition with the help of an example −

- is_digesting(X,Y) :- just_ate(X,Y).
- is_digesting(X,Y) :-just_ate(X,Z),is_digesting(Z,Y).

So this predicate is recursive in nature. Suppose we say that *just_ate(deer, grass),* it means *is_digesting(deer, grass)* is true. Now if we say *is_digesting(tiger, grass),* this will be true if *is_digesting(tiger, grass)* :- *just_ate(tiger, deer), is_digesting(deer, grass),* then the statement *is_digesting(tiger, grass)* is also true.

There may be some other examples also, so let us see one family example. So if we want to express the predecessor logic, that can be expressed using the following diagram –


X
Parent
Y
Parent
Z
predecessor
X
Parent
Y1
Parent
Y2
Z
predecessor
X
Parent
Y1
Parent
Y2
Parent
Y3
Parent
Z
predecessor
X
Parent
Y
...
Z
predecessor
predecessor

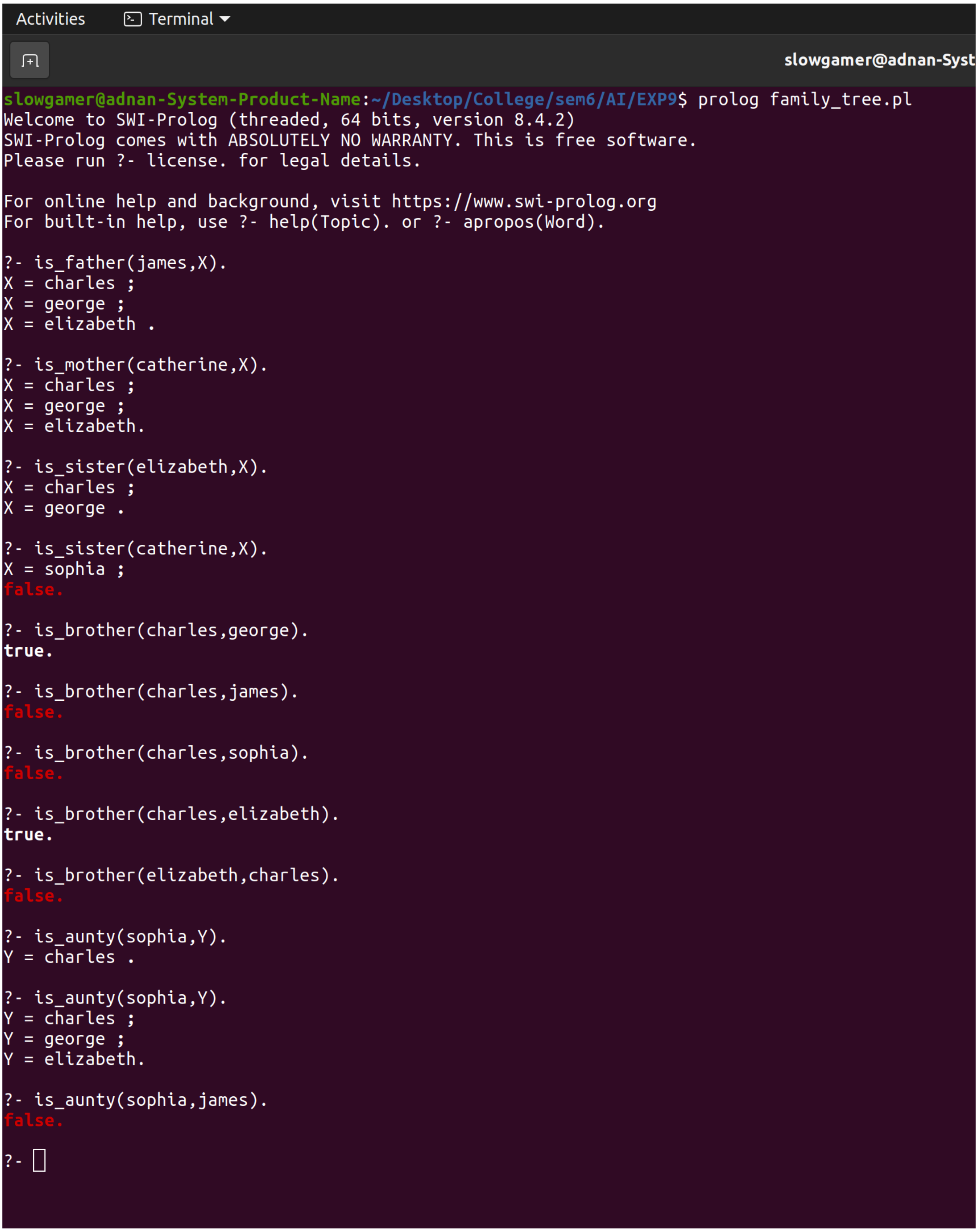
So we can understand the predecessor relationship is recursive. We can express this relationship using the following syntax −

- predecessor(X, Z) :- parent(X, Z).
- predecessor(X, Z) :- parent(X, Y),predecessor(Y, Z).

Code:

```
male(james).
male(charles).
male(george).
male(james).

female(catherine).
female(elizabeth).
female(sophia).

mother(catherine,charles).
mother(catherine,george).
mother(catherine,elizabeth).

sister(catherine,sophia).

partner(james,catherine).

is_father(X,Y):- male(X),partner(X,Z),mother(Z,Y).

is_mother(X,Y):- mother(X,Y).

is_sister(X,Y):- sister(X,Y); (female(X), mother(Z,X),mother(Z,Y)).

is_brother(X,Y):- male(X), mother(Z,X), mother(Z,Y).

is_aunty(X,Y):- female(X), sister(Z,X), mother(Z,Y).
```

Output:



```
slowgamer@adnan-System-Product-Name:~/Desktop/College/sem6/AI/EXP9$ prolog family_tree.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- is_father(james,X).
X = charles ;
X = george ;
X = elizabeth .

?- is_mother(catherine,X).
X = charles ;
X = george ;
X = elizabeth.

?- is_sister(elizabeth,X).
X = charles ;
X = george .

?- is_sister(catherine,X).
X = sophia ;
false.

?- is_brother(charles,george).
true.

?- is_brother(charles,james).
false.

?- is_brother(charles,sophia).
false.

?- is_brother(charles,elizabeth).
true.

?- is_brother(elizabeth,charles).
false.

?- is_aunty(sophia,Y).
Y = charles .

?- is_aunty(sophia,Y).
Y = charles ;
Y = george ;
Y = elizabeth.

?- is_aunty(sophia,james).
false.

?- 
```

Conclusion: We have successfully implemented family tree using recursion in Prolog.