Final Review: Interpreters

Space

Characters

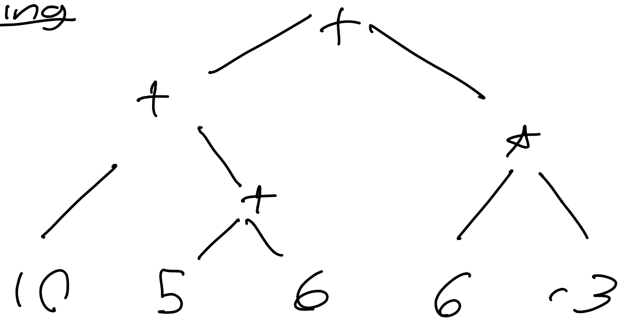
1	0	.	+	.	(	5	+	6	)	+	6	*	-	3

Lexing

Tokens

10	+	(	5	+	6	)	+	6	*	-3

Parsing



Abstract syntax tree

Evaluation

```
type bop = Add | Mult

type expr =
   | Int of int
   | Binop of bop * expr * expr
```

```
↳ let rec eval (e: expr) : int =
    match e with
     | Int n → n
     | Binop (op, e1, e2) →
        match op with
         | Add → (eval e1) + (eval e2)
         | Mult → (eval e1) * (eval e2)
```

Tools

Eval: by hand

Parser
 Menhir: parser generator
  .mly ⟶ .ml

look at example interpreter?

Lexer
 ocamllex
  .mll ⟶ .ml

---

1 0 . + (5 + 6) . + . 5 * − 3