


What is Lvalue???

LVALUE

Some interesting programming lang in the blockchain approach

<https://sci-hub.se/https://ieeexplore.ieee.org/abstract/document/7965268>

Interesting keywords:

[https://scholar.google.com/scholar?](https://scholar.google.com/scholar?hl=es&as_sdt=0%2C5&q=programming+language+usability&btnG=)

[hl=es&as_sdt=0%2C5&q=programming+language+usability&btnG=](https://scholar.google.com/scholar?hl=es&as_sdt=0%2C5&q=programming+language+usability&btnG=)

- I think in perspective that this no need the array util, but a I need to do more research.

So is possible create a compiler with creating a language?
A smart contract that have the immutability stuff?

<https://craftinginterpreters.com/contents.html>

$$\left\langle \left\langle \$\text{PGM:Pgm} \right\rangle_k \left\langle .\text{Map} \right\rangle_{\text{state}} \right\rangle_T$$

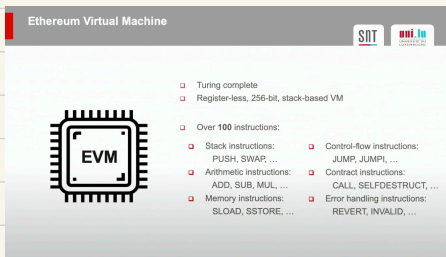
Figure 2. IMP Configuration

```
Pgm ::= "int" Ids ";" Stmt Ids ::= List{Id, ","}
AExp ::= Int | Id | "-" Int | AExp "/" AExp
       > AExp "+" AExp | "(" AExp ")"
BExp ::= Bool | AExp "<=" AExp | "!" BExp
       > BExp "&&" BExp | "(" BExp ")"
Block ::= "{" "}" | "{" Stmt "}"
Stmt ::= Block | Id "=" AExp ";"
       | "if" "(" BExp ")" Block "else" Block
       | "while" "(" BExp ")" Block > Stmt Stmt
```

Figure 3. The Syntax of IMP

Control Flow Structure Recovery

- Separate each public function subgraph
- Use structural analysis (M. Sharir 1980)
 - Match subgraphs to control constructs (while, if then else)
 - Collapse matched subgraphs



Some different thing about compiler blockchain and normal
Is that the printer, así and not stack makes more large projects.
What more differences?

<https://github.com/rust-lang/rust/blob/0.1/src/libcore/int.rs>

<https://github.com/golang/go/tree/go1.0.1/src/pkg/runtime>

some extended works

Add concurrency:

<https://sci-hub.se/https://link.springer.com/article/10.1007%2Fs00446-019-00357-z>

Theory vs practice:

Typescript roadmap show all the features that a compiler have.
<https://github.com/microsoft/TypeScript/wiki/Roadmap>