

Projet de programmation fonctionnelle et de traduction des langages

MDAA Saad — El Bouzekraoui Younes

Département Sciences du Numérique - Deuxième année
2020-2021

Contents

1	Introduction	3
2	Pointeurs	3
2.1	Modifications de la grammaire	3
2.2	Jugement de typage	3
3	Surcharge des fonctions	3
4	Types énumérés	3
4.1	Modifications de la grammaire	3
4.2	Jugement de typage	4
5	Switch/case	4
5.1	Modifications de la grammaire	4
5.2	Jugement de typage	4

1 Introduction

Le but de ce projet est de ajouter les extensions du langage Rat réalisé en TP de traduction des langages en Ocaml afin de traiter les nouvelles constructions

- pointeurs
- surcharge des fonctions
- types énumérés
- switch/case

2 Pointeurs

2.1 Modifications de la grammaire

On ajoute les règles suivantes :

- $\text{TYPE} \rightarrow \text{TYPE} *$
- $E \rightarrow A$
- $A \rightarrow * A$
- $A \rightarrow \text{id}$
- $E \rightarrow \text{null}$
- $E \rightarrow \text{new TYPE}$
- $E \rightarrow \& \text{id}$
- $I \rightarrow A = E$

Afin de traiter l'extension des pointeurs on doit ajouter un nouveau type dans **type.ml** : Pointeur of typ, un pointeur sur null a un type de Pointeur(Undefined)

2.2 Jugement de typage

$$\frac{\sigma \vdash a : \text{Pointeur}(\tau)}{\sigma \vdash *a : \tau} \quad (1)$$

$$\frac{\sigma \vdash a : \tau}{\sigma \vdash \&a : \text{Pointeur}(\tau)} \quad (2)$$

$$\sigma \vdash \text{null} : \text{Pointeur}(\text{Undefined}) \quad (3)$$

$$\frac{\sigma \vdash a : \tau}{\sigma \vdash \text{new } a : \text{Pointeur}(\tau)} \quad (4)$$

3 Surcharge des fonctions

4 Types énumérés

4.1 Modifications de la grammaire

On ajoute les règles suivantes :

- $\text{Main} \rightarrow \text{ENUMS PROG}$
- $\text{ENUMS} \rightarrow \Lambda$
- $\text{ENUMS} \rightarrow \text{ENUM ENUMS}$
- $\text{ENUM} \rightarrow \text{enum tid IDS}$

- $\text{IDS} \rightarrow \text{tid}$
- $\text{IDS} \rightarrow \text{tid}, \text{IDS}$
- $\text{TYPE} \rightarrow \text{tid}$
- $\text{E} \rightarrow \text{tid}$

4.2 Jugement de typage

$$\frac{\sigma \vdash E_1 : \text{Enum}(n), \sigma \vdash E_2 : \text{Enum}(n)}{\sigma \vdash (E_1 = E_2) : \text{bool}} \quad (5)$$

5 Switch/case

5.1 Modifications de la grammaire

On ajoute les règles suivantes :

- $\text{I} \rightarrow \text{switch (E) LCase}$
- $\text{LCase} \rightarrow \text{Case LCase}$
- $\text{LCase} \rightarrow \Lambda$
- $\text{Case} \rightarrow \text{case tid} : \text{IS B}$
- $\text{Case} \rightarrow \text{case entier} : \text{IS B}$
- $\text{Case} \rightarrow \text{case true} : \text{IS B}$
- $\text{Case} \rightarrow \text{case false} : \text{IS B}$
- $\text{Case} \rightarrow \text{default} : \text{IS B}$
- $\text{B} \rightarrow \Lambda$
- $\text{B} \rightarrow \text{break};$

5.2 Jugement de typage

$$\frac{\sigma \vdash E : \tau, \sigma \vdash E_1 : \tau, \sigma \vdash \text{Bloc}_1 : \text{void}, \sigma \vdash \text{default} : \text{Undefined}, \sigma \vdash \text{Bloc}_3 : \text{void}}{\sigma \vdash \text{switch}(E) \text{case } E_1 \text{Bloc}_1 \dots \text{default } \text{Bloc}_3 : \text{void}, []} \quad (6)$$