Typing rules → JustAdd code

Try out on [Object language]:

https://github.com/lu-cs-sde/metadl-jastadd

$$e ::= \text{true}$$
$$| \quad \langle e \rangle \text{ or } \langle e \rangle$$
$$| \quad \text{false}$$
$$| \quad [\langle e \rangle]$$
$$| \quad 0$$

$$\overline{0 : \text{Int}}$$

// or something else
so we can get a
type error

$$\overline{\text{true} : \text{Bool}} \qquad \overline{\text{false} : \text{Bool}}$$

$$\frac{e_1 : \text{Bool} \quad e_2 : \text{Bool}}{e_1 \text{ or } e_2 : \text{Bool}}$$

Later:

$$\frac{e : \tau}{[e] : \text{List}[\tau]}$$

(or simpler rule for other construct)

We have:
- parser for typing rules → AST

We want
- access to [object language] grammar
- error handling

org. jastadd. JastAdd.compile
↳ read Ast files

Grammar object, cf.:
MetaGrammar. jrag
declsByName