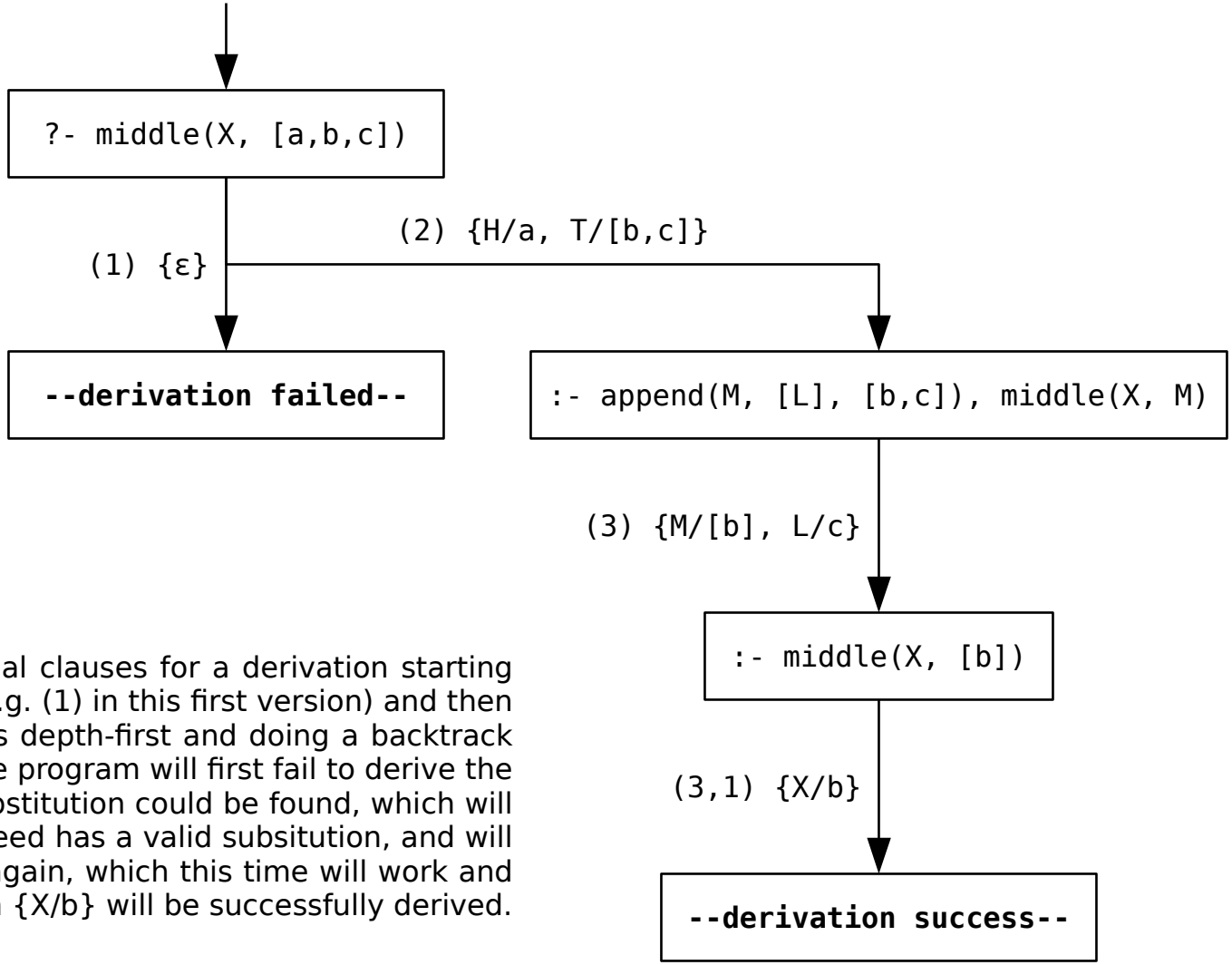
Program Version 1:

