Declaratieve Talen
Prolog 2

1 Expressive

Implement an `eval/3` predicate that evaluates arithmetic expressions given a list of assignments to the variables that appear in the expression. The following example shows the evaluation of $x + (2 * y)$, where $x = 2$ and $y = 3$.

```
?- eval(plus(var(x),times(int(2),var(y))),[pair(x,2),pair(y,3)],Value).
Value = 8.
```

An arithmetic expression is one of the following Prolog terms, where E, E_1, and E_2 are arithmetic expressions themselves:

- `int(`I`)` where I is an integer: `int(4)` denotes the value 4;
- `var(`A`)` where A is an atom: `var(x)` denotes the variable x;
- `plus(`E_1`,`E_2`)`;
- `times(`E_1`,`E_2`)`;
- `pow(`E_1`,`E_2`)`;
- `min(`E`)` to represent $-E$.

The built-in operator for exponentiation is`(**)`. For example, `8 is 2**3`.

```
?- eval(min(int(3)),[],Value).
Value = -3.

?- eval(plus(int(2),var(x)),[pair(x,3)],Value).
Value = 5.

?- eval(plus(pow(var(x),var(y)),min(plus(times(int(3),var(z)),min(var(y))))),
[pair(x,2),pair(y,3),pair(z,5)],Value).
Value = -4.
```