

Ponts analyse syntaxique-sémantique

Enogad Le Biavant-Frederic

Alain René Lesage MP2I

2024

L'idée

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

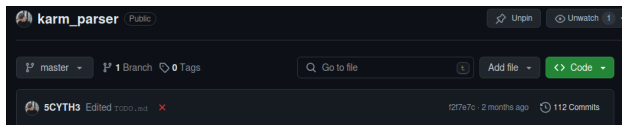
Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

Karm, 2022



L'idée

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

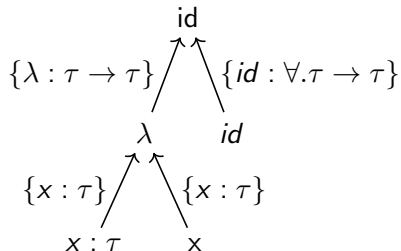
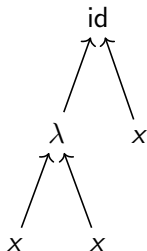
Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

```
1 ( let id = \x.x in id )
```



Grammaire

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

$$\begin{array}{lcl} \text{Expr} & e & ::= \\ & & x \\ & & | \quad f \ e \\ & & | \quad \text{let } x = e_1 \text{ in } e_2 \\ & & | \quad \backslash x.e \\ & & | \quad '(\ e \)' \end{array}$$

Hindley-Milner

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

$$\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma} \text{ var}$$

$$\frac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x. e : \tau \rightarrow \tau'} \text{ abs}$$

$$\frac{\Gamma \vdash f : \tau \rightarrow \tau' \quad \Gamma \vdash e : \tau}{\Gamma \vdash f \ e : \tau'} \text{ app}$$

Hindley-Milner

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

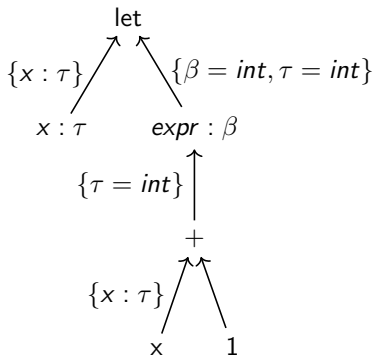
Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

- 1 Assignment de variables de types aux expressions
- 2 Génération de contraintes
- 3 Substitutions
- 4 Unification
- 5 Instantiation, généralisation



Ma démonstration

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

```
1 let parse_let (program: program) (env: Types.env): traversal =
2   match program with
3   | Let :: Id i :: Assign :: t ->
4     let open Types in
5     let { expr = body; t = t1; subst = subst1; rest; } = parse_expr t
6       env in
7     let t' = generalize env (apply_subst subst1 t1) in
8     let env' = extend_env env i t' in
9     let { expr = in_expr; t = t2; subst = subst2; rest = rest'; } =
10      parse_ins rest env' in
11      {
12        expr = Let (i, body, in_expr);
13        t = t2;
14        subst = subst2 ** subst1;
15        rest = rest';
16      }
17   | _ -> failwith "Expected 'Let'."
18
```

Résultats

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

Parsing
Typesystem

**Ma
démonstration**

Automates
d'Arbres

Et après ?

Bah y'a rien mdr

Automates d'Arbres

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions

Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

Definition (Alphabet gradué)

$$\Sigma \equiv (\Sigma, ar)$$

Definition (Automates d'arbre)

$$\mathcal{A} = (Q, \Sigma, F, \Delta)$$

Et après ?

Ponts analyse
syntaxique-
sémantique

Enogad Le
Biavant-
Frederic

Présentation
générale

Définitions
Parsing
Typesystem

Ma
démonstration

Automates
d'Arbres

Et après ?

- Benchmarks plus précis
- Calcul formel de complexité / correction et terminaison
- Constraint-Based Set Analysis
- Raffiner l'utilisation d'automates d'arbres