An Exploration of String Diagrams in Computer Science

Dominic Verity
Centre of Australian Category Theory
Macquarie University

DRHE Workshop
University of Newcastle
10th November 2018

VECT CATEGORY OF FINITE DIMENSIONAL $\mathbb{R}$- VECTOR SPACES

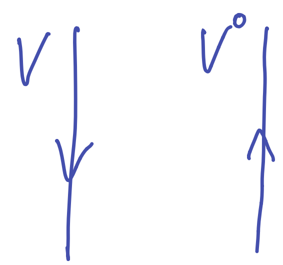
$\otimes$ TENSOR PRODUCT

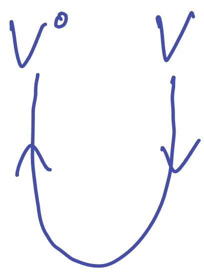
BASIS $v_1 \cdots v_n$

V A VECTOR SPACE

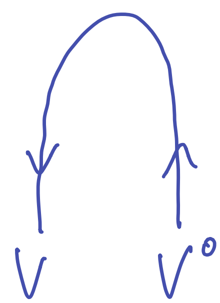
V° ITS DUAL $\mathrm{Lin}(V, \mathbb{R})$

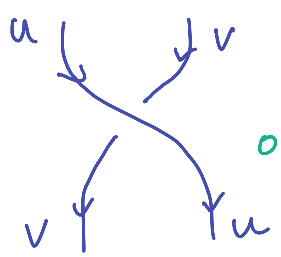
DUAL BASIS $\hat{v}_1 \cdots \hat{v}_n$





$$V^\circ \otimes V \longrightarrow \mathbb{R}$$
$$\hat{v}_i \otimes v_j \longmapsto \delta_{ij}$$



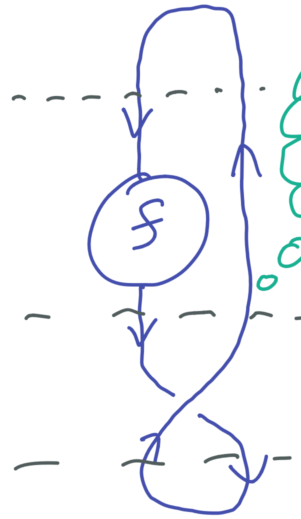


$$\mathbb{R} \longrightarrow V \otimes V^\circ$$
$$1 \longmapsto \sum v_i \otimes \hat{v}_i$$

$$s_{u,v} : U \otimes V \to V \otimes U$$
$$u_i \otimes v_j \mapsto v_j \otimes u_i$$

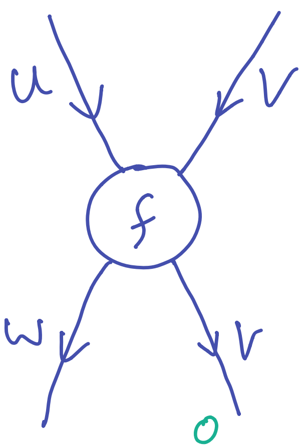
A SQUARE MATRIX

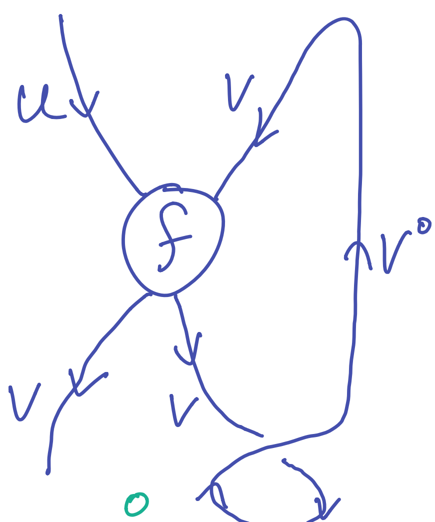
(λ_{ij})



THE TRACE OF f

$$
\begin{array}{ccccccccc}
\mathbb{R} & \longrightarrow & V\otimes V^{\circ} & \longrightarrow & V\otimes V^{\circ} & \cong & V^{\circ}\otimes V & \longrightarrow & \mathbb{R} \\
1 & \longmapsto & \sum_i e_i\otimes \hat{e}_i & \longmapsto & \sum_i f(e_i)\otimes \hat{e}_i & \longmapsto & \sum_{ij} \lambda_{ij}(\hat{e}_i\otimes e_j) & \longmapsto & \sum_i \lambda_{ii}
\end{array}
$$





$f: U\otimes V \to W\otimes V$
AN $(mn \times rn)$-MATRIX

GENERALIZED TRACE $Tr^{W}_{U,V}(f)$

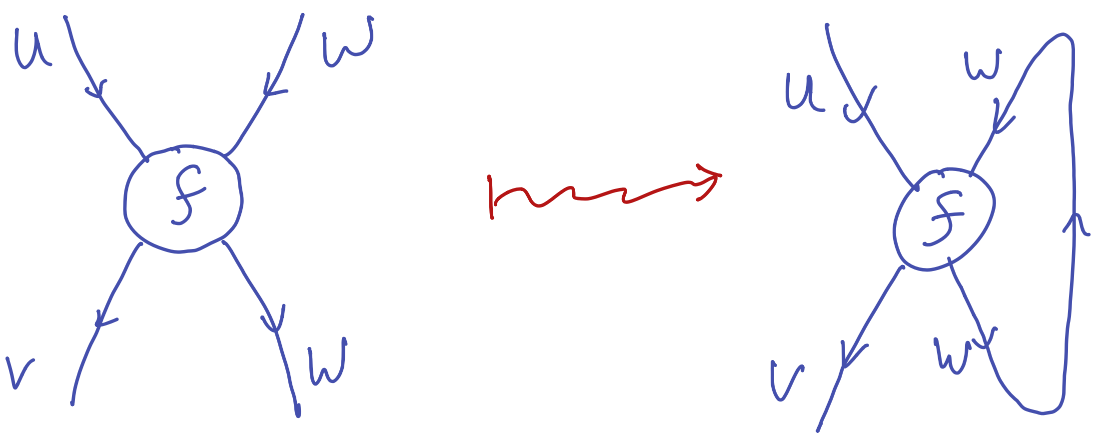
AN AXIOMATISATION OF TRACE

$(\mathcal{C}, \otimes, I)$ SYMMETRIC MONOIDAL CATEGORY

THINK $(\mathrm{Vect}, \otimes, \mathbb{R})$ IF YOU LIKE

CAN ASSUME THIS IS STRICTLY MONOIDAL!

ADD AN EXTRA OPERATION ON ARROWS

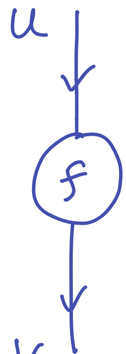


$$f : U \otimes W \to V \otimes W \rightsquigarrow Tr^{W}_{U,V}(f) : U \to V$$

$$Tr^{W}_{U,V} : Hom(U \otimes W, V \otimes W) \to Hom(U,V)$$

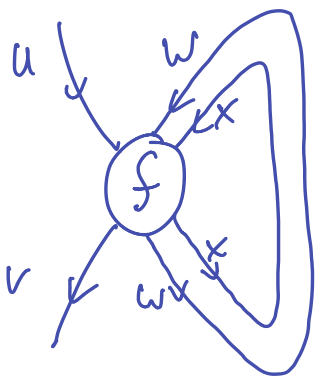
QUESTION: WHAT AXIOMS SHOULD THIS ABSTRACT TRACE OPERATION SATISFY?

ANSWER: AXIOMS ARE EASIER TO IMAGINE IF WE THINK DIAGRAMMATICALLY.

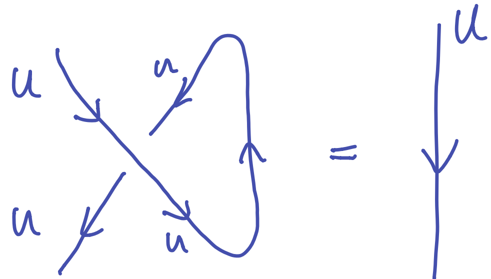


$$f = Tr^{I}_{u,v}(f)$$

VANISHING



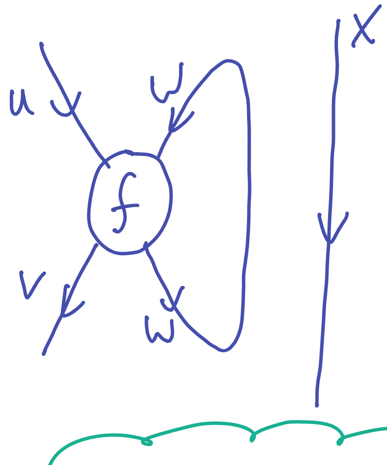
$$Tr^{w}_{u,v}\left(Tr^{x}_{u\otimes w,\, v\otimes w}(f)\right) = Tr^{w\otimes x}_{u,v}(f)$$



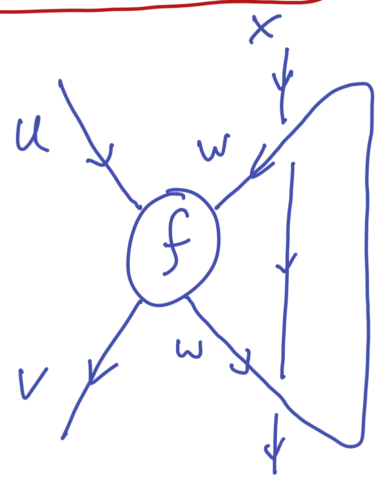
YANKING

$$Tr^{u}_{u,u}(s_{u,u}) = id_u$$

SUPERPOSITION



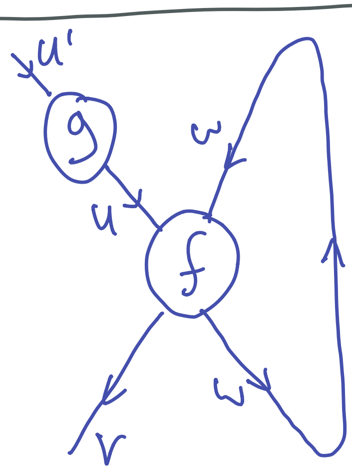
=



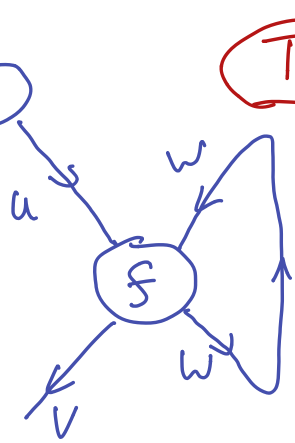
$$\mathrm{Tr}^{w}_{u,v}(f) \otimes X = \mathrm{Tr}_{u\otimes X,\, v\otimes X}\big((u \otimes s_{X,w}) \cdot (f \otimes X) \cdot (v \otimes s^{-1}_{X,w})\big)$$

TIGHTENING

"NATURALITY"



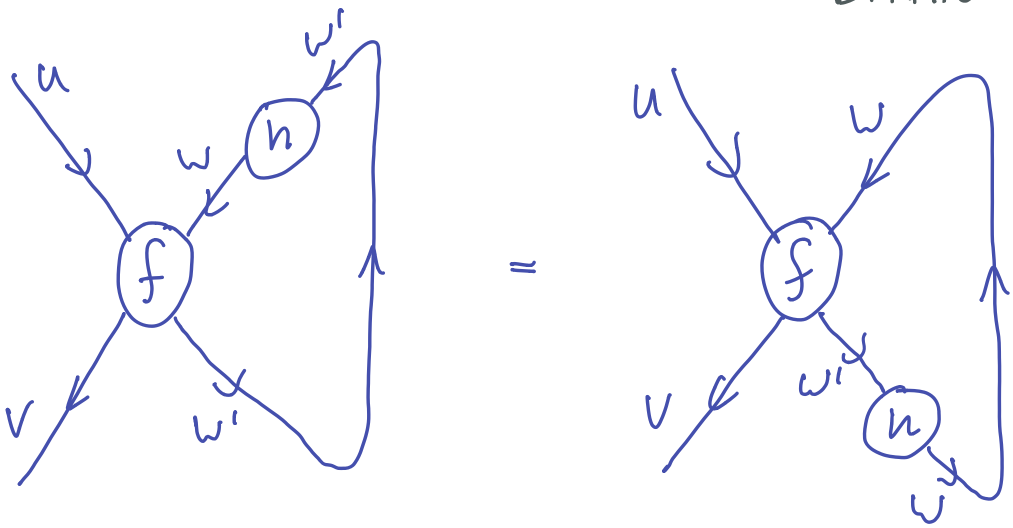
=



$$\mathrm{Tr}^{w}_{u',v}(f \cdot (g \otimes w)) = \mathrm{Tr}^{w}_{u,v}(f) \cdot g$$

SLIDING

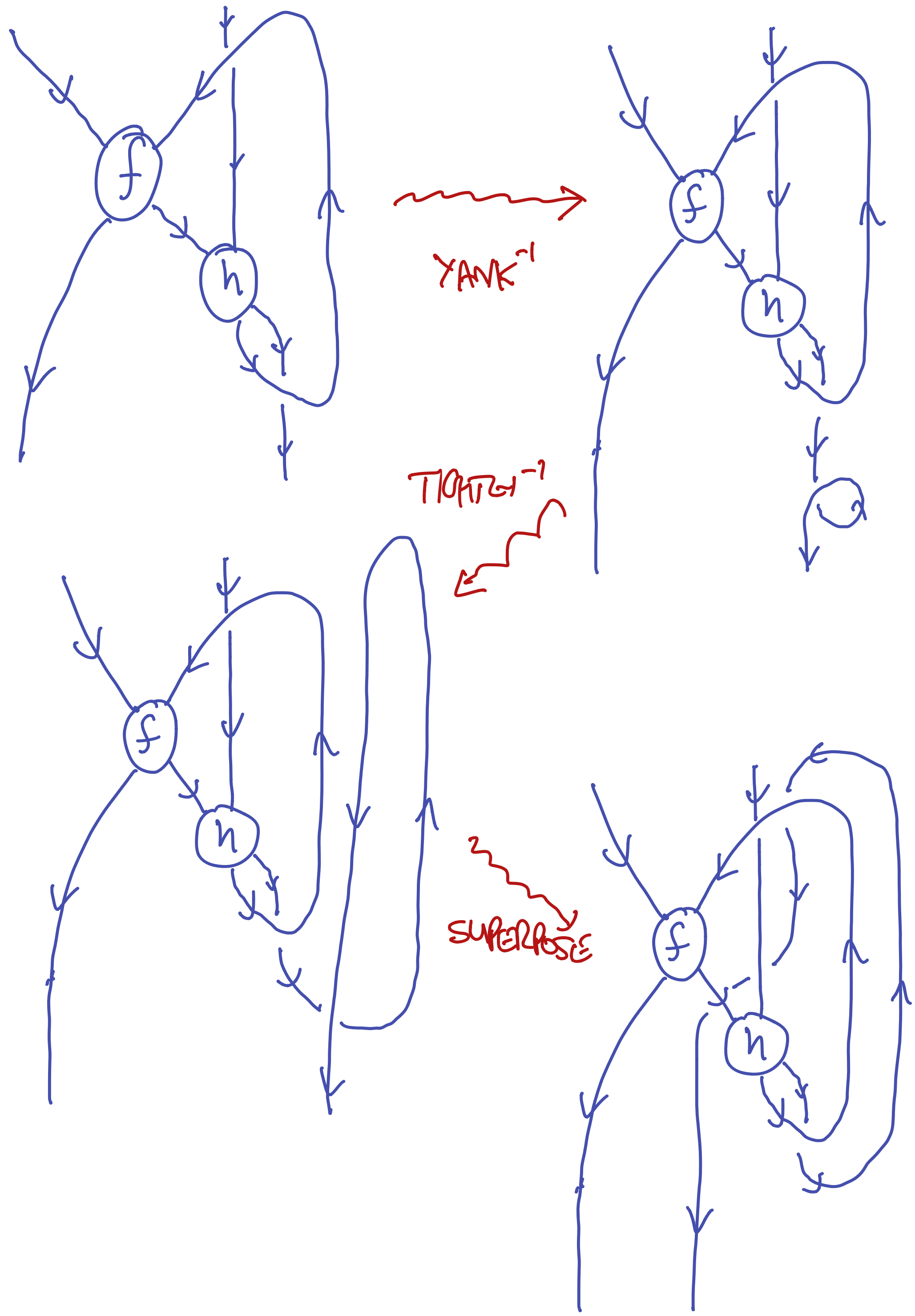
"DINATURALITY"

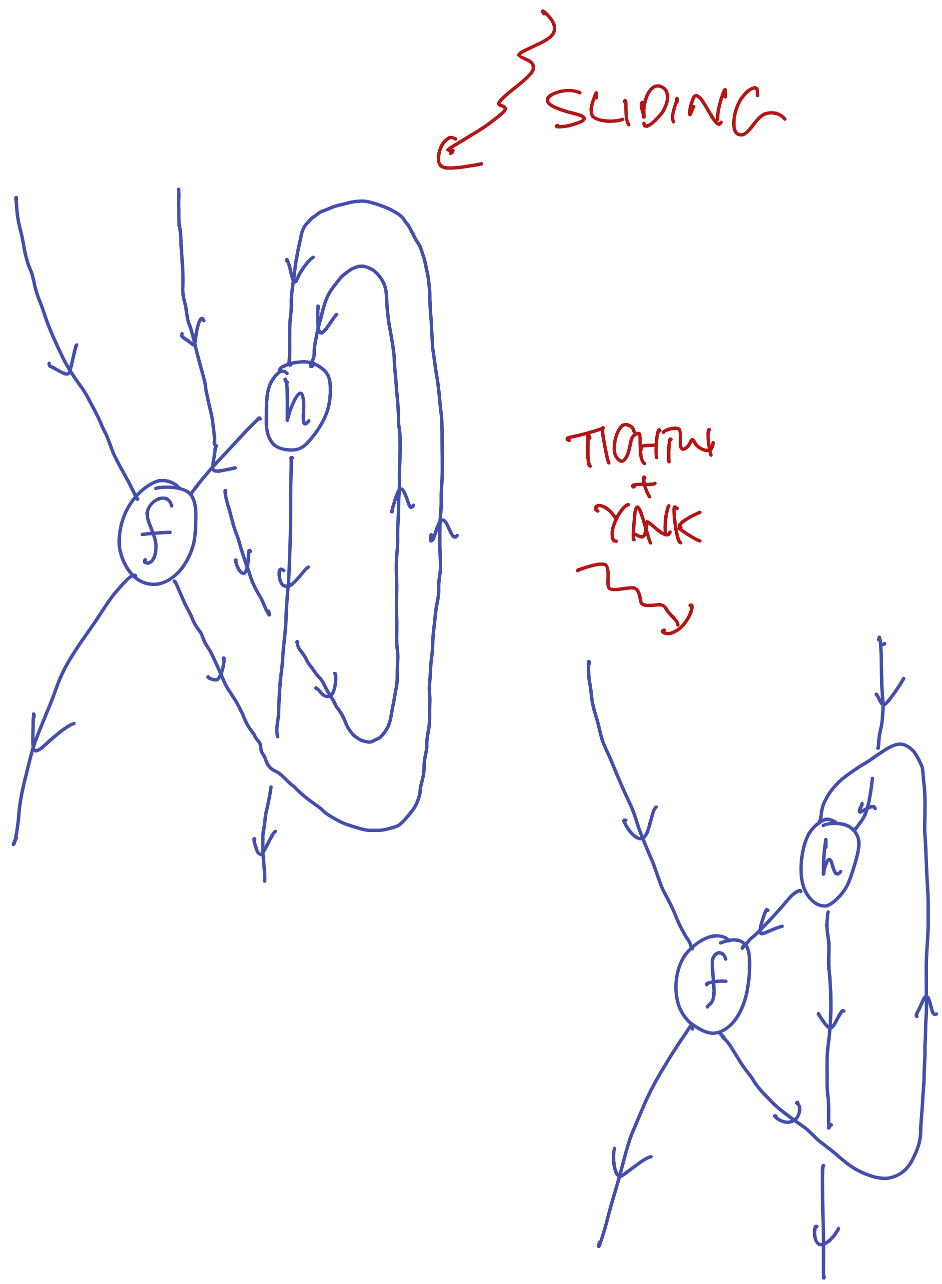


$$Tr^{w'}_{u,v}\left(f \cdot (u \otimes h)\right) = Tr^{w}_{u,v}\left((v \otimes h) \cdot f\right)$$

A MONOIDAL CATEGORY WITH AN ABSTRACT TRACE SATISFYING THESE AXIOMS IS CALLED A TRACED MONOIDAL CATEGORY.

A CALCULATION EXAMPLE

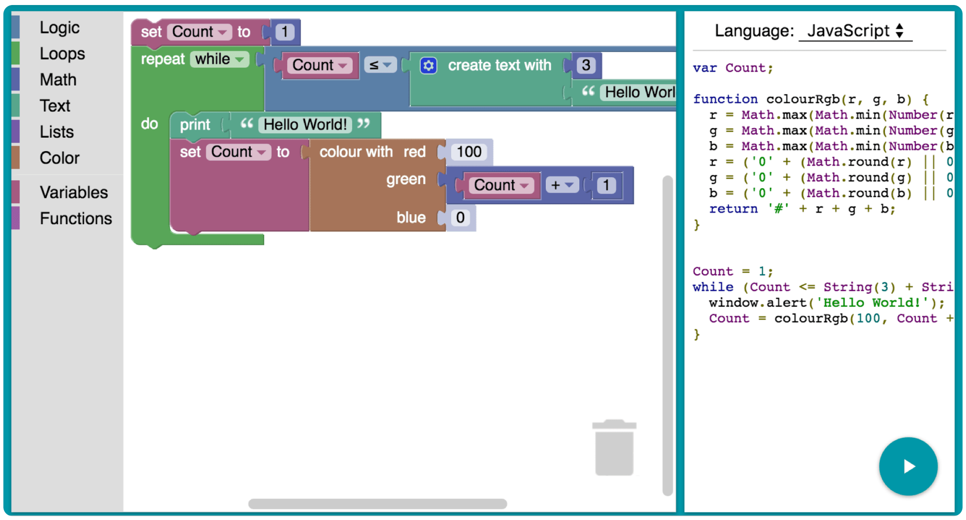




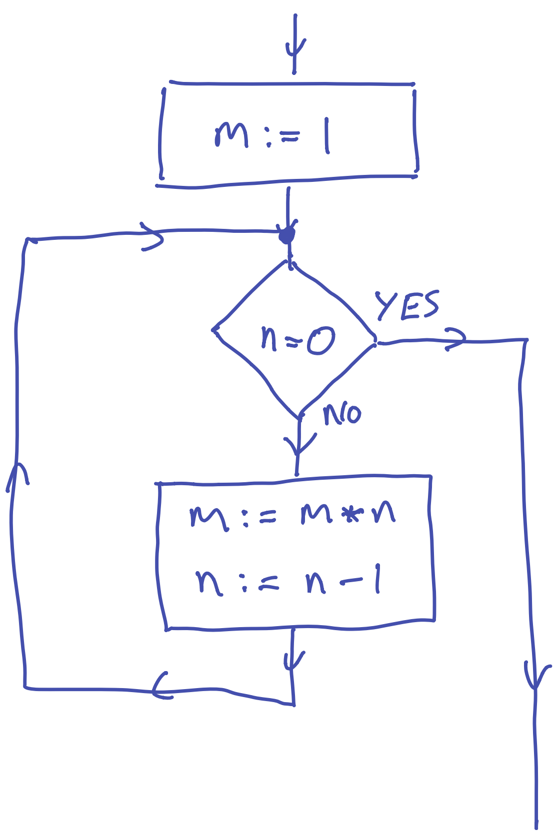
PROOF OF "COMPLETENESS" OF THESE AXIOMS – THE Int CONSTRUCTION.
A GENERALISATION OF THE CONSTRUCTION OF THE RATIONAL NUMBERS

FLOWCHARTS

(CONTROL FLOW GRAPHS)



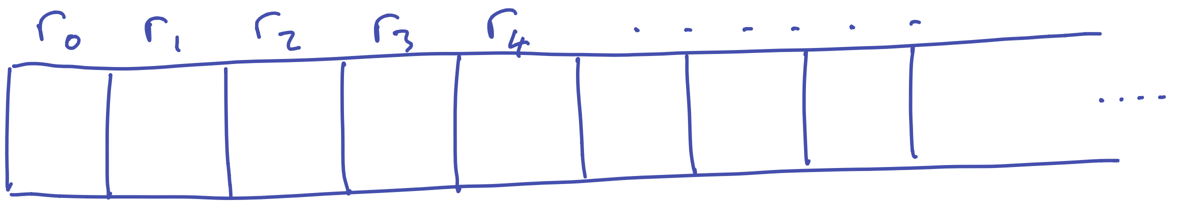
Google Blockly Flowchart View



FACTORIAL

ABACUS MACHINES

(TURING REVISITED)



A MACHINE WITH A COUNTABLY INFINITE SET OF REGISTERS, EACH OF WHICH CONTAINS A NATURAL NUMBER.

Var SET OF NAMES FOR THESE REGISTERS

VARIABLES

$s : Var \to \mathbb{N}$ A MACHINE STATE

$States := \mathbb{N}^{Var}$ SET OF POSSIBLE MACHINE STATES.

ABACUS INSTRUCTIONS

$$\boxed{x := x + 1 \ ; \ \text{GOTO } i}$$

INCREMENT

$$\boxed{\begin{array}{l} \text{IF } x = 0 \text{ GOTO } i \\ \qquad\qquad \text{ELSE } x := x - 1 \ ; \ \text{GOTO } j \end{array}}$$

CONDITIONAL DECREMENT

AN ABACUS PROGRAM

1: IF $x = 0$ GOTO 3
ELSE $x := x - 1$; GOTO 2

2: $y := y + 1$; GOTO 1

3: ...

DESTRUCTIVE ADD OF x AND y

NON-DESTRUCTIVE ADD

1: IF $z = 0$ GOTO 2
ELSE $z := z - 1$; GOTO 1

2: IF $x = 0$ GOTO 5
ELSE $x := x - 1$; GOTO 3

3: $z := z + 1$; GOTO 4

4: $y := y + 1$; GOTO 2

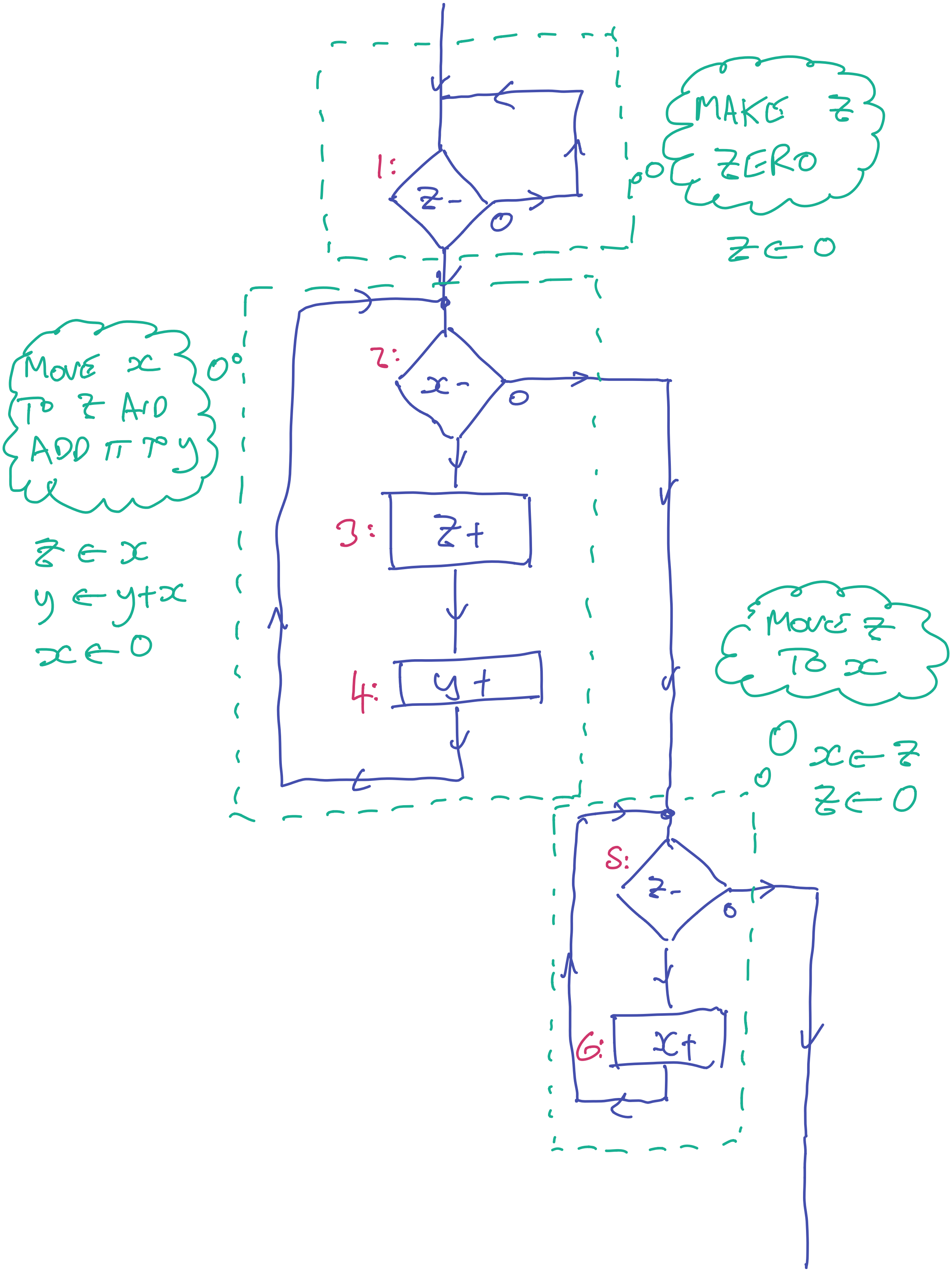
5: IF $z = 0$ GOTO 7
ELSE $z := z - 1$; GOTO 6

6: $x := x + 1$; GOTO 5

7: ....

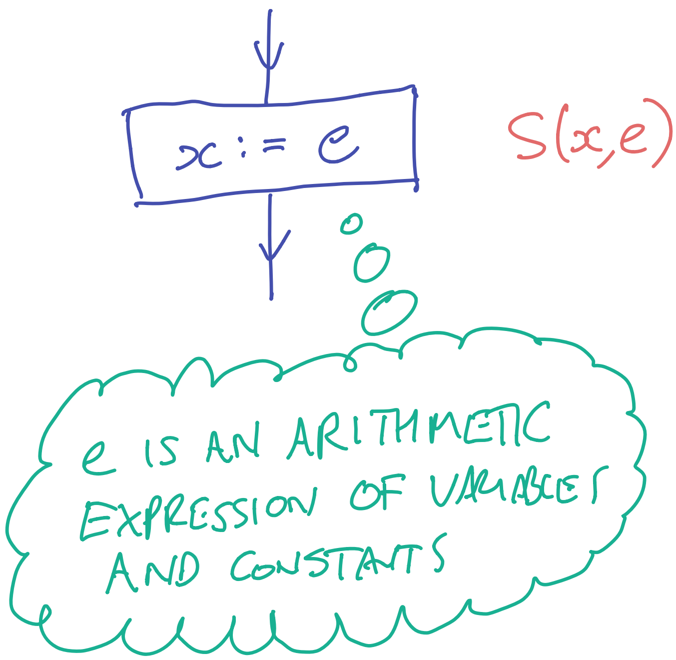
THIS IS ALREADY GETTING A LITTLE <u>ONEROUS</u>

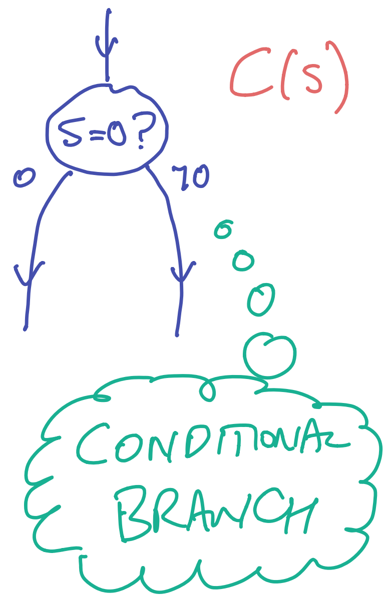
MAYBE A FLOWCHART WILL MAKE THINGS CLEARER!

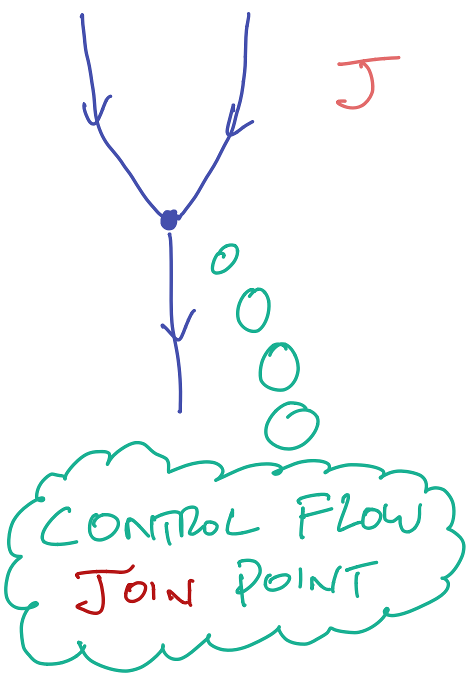

MAKE z ZERO
z ← 0
1:
z-
0
MOVE x TO z AND ADD IT TO y
z ← x
y ← y+x
x ← 0
2:
x-
0
3:
z+
4:
y+
MOVE z TO x
x ← z
z ← 0
5:
z-
0
6:
x+

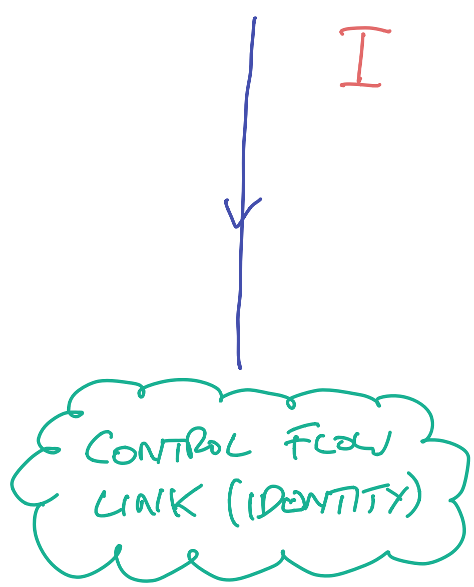
THE LANGUAGE OF FLOWCHARTS

BASIC BUILDING BLOCKS:

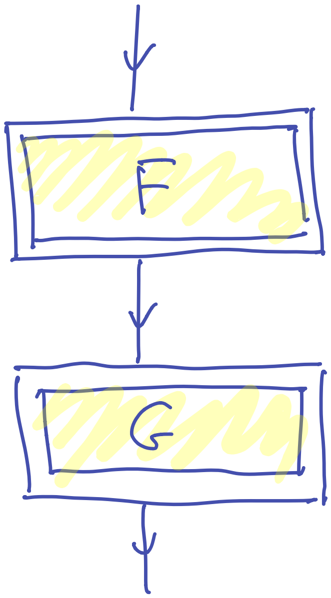








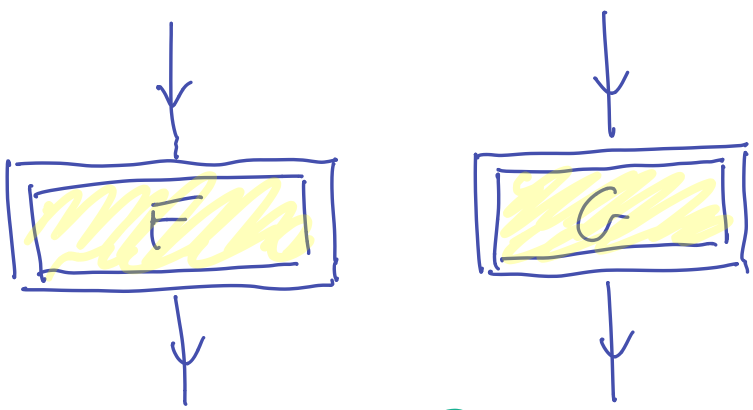
OPERATORS



$F;G$

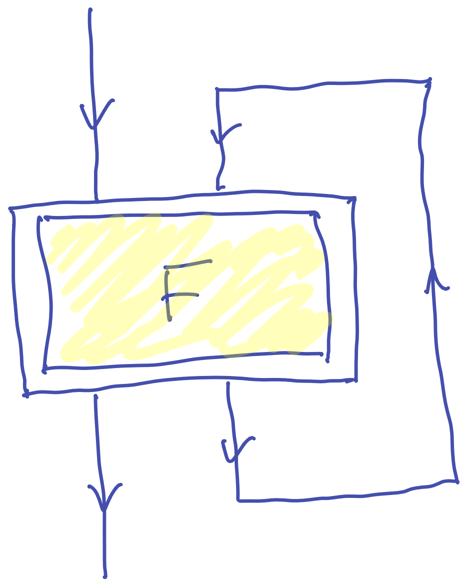
SEQUENTIAL COMPOSITION

F AND G ARE FLOWCHART TERMS



$F|G$

PARALLEL COMPOSITION



$Tr(F)$

FEEDBACK OR ITERATION

THIS LANGUAGE FREELY GENERATES A <u>CATEGORY</u> $\mathcal{L}_F$

- OBJECTS NATURAL NUMBERS n
- $S(x, e) : \underline{1} \longrightarrow \underline{1}$

  $C(b) : \underline{1} \longrightarrow \underline{2}$

  $J : \underline{2} \longrightarrow \underline{1}$ $\quad I : \underline{1} \longrightarrow \underline{1}$

  $F : \underline{n} \to \underline{m} \quad G : \underline{m} \to \underline{r} \rightsquigarrow F;G : \underline{n} \to \underline{r}$

  $F : \underline{n} \to \underline{m} \quad G : \underline{n'} \to \underline{m'} \rightsquigarrow F|G : \underline{n+n'} \to \underline{m+m'}$

  $F : \underline{n+1} \to \underline{m+1} \rightsquigarrow Tr(F) : \underline{n} \to \underline{m}$

THIS IS <u>SOME</u> KIND OF FORMAL LANGUAGE OF FLOWCHARTS

BUT IT LACKS:

- **SEMANTICS** — WHAT IS THE FORMAL MEANING OF THESE LINGUISTIC FLOWCHARTS IN TERMS OF "MECHANICAL" COMPUTATION.
- **ALGEBRA** — WITH WHICH TO PROVE THAT TWO FLOWCHARTS DEFINE THE SAME COMPUTATION

THE SEMANTIC CATEGORY

PAR THE CATEGORY OF SETS AND PARTIAL FUNCTIONS.

$+$ DISJOINT UNION

$\emptyset$ EMPTY SET - UNIT FOR $+$

THIS IS A SYMMETRIC MONOIDAL CATEGORY.

CRUCIALLY $(\mathbf{PAR}, +, \emptyset)$ ADMITS A COMPUTATIONALLY IMPORTANT TRACE

$$f : U + W \longrightarrow V + W$$

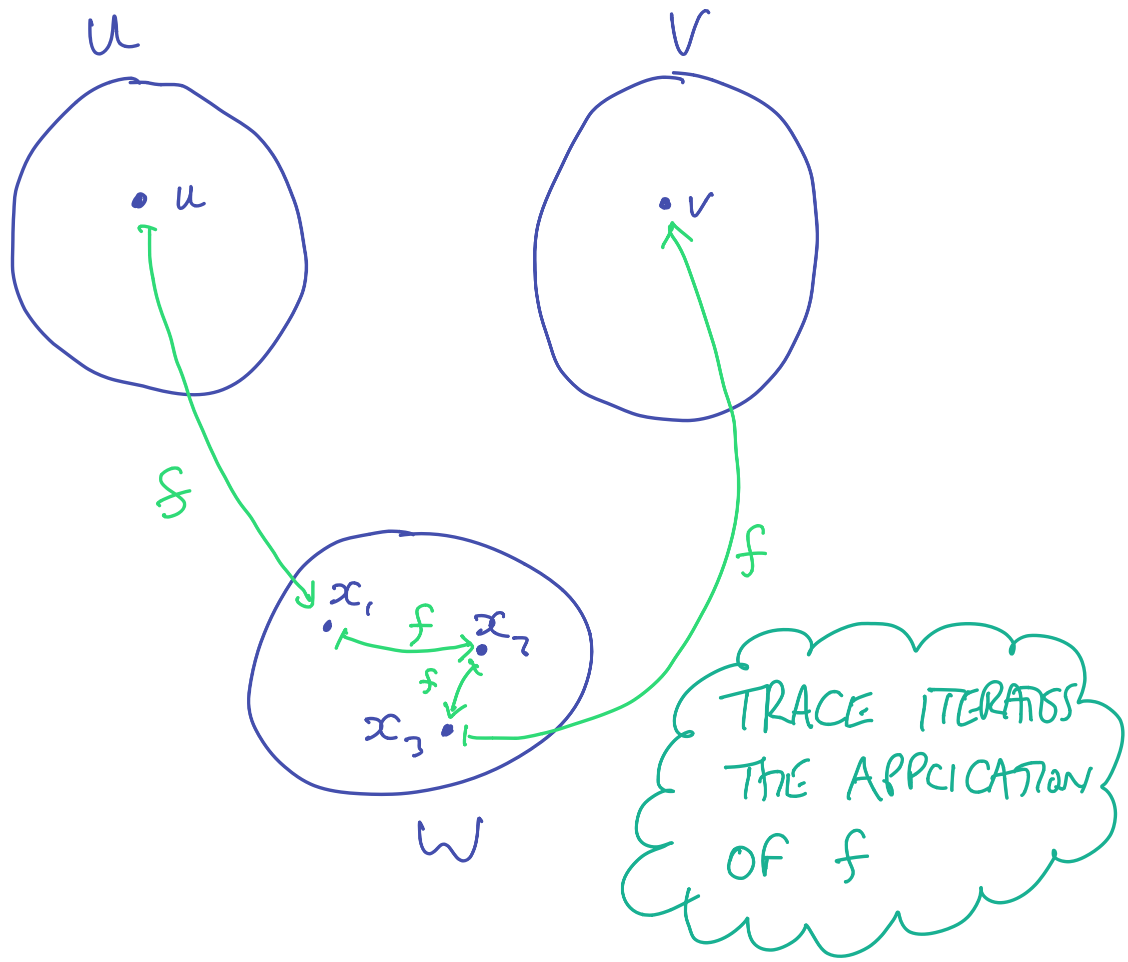
$$\mathrm{Tr}^{W}_{u,v}(f)(u) = v \quad \text{IFF} \quad \exists\, x_0 \cdots x_n \text{ S.T.}$$

$u = x_0$ AND $v = x_n$ AND $x_i \in W$ $(0 < i < n)$

AND $f(x_i) = x_{i+1}$ $(0 \leq i < n)$

WE CALL $[x_0, \ldots, x_n]$ AN "EXECUTION TRACE"

USE OF "TRACE" HERE A HISTORICAL COINCIDENCE



ITERATING f MAY NEVER LEAD US TO V.

$\Longrightarrow$

EVEN IF f IS TOTAL ITS TRACE $Tr^{W}_{U,V}(f)$ NEED NOT BE.

ALSO IF f IS PARTIAL THEN THE PROCESS OF ITERATING IT MAY GET <u>STUCK</u>

PROPOSITION THE ITERATION TRACE DEFINED ABOVE MAKES $(\underline{\mathrm{P}}_{\underline{\mathrm{AR}}}, +, \emptyset)$ INTO A (SYMMETRIC) TRACED MONOIDAL CATEGORY. $\square$

THIS TMC IS THE UR-EXAMPLE OF AN

ITERATION CATEGORY

THE SEMANTIC FUNCTOR

$$[\![\bullet]\!] : \mathcal{L}_F \longrightarrow \mathrm{Par}$$

SEMANTIC BRACKETS

RECALL

$State := \mathbb{N}^{Var}$

$$[\![\underline{n}]\!] := \underbrace{State + State + \cdots + State}_{n \text{ COPIES}}$$

$$[\![S(x, e)]\!] : State \longrightarrow State$$

$$s \longmapsto s[x \mapsto [\![e]\!](s)]$$

COMPUTE e WITH VALUES OF VARIABLES GIVEN BY s THEN BIND x TO THAT VALUE

$$
\begin{aligned}
[\![C(b)]\!] : \mathrm{State} &\longrightarrow \mathrm{State} + \mathrm{State} \\
s &\longmapsto \begin{cases} in_0(s) & \text{IF } s(b) = 0 \\ in_1(s) & \text{IF } s(b) \neq 0 \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
[\![J]\!] : \mathrm{State} + \mathrm{State} &\longrightarrow \mathrm{State} \\
in_0(s) &\longmapsto s \\
in_1(s) &\longmapsto s
\end{aligned}
$$

$$[\![F ; G]\!] := [\![G]\!] \circ [\![F]\!]$$

COMPUTER SCIENTISTS LIKE TO COMPOSE THEIR IMPERATIVE PROGRAMS IN THE "NATURAL ORDER"

$$[\![F \mid G]\!] := [\![F]\!] + [\![G]\!]$$

$$[\![Tr(F)]\!] := Tr^{[\![i]\!]}_{[\![n]\!], [\![m]\!]}([\![F]\!])$$

WE SAY THAT THIS SEMANTICS IS DEFINED COMPOSITIONALLY

PROPOSITION

A PARTIAL FUNCTION

$$f : \mathrm{State} \longrightarrow \mathrm{State}$$

DESCRIBES THE INPUT/OUTPUT BEHAVIOUR OF AN ABACUS MACHINE IFF THERE EXISTS SOME $F : \underline{1} \to \underline{1}$ IN $\mathcal{L}_F$ WITH $f = [\![F]\!]$. $\square$

PROPOSITION

ON TAKING EQUIVALENCE CLASSES OF ARROWS OF $\mathcal{L}_F$ UNDER THE EQUIVALENCE RELATION:

$$F \sim G \Leftrightarrow [\![F]\!] = [\![G]\!]$$

WE GET A CATEGORY <u>*Flow*</u>

THIS INHERITS A TRACED MONOIDAL CATEGORY STRUCTURE FROM <u>Par</u> WHICH IS COMPATIBLE WITH THE INTENDED MEANINGS OF THE COMBINATORS $F;G$, $F|G$ AND $Tr(F)$. $\square$

THIS IS THE TMC OF FLOWCHARTS FOR THIS NOTION OF COMPUTATION.

WE MIGHT NOW IDENTIFY FLOWCHART DIAGRAMS WITH DIAGRAMS IN THE TMC Flow.

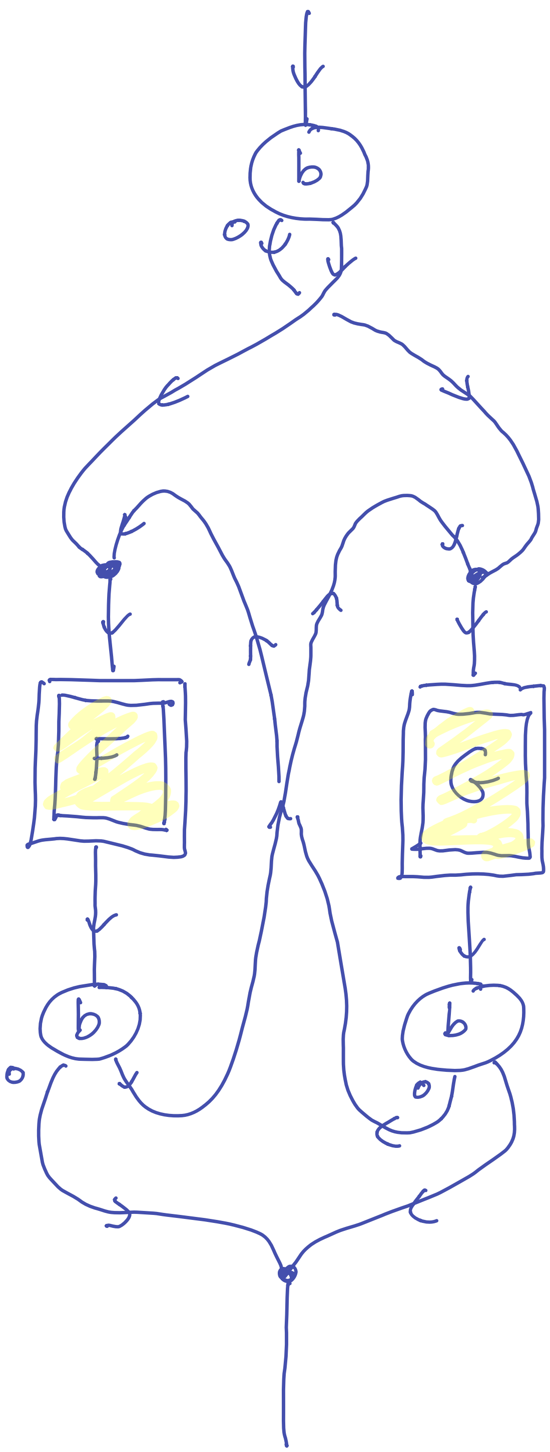
$\Longrightarrow$

MANY ALGEBRAIC PROPERTIES OF FLOWCHARTS MAY BE DERIVED FROM THE TMC FORMALISM

$\Longrightarrow$

FLOWCHARTS THAT ARE SHOWN TO BE EQUAL USING TMC DIAGRAMMATIC REASONING DEFINE THE SAME MACHINE

