Semantic analysis

Marcin Kuta

Theory of Compilation
Laboratory 4

Type system

- safe or unsafe
- static or dynamic
- explicit or implicit

Type checking

Type checking

```
from collections import defaultdict
ttype = defaultdict(lambda: defaultdict(lambda:
    defaultdict(lambda: TS.UNKNOWN)))
...

ttype['+'][TS.INT][TS.INT] = TS.INT
ttype['+'][TS.INT][TS.FLOAT] = TS.FLOAT
ttype['+'][TS.FLOAT][TS.INT] = TS.FLOAT
ttype['+'][TS.FLOAT][TS.FLOAT] = TS.FLOAT
```

Type checker

Type checker

```
class TypeChecker(object):

    def visit(self, node):
        method = 'visit_' + node.__class__.__name__
        visitor = getattr(self, method,
                                self.generic_visit)
        return visitor(node)


    def visit_BinExpr(self, node):
        type1 = self.visit(node.left)
        type2 = self.visit(node.right)
        op    = node.op;
        # ...
        #
```

```
class SymbolTable(object):

    # parent scope and symbol table name
    def __init__(self, parent, name):
        ...

    # put variable symbol or fundef under <name>
    entry
    def put(self, name, symbol):
        ...

    # get variable symbol or fundef from <name> entry
    def get(self, name):
        ...

    def getParentScope(self):
        ...

    def pushScope(self, name):
        ...

    def popScope(self):
        ...
```

Break and continue

```
continue;
break;
```

References

1. https://sly.readthedocs.io/en/latest/sly.html