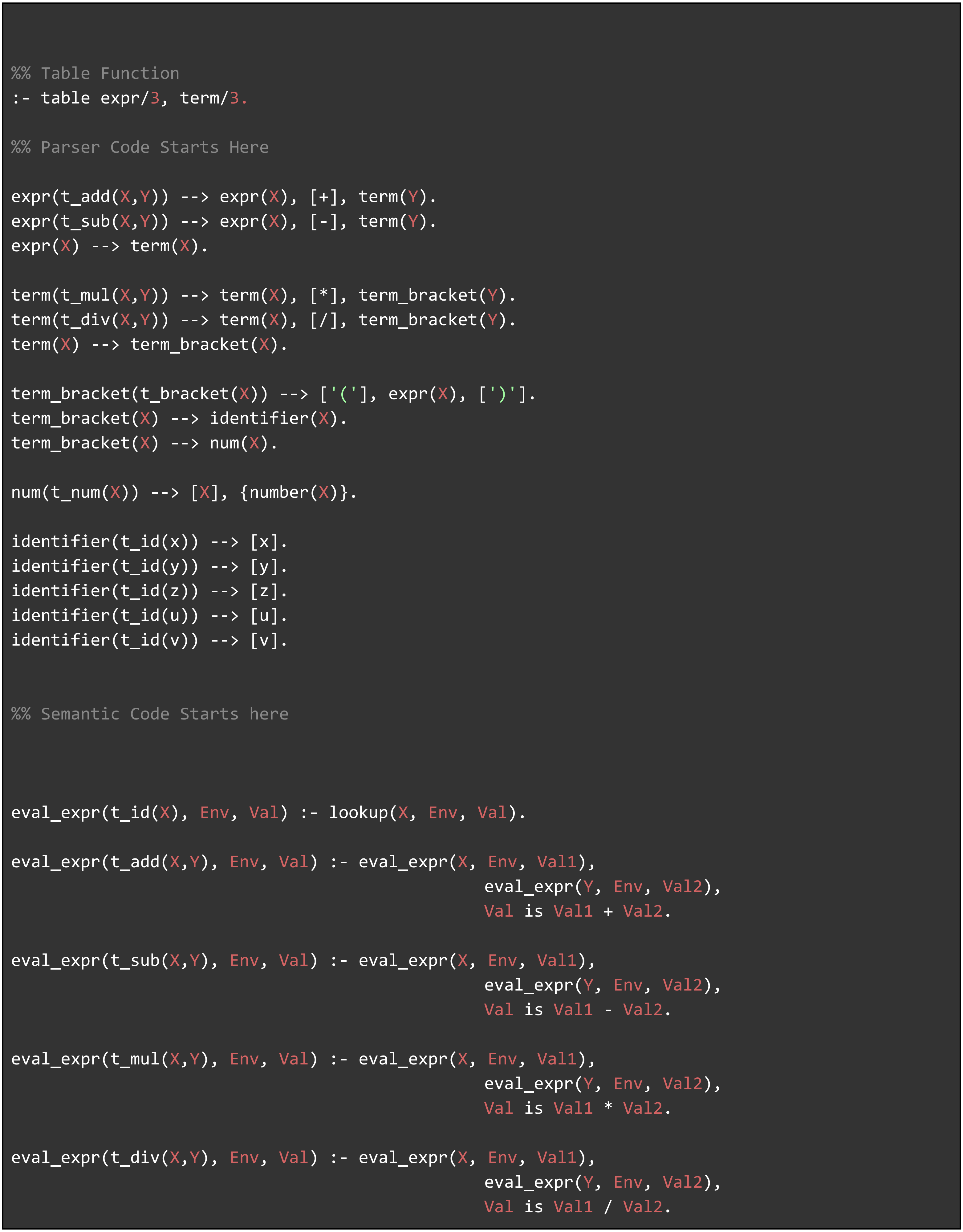

```
%% Table Function
:- table expr/3, term/3.

%% Parser Code Starts Here

expr(t_add(X,Y)) --> expr(X), [+], term(Y).
expr(t_sub(X,Y)) --> expr(X), [-], term(Y).
expr(X) --> term(X).

term(t_mul(X,Y)) --> term(X), [*], term_bracket(Y).
term(t_div(X,Y)) --> term(X), [/], term_bracket(Y).
term(X) --> term_bracket(X).

term_bracket(t_bracket(X)) --> ['('], expr(X), [')'].
term_bracket(X) --> identifier(X).
term_bracket(X) --> num(X).

num(t_num(X)) --> [X], {number(X)}.

identifier(t_id(x)) --> [x].
identifier(t_id(y)) --> [y].
identifier(t_id(z)) --> [z].
identifier(t_id(u)) --> [u].
identifier(t_id(v)) --> [v].


%% Semantic Code Starts here



eval_expr(t_id(X), Env, Val) :- lookup(X, Env, Val).

eval_expr(t_add(X,Y), Env, Val) :- eval_expr(X, Env, Val1),
                                                eval_expr(Y, Env, Val2),
                                                Val is Val1 + Val2.

eval_expr(t_sub(X,Y), Env, Val) :- eval_expr(X, Env, Val1),
                                                eval_expr(Y, Env, Val2),
                                                Val is Val1 - Val2.

eval_expr(t_mul(X,Y), Env, Val) :- eval_expr(X, Env, Val1),
                                                eval_expr(Y, Env, Val2),
                                                Val is Val1 * Val2.

eval_expr(t_div(X,Y), Env, Val) :- eval_expr(X, Env, Val1),
                                                eval_expr(Y, Env, Val2),
                                                Val is Val1 / Val2.
```

```
eval_expr(t_bracket(X), Env, Val) :- eval_expr(X, Env, Val).



eval_expr(t_num(X),_,X).


%% Lookup Function

lookup(Key,[(Key,Value)|_],Value).
lookup(Key,[_|Tail],Value) :- lookup(Key, Tail, Value).
```

```
?- expr(T,[2,+,3,*,x,/,y],[]),eval_expr(T,[(x,3),(y,4)],R).
T = t_add(t_num(2), t_div(t_mul(t_num(3), t_id(x)), t_id(y))),
R = 4.25 ;


?- expr(T,[2,+,3,*,x,/,y,+,1],[]),eval_expr(T,[(x,3),(y,4)],R).
T = t_add(t_add(t_num(2), t_div(t_mul(t_num(3), t_id(x)), t_id(y))), t_num(1)),
R = 5.25 ;
```