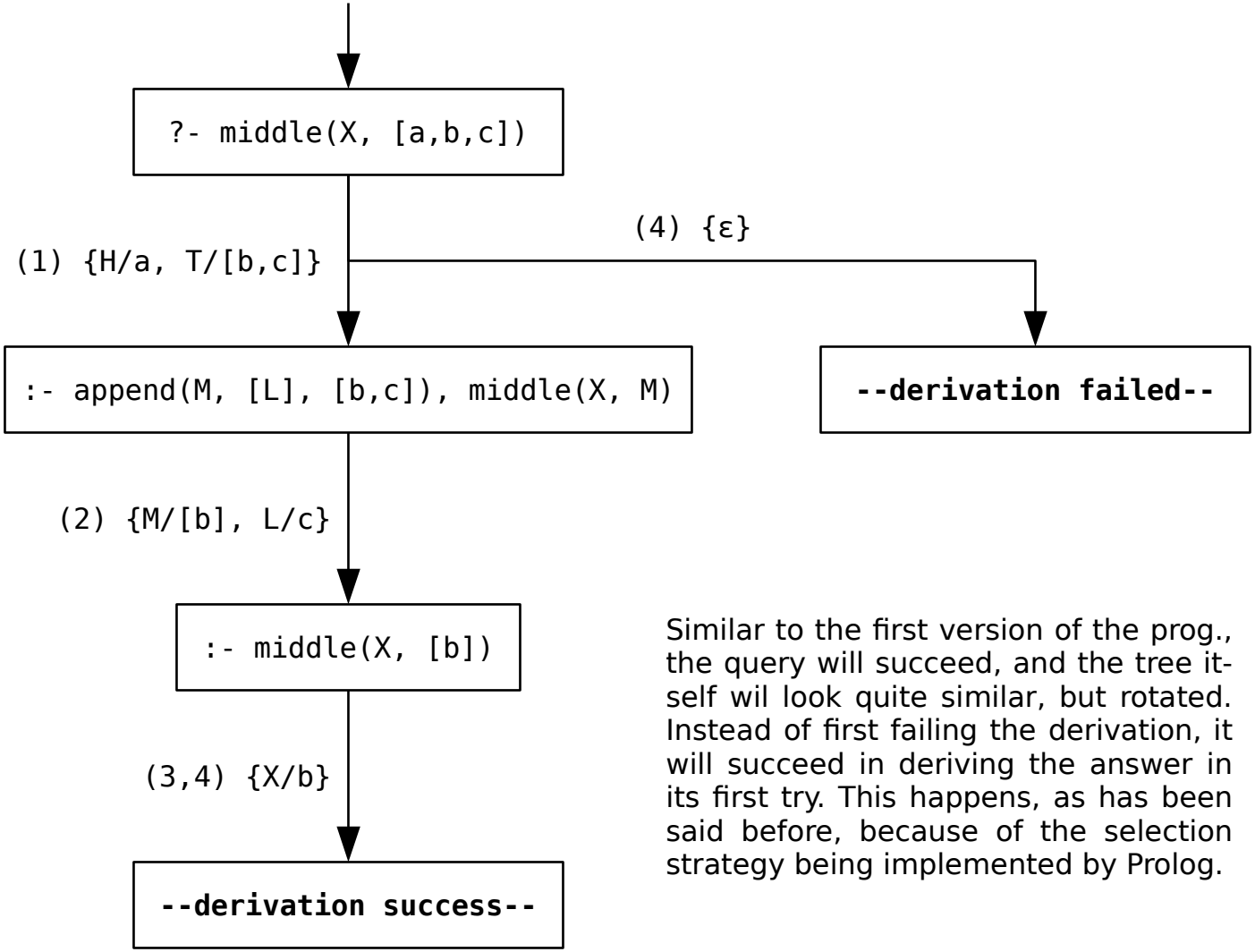
```
(1) middle(X, [X]).
(2) middle(X, [H|T]) :-
(3)   append(M, [L], T),
(4)   middle(X, M).
```



Since prolog selects potential clauses for a derivation starting from the top-most clause (e.g. (1) in this first version) and then traverses potential solutions depth-first and doing a backtrack if needed. Our version of the program will first fail to derive the (1) clause since no valid substitution could be found, which will then attempt (2), which indeed has a valid subsitution, and will continue to then apply (1) again, which this time will work and the solution/goal subsitution {X/b} will be successfully derived.

