

Declaration

fun $f(\text{in copy } x, \text{inout copy } y, \text{out copy } z, \text{ref } u)$ returns r

global g
local l

do $x := x+1; y := y+1; \dots; l := l+1$

endfun

// enthält alle vorkommenden Mechanismen

// aber Fake-Funktion im strengen Sinn,
da auch out-Param. vorkommen

Aufruf

$f(\overbrace{a1+a2}^{\text{R-wert}}, \underbrace{b}_{\text{L-wert}}, \underbrace{c}_{\text{L-wert}}, \underbrace{d}_{\text{L-wert}})$ // g global

\downarrow \downarrow \downarrow \downarrow
 x y z u

Reihenfolge

1. return
2. inout copy + out copy // Reihenfolge wichtig !
3. in copy + ref
Reihenfolge beliebig

Beispiel

$f(a1+a2, b, b, d)$

$y \rightarrow b;$ oder $z \rightarrow b;$
 $z \rightarrow b;$ $y \rightarrow b;$

↓
Steht Speicher für z ,
dann für y auch Stack

Notation

\sim : nicht gebraucht

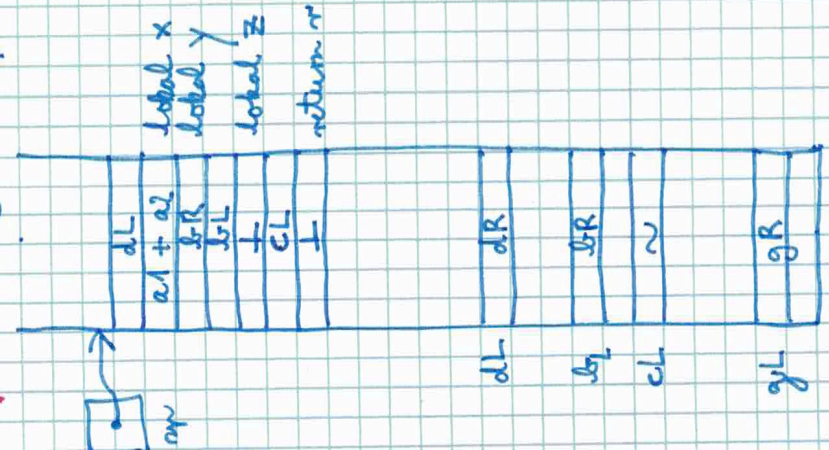
\perp : undefiniert


```

//lineBlock 1: //return r
codeLExp g II c II;
//lineBlock 1; //out copy z
codeLExp g II d II; //uninit copy y
Dump:
Dump:
codeRExp g II a1 + a2 II;
codeLExp g II d II;

```

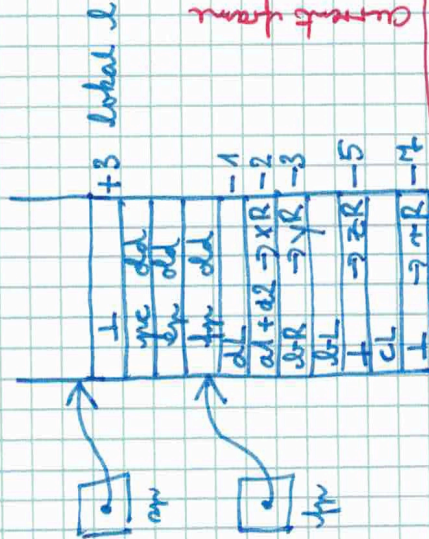
Call



Call (g(t))

dl = local 1, // local 1

Call



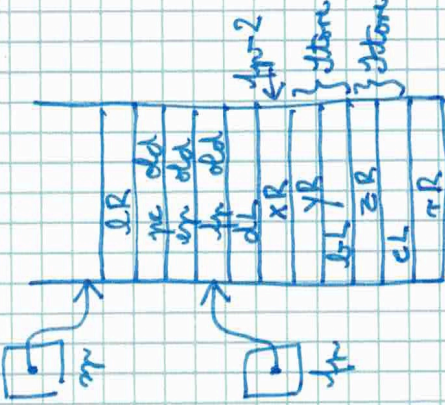
dL dR

gL gR

codeCmd g II body II

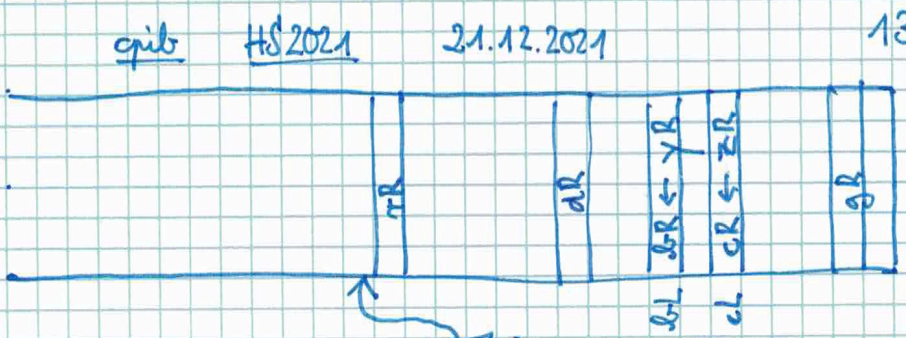
Return 2

Call



dR

gR



bL dR ← yR
cL cR ← zR

gR