

- Eliminarea recursivității la stângă:

$$X \rightarrow x_1 | \dots | x_n | \beta_1 | \dots | \beta_m$$

$$X \rightarrow \beta_1 x' | \dots | \beta_m x'$$

$$x' \rightarrow \alpha_1 x' | \dots | \alpha_n x' | \epsilon$$

- Recursive descent (slide 166, ex: 167)

- LL(1) (slide 197, ex: 199)

- Multiple First (slide 239)

- Multiple Follow (slide 245)

- LL(1)  $\rightarrow$  construire tabel: tabel  $[X, t] = 1$  dacă  $\cancel{\text{First}(X)} \rightarrow t \in \text{First}(L)$  sau  $\epsilon \in \text{First}(L)$  și  $t \in \text{Follow}(X)$  (slide 265)

- Conflicts LR(0) (slide 350)

- SLR(1) (slide 353)  $\rightarrow$  Facem reduce dacă terminalul este în  $\text{Follow}(X)$ , altfel facem shift.

- Reguli de tipare:

- int, add, string, Bool (470-473)

- merge (474)

- not și while (475)

- variabile (477-478)

- let (484)

- let cu inițializare (488)

- atribuire (489)

- ~~atribute~~ attribute inițializate (490)

- derivare (494)

- case (495)

- blocuri (496)

- apeluri de metodă (500)

- apeluri statice de metodă (501)

- self (516)

- apeluri de metodă cu ~~self~~ SELF\_TYPE (520)



- Instrucțiuni MIPS (557)
- ~~for~~ generare cod constantă (560)
- generare cod pt. adunare (561)
- generare cod pt. decizii (563)
- generare de cod pt. apelurile de funcție (577)
- generare de cod pt. definiții de funcție (578)
- ——— " ——— parametri (579)
- locații temporare (581-584)
- Reprezentarea obiectelor în memorie (590)

Inta; a.k.a  
Objecto; a.k.a

da