Program Version 2:

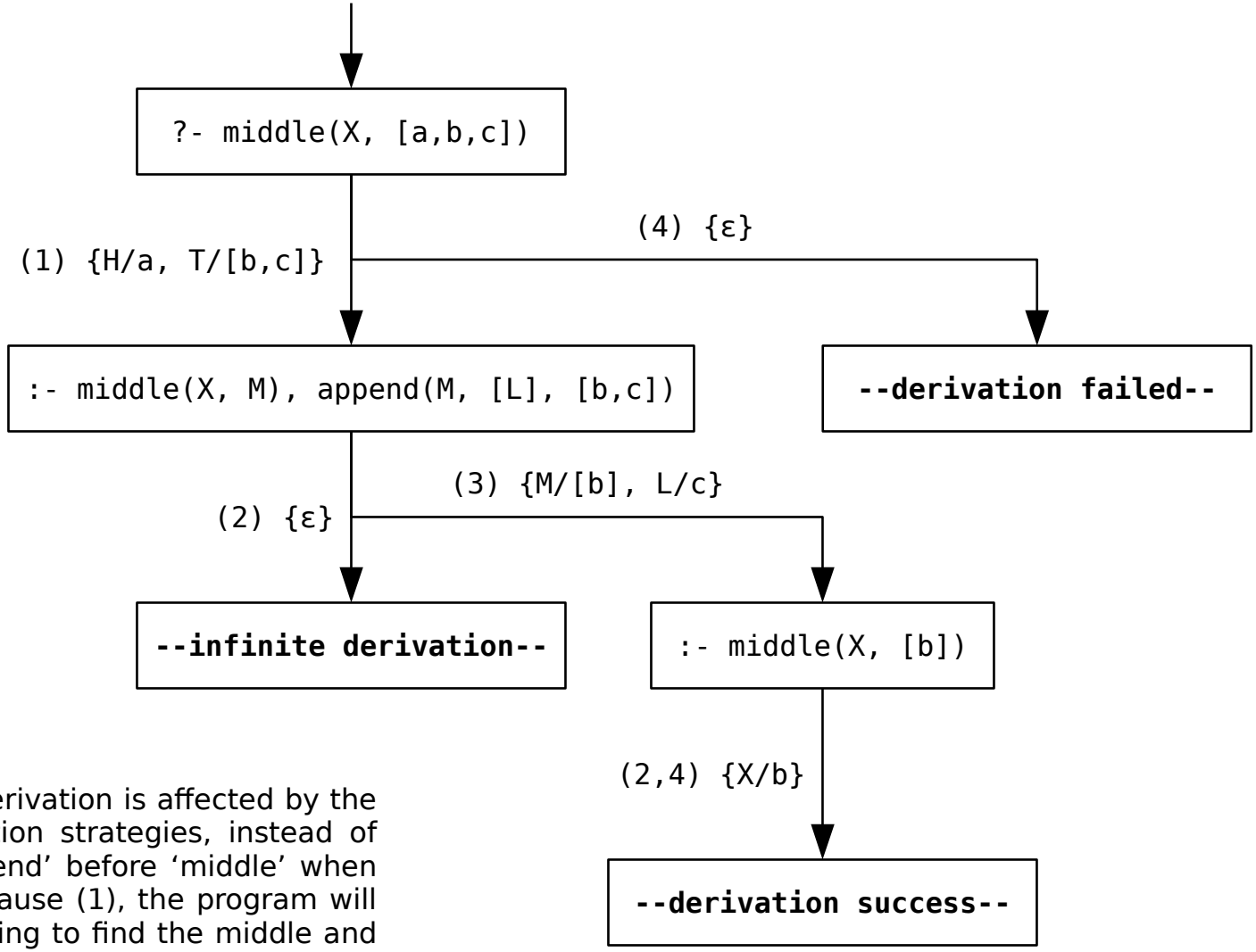
```
(1) middle(X, [H|T]) :-
(2)   append(M, [L], T),
(3)   middle(X, M).
(4) middle(X, [X]).
```



Similar to the first version of the prog., the query will succeed, and the tree itself wil look quite similar, but rotated. Instead of first failing the derivation, it will succeed in deriving the answer in its first try. This happens, as has been said before, because of the selection strategy being implemented by Prolog.

Program Version 3:

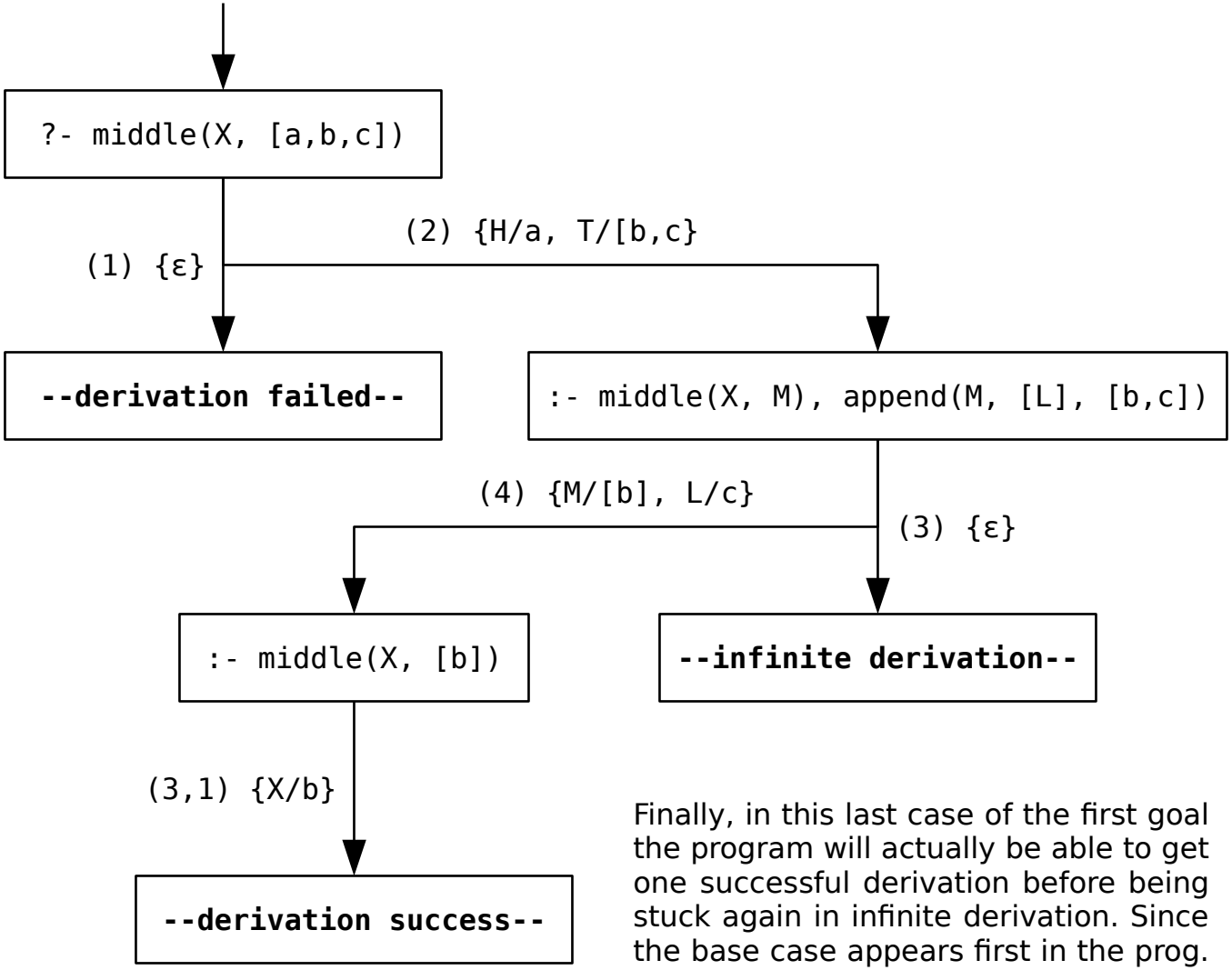
```
(1) middle(X, [H|T]) :-
(2)   middle(X, M),
(3)   append(M, [L], T).
(4) middle(X, [X]).
```



Again, the derivation is affected by the Prolog selection strategies, instead of picking ‘append’ before ‘middle’ when expanding clause (1), the program will find itself trying to find the middle and with no additional information gained. Look at program version 1 and 2, when the ‘append’ was before, allowing the derivation to actually resolve variables.

Program Version 4:

```
(1) middle(X, [X]).
(2) middle(X, [H|T]) :-
(3)   middle(X, M),
(4)   append(M, [L], T).
```



Finally, in this last case of the first goal the program will actually be able to get one successful derivation before being stuck again in infinite derivation. Since the base case appears first in the prog. it will be evaluated first, and ‘append’ will already have been substituted, the base case can then just plug values in.