



Why GitHub?

Enterprise

Explore

Marketplace

Pricing

Search



Sign in

Sign up

sh-soltanpour / compilerPhase2

Watch

0

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Security

Insights

1-4 done

Browse files

Loading branch information



tinayz committed Dec 4, 2017

1 parent 16233d1

commit a3d36416b3a6dd75187ba4ce8d73737013d6dde6

Showing 4 changed files with 101 additions and 36 deletions.

Unified

Split

67 Atalk.g4



@@ -14,58 +14,52 @@ grammar Atalk;

```
14 14      print("Stack offset: " + SymbolTable.top.getOffset(Register.SP));
15 15      SymbolTable.pop();
16 16  }
```

```
17 - void putLocalVar(String name, Type type) throws ItemAlreadyExistsException{
18 -     SymbolTable.top.put(
19 -         new SymbolTableLocalVariableItem(
20 -             new Variable(name, type),
21 -             SymbolTable.top.getOffset(Register.SP)
22 -         )
23 -     );
24 - }
25 - void putGlobalVar(String name, Type type) throws ItemAlreadyExistsException{
26 -     SymbolTable.top.put(
27 -         new SymbolTableGlobalVariableItem(
28 -             new Variable(name, type),
29 -             SymbolTable.top.getOffset(Register.GP)
30 -         )
31 -     );
32 - }
```

```
33 17  }
34 18  program:
35 19      {beginScope();}(actor | NL)*{endScope();}
36 20  ;
37 21
38 22  actor:
39 23      'actor' name=ID '<' mailboxSize=CONST_NUM '>' NL
```

```

40 - {
41 -     try{
42 -         Tools.putActor($name.text,Integer.parseInt($mailboxSize.text));
43 -     }
44 -     catch(ItemAlreadyExistsException e) {
45 -         print("Exists");
46 -     }
47 -     finally{beginScope(); }
24 + {
25 +     boolean error;
26 +     error = Tools.putActor($name.text, Integer.parseInt($mailboxSize.text));
27 +     if (error){
28 +         print("actor darim:");
48 29     }
30 +     beginScope();
31 + }
49 32     (state | receiver | NL)*
50 33     'end' (NL | EOF)
51 34     {endScope();}
52 35     ;
53 36     state:
54 37         type name = ID (',' ID)* NL
55 38         {
56 -         try{
57 -             putGlobalVar($name.text,$type.return_type);
39 +         boolean error = Tools.putGlobalVar($name.text,$type.return_type);
40 +         if (error){
41 +             print("global darim:P");
58 42         }
59 -         catch(ItemAlreadyExistsException e) {
60 -             print("Exists");
61 -         }
43 +
62 44         }
63 45         ;
64 46
65 47         receiver:
66 -         'receiver' ID '(' (type ID (',' type ID)*)? ')' NL
48 +         {
49 +             ArrayList<Type> arguments = new ArrayList<Type>();
50 +         }
51 +         'receiver' name = ID '(' ( type ID {arguments.add($type.return_type);}(',' type ID{arguments.add($type.return_type)
52 +         {boolean error;
53 +         error = Tools.putReceiver($name.text,arguments);
54 +         if (error){
55 +             print("receiver ham darim:D");
56 +         }
57 +         beginScope();
58 +         }
67 59         statements
68 60         'end' NL
61 +         {endScope();}
62 +
69 63         ;
70 64
71 65         type returns [Type return_type]:
72 @@ -96,9 +90,20 @@ statement:
96 90         ;
97 91
98 92         stm_vardef:
99 -         type ID ('=' expr)? (',' ID ('=' expr)?)* NL
93 +         {
94 +             ArrayList<String> names = new ArrayList<String>();
95 +         }
96 +         type name = ID{names.add($name.text);} ('=' expr)? (',' name2 = ID{names.add($name2.text);} ('=' expr)?)* NL
97 +         {for(int i = 0 ; i < names.size() ; i++){
98 +             boolean error = Tools.putLocalVar(names.get(i), $type.return_type );

```

```
98 + boolean error = tools.putLocalVar(names.get(1), type.return_type );
99 + if(error){
100 +     print("Localam darim:|");
101 + }
102 + }
103 + }
100 104 ;
101 105
106 +
102 107 stm_tell:
103 108 (ID | 'sender' | 'self') '<<' ID '(' (expr (',' expr)*? ') ' NL
104 109 ;
```

3 Receiver.java

```
... @@ -1,4 +1,5 @@
1 - public class Actor {
1 + import java.util.*;
2 + public class Receiver {
2 3     public Receiver(String name, ArrayList<Type> argumentTypes) {
3 4         this.name = name;
4 5         this.argumentTypes = argumentTypes;
```

10 SymbolTableReceiverItem.java

```
... @@ -1,3 +1,5 @@
1 + import java.util.ArrayList;
2 +
1 3     public class SymbolTableReceiverItem extends SymbolTableItem {
2 4         public SymbolTableReceiverItem(Receiver receiver, int offset) {
3 5             this.receiver = receiver;
@@ -6,7 +8,13 @@ public SymbolTableReceiverItem(Receiver receiver, int offset) {
6 8
7 9     @Override
8 10    public String getKey() {
9 -     return receiver.getName();
11 +     ArrayList <Type> argumentTypes = receiver.getArgumentTypes();
12 +     String name = receiver.getName();
13 +     String key = name+'#';
14 +     for(int i = 0 ; i < argumentTypes.size(); i++){
15 +         key += argumentTypes.get(i).toString();
16 +     }
17 +     return key;
10 18 }
11 19
12 20     public int getOffset() {
```

57 Tools.java

```
...    ...    @@ -1,7 +1,58 @@
1      + import java.util.*;
2      +
1      3      public class Tools {
2      4
3      3      - static public void putActor(String name, int mailboxSize) throws ItemAlreadyExistsException {
4      4      - SymbolTable.top.put(new SymbolTableActorItem(new Actor(name, mailboxSize), SymbolTable.top.getOffset(Register.GP)));
5      5      + static public boolean putActor(String name, int mailboxSize ) {
6      6      + boolean error = false;
7      7      + try {
8      8      + SymbolTable.top.put(new SymbolTableActorItem(new Actor(name, mailboxSize), SymbolTable.top.getOffset(Register.GP)));
9      9      + } catch (ItemAlreadyExistsException e) {
10     10     + error = true;
11     11     + Tools.putActor(name+"_temp_", mailboxSize);
12     12     + }
13     13     + return error;
14     14     + }
15     15     +
16     16     + static public boolean putReceiver(String name, ArrayList <Type> arguments) {
17     17     + boolean error = false;
18     18     + try {
19     19     + SymbolTable.top.put(new SymbolTableReceiverItem(new Receiver(name, arguments), SymbolTable.top.getOffset(Register.GP)));
20     20     + catch (ItemAlreadyExistsException e) {
21     21     + error = true;
22     22     + Tools.putReceiver(name+"_temp_", arguments);
23     23     + }
24     24     + return error;
25     25     + }
26     26     + static public boolean putLocalVar(String name, Type type){
27     27     + boolean error = false;
28     28     + try{
29     29     + SymbolTable.top.put(
30     30     + new SymbolTableLocalVariableItem(
31     31     + new Variable(name, type),
32     32     + SymbolTable.top.getOffset(Register.SP)
33     33     + )
34     34     + );
35     35     + }
36     36     + catch (ItemAlreadyExistsException e) {
37     37     + error = true;
38     38     + Tools.putLocalVar(name+"_temp_", type);
39     39     + }
40     40     + return error;
41     41     + }
42     42     + static public boolean putGlobalVar(String name, Type type){
43     43     + boolean error = false;
44     44     + try{
45     45     + SymbolTable.top.put(
46     46     + new SymbolTableGlobalVariableItem(
47     47     + new Variable(name, type),
48     48     + SymbolTable.top.getOffset(Register.GP)
49     49     + )
50     50     + );
51     51     + }
52     52     + catch (ItemAlreadyExistsException e) {
53     53     + error = true;
54     54     + Tools.putGlobalVar(name+"_temp_", type);
55     55     + }
56     56     + return error;
57     57     }
58     58     }
59     59     }
```

0 comments on commit `a3d3641`

Please [sign in](#) to comment.

© 2019 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Help](#)

[Contact GitHub](#)

[Pricing](#)

[API](#)

[Training](#)

[Blog](#)

[About](#)