

Team Project

SASL COMPILER

Kevin Jeuter

Intro

- Virtual Machine
- Defhashmap and Pair
- Unique Challenges and Solutions

Virtual machine

```
private Node reduction(Node expr){
    if(At.isAt(expr)) {
        return atExpr(expr);
    }
    else if(Builtin.isS(expr)){
        return sExpr();
    }
    else if(Builtin.isK(expr)) {
        return kExpr();
    }
    else if(Builtin.isI(expr)) {
        return iExpr();
    }
    else if(Builtin.isPLUS(expr)) {
        return plusExpr();
    }
    else if(Builtin.isPrePlus(expr)) {
        return prePlusExpr();
    }
    else if(Builtin.isMinus(expr)) {
        return minusExpr();
    }
    else if(Builtin.isPreMinus(expr)) {
        return preMinusExpr();
    }
    else if(Builtin.isMul(expr)) {
        return mulExpr();
    }
    else if(Builtin.isDiv(expr)) {
        return divExpr();
    }
    else if(Builtin.isNot(expr)) {
        return notExpr();
    }
    else if(Builtin.isCond(expr)) {
        return condExpr();
    }
    else if(Builtin.isAnd(expr)) {
        return andExpr();
    }
    else if(Builtin.isOr(expr)) {
        return orExpr();
    }
    else if(Builtin.isGrt(expr)) {
        return grtExpr();
    }
    else if(Builtin.isGrt(expr)) {
        return grtExpr();
    }
    else if(Builtin.isLes(expr)) {
        return lesExpr();
    }
    else if(Builtin.isEqu(expr)) {
        return equExpr();
    }
    else if(Builtin.isGeq(expr)) {
        return geqExpr();
    }
    else if(Builtin.isLeq(expr)) {
        return leqExpr();
    }
    else if(Builtin.isNeg(expr)) {
        return negExpr();
    }
    else if(Builtin.isColon(expr)) {
        return pair();
    }
    else if(Builtin.isHd(expr) || Builtin.isTl(expr)) {
        return headOrTail(expr);
    }
    else {
        return expr;
    }
}
```

Short code snippet of my VM

- Each method calls reduction recursively, to reduce the program completely
- Stack to beginning is empty, fills through „atExpr“ until it finds another expression
- Pairs (lists) don't get reduced in the reduction method, but in the print method
- Lists not ending in „nil“ will not be accepted, as [...] lists are not implemented

Defhashmap and pair

- Definition Nodes of AST are created with a left and a right part
 - On the left side I used a HashMap, to bind definition names to their expressions
 - On the right side is the Node, that contains the expression
- “DefHashMap” used to improve readability
- Pairs were created for the definition nodes

```
public class DefHashMap {  
    private HashMap<String, Pair<ArrayList<String>, Node>> definitions;  
  
    //Create new empty HashMap with the call DefHashMap()  
    public DefHashMap() {  
        HashMap<String, Pair<ArrayList<String>, Node>> definitions = new HashMap<String, Pair<ArrayList<String>, Node>>();  
        this.definitions = definitions;  
    }  
  
    //Create a DefHashMap built from another HashMap with the call DefHashMap(HashMap...)  
    public DefHashMap(HashMap<String, Pair<ArrayList<String>, Node>> x) {  
        this.definitions = x;  
    }  
  
    public void put(String defName, Pair<ArrayList<String>, Node> param) {  
        definitions.put(defName, param);  
    }  
  
    public HashMap<String, Pair<ArrayList<String>, Node>> returnHashMap(){  
        return definitions;  
    }  
}
```

```
public class Pair<F, S> {  
    private final F first;  
    private S second;  
  
    public Pair(F first, S second) {  
        this.first = first;  
        this.second = second;  
    }  
}
```


Challenges

- Alone, partner left early on
- No prior programming experience
- Never worked independently on a project
- Not everything was implemented due to lack of time
 - „Where“ not implemented
 - [...] lists not implemented
 - Optimizers not implemented
- Virtual Machine differs from the handout
 - Not as powerful and fast as it could be
 - Can still run programs reliably and relatively fast

ME WAITING FOR MY CODE TO
BUILD TO SEE HOW MANY ERRORS I
WILL GET



**THANKS FOR PAYING
ATTENTION**

**NOW DON'T ASK
QUESTIONS**