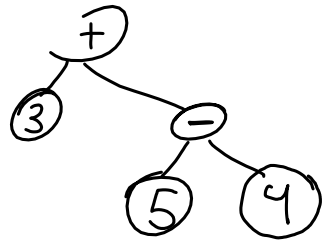


MAC316 24/3/8

(94)

in 1 + 2
por 1 2 +
pu (+ 1 2)

3 + (5 - 4)
3 5 4 - +
(+ 3 (- 5 4))



AE
(add (num3) sub (num5)
(num4)))

1) Ler expressões pré-fixas com +, - e números

2) Converter p/ uma lista "interna"

para "(+ 1 3))

↳ [add {num 1} {num 3}]

↳ (calc '(add (num 1) (num 3)))

↳ 4

MAC316

(95)

(define-type ID (VAR-ID (FIELD (CONTRACT)...)...))

↑ nome do tipo (AE)	↑ variante add sub num ⋮	ident	predicador e valor respe
---------------------------------	---	-------	-----------------------------------

- Guia o tipo ID
- Guia o predicador ID?
- Guia construtores para cada variante VAR-ID (number? S)

```
(define-type AE
  (add (lhresq. number?)
        (rhr number?))
  (sub (lhr AE?)
        (rhr AE?))
  (num (n number?)))
```

parser ← (f 1 2) → reserp
 ↳ (add (num 1) (num 2))

MITC 316

(96)

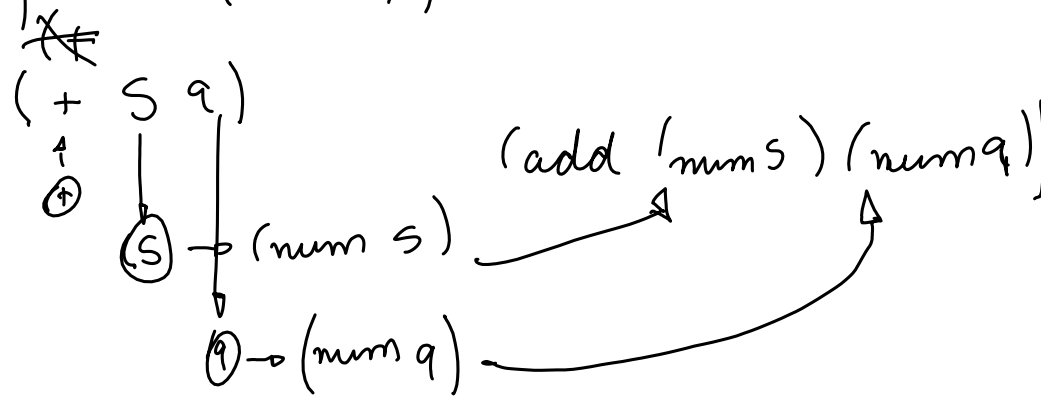
```

(define (parse rexp)
  (cond
    [(number? rexp) (num rexp)]
    [(constr? rexp)
     (case (car rexp)
       [(+) (add (parse (second rexp))
                 (parse (third rexp)))]
       [(-) (sub (parse (second rexp))
                 (parse (third rexp)))]
       )])
  )

```

(parse 2)
 (parse '(+ 5 9))

rexp 2 randa (num 2)



MAC 316

97

(type-case TIPO expr
(VAR-ID (FIELD...)) RESULTADO)

: num n

) add lhr rhr

sub lhr rhr

define (calc uma-ae)

Example

(type-case AE uma-ae)

[num (n) ⁿ]

[add (a1 a2) (+ (calc a1) (calc a2))]

[sub (a1 a2) (- (calc a1) (calc a2))]

remãntica

desafio: retornar expressão infixa. bônus até sexta!

26/3/8

{ with { ^{id} símbolo expr } expr }

{ _i id símbolo }

Substituição: troca toda ocorrência de i em e' por v