

Ocaml

Autores : Felipe Assad , Jorge Chagas, Thiago Augusto

Data : 14/06/2019

Universidade Federal Fluminense

Objetivos desta apresentação

- Apresentar as modificações no Automaton.ml e automatonType.ml

O que foi feito

- Redefinimos no automatonType.ml os tipos indutivos

```
type valueStackOptions =  
  | Int of int  
  | Str of string  
  | Bool of bool  
  | LoopValue of command  
  | CondValue of command  
  | Bind of loc  
  | Locations of int list  
  | Env of (string, bindable) Hashtbl.t  
  
and storable =  
  | Integer of int  
  | Boolean of bool  
  | Pointer of loc  
  
and bindable =  
  | Loc of loc  
  | IntConst of int  
  | BoolConst of bool  
  
and loc =  
  | Location of int;;
```

O que foi feito

- Fizemos as atualizações necessárias

```
| Loc(x) -> (  
  (Stack.push (Bind(Loc(x))) valueStack );  
  (Stack.push (Bind(x)) valueStack );  
  ,
```

O que foi feito

- Ajuste do Bind | `Bind(y) -> (`

```
let possibleEnv = (Stack.top valueStack) in
match possibleEnv with
| Env(x) -> (
  let env = (Stack.pop valueStack) in
  match env with
  | Env(e) -> (
    let newEnv = (Hashtbl.copy e) in
    (Hashtbl.add newEnv st (Loc(y)) );
    (Stack.push (Env(newEnv)) valueStack );
  );
  | _ -> raise (AutomatonException "Error on #BIND" );
);
| _ -> (
  let newEnv = (Hashtbl.create 3) in
  (Hashtbl.add newEnv st (Loc(y)));
  (Stack.push (Env(newEnv)) valueStack );
);
);
```

O que foi feito

- Podemos agora inserir comentários no código

```
rule token = parse
  [ ' ' '\t' '\r' '\n' ] { token lexbuf }      (* skip blanks *)
|  "#"([^\n']+ ) { token lexbuf }      (* skip comment lines *)
|  (['-']? ['0'-'9']+) as lxm { NUMBER( int_of_string lxm) }
```

O que foi feito

- Makefile atualizado

```
+  
+ p1:  
+     git pull origin p1  
+     git checkout p1  
+     $(MAKE) run  
+  
+ p2:  
+     git pull origin p2  
+     git checkout p2  
+     $(MAKE) run  
+
```

O que não foi feito e por quê?

- Elaborar testes
- Tratar todos os pattern matches do autômato

Por quê?

- Porque antes fizemos os ajustes apresentados aqui

- É obrigado fazer as operações aritméticas com Dref e Valref?

Avaliação da evolução do trabalho

- Implementar um parser para a linguagem Imp-1 estendendo Imp-0 com declarações de variáveis e constantes. (OK)
- Implementar IR-mark1: (i) Interpreting Automata com ambientes (OK), (ii) declarações de variáveis e constantes. (1/2 OK)
- Implementar um compilador de Imp-1 para IR-mark1. (4/5 OK)