Ruler Framework

Grammar

```
e ::= x, 0, e + e, e * e, …
```

Interpreter

```
match e {
  | const   => const
  | var (v) => lookup (v)
  | e1 + e2 => eval (e1) + eval(e2)
  | e1 * e2 => eval (e1) * eval(e2)
  …
}
```

Validator

```
SMT / model check / fuzz
```

Term Enumeration Modulo Equivalence → **Candidate Rule Generation** → **Rule Selection**

Rewrites

```
        x + 0 = x
        x * 1 = x
        x - 0 = x
        x / 1 = x
      x + y = y + x
x + (y + z) = (x + y) + z
x * (y * z) = (x * y) * z
```