

# Summary of “Typed Assembly Language”[1]

CHRISTOPH MENDE, NIKOLAJ-JENS SCHWARTZ

---

## 1 TAL-0: CONTROL-FLOW-SAFETY

## 2 THE TAL-0 TYPE SYSTEM

The TAL-0 Type System tries to ensure that any state  $M$  always has a following state  $M'$  and as such can never get stuck. To ensure that, the type system needs to distinguish between labels and integers to make sure that the control flow follows labels and that the typing is preserved so that we cannot reach a stuck state after transferring control to a label.

The type system in our TAL classifies four different type constructors. These are *int*,  $code(\Gamma)$ ,  $\alpha$  and  $\forall\alpha.\tau$ . The first type is *int* for simple integer values. The second,  $code(\Gamma)$  is for labels and the types that are expected in the registers.  $\Gamma$  is a *register file type*, a total function that maps every register to a specific type. The last two types are for polymorphism,  $\alpha$  is a type variable that can be substituted for any type and  $\forall\alpha.\tau$  is a term that includes the type variable  $\alpha$ .

Additionally, our type system includes a  $\Psi$  that maps every label to a type.

## REFERENCES

- [1] Benjamin C. Pierce. 2004. *Advanced Topics in Types and Programming Languages*. The MIT Press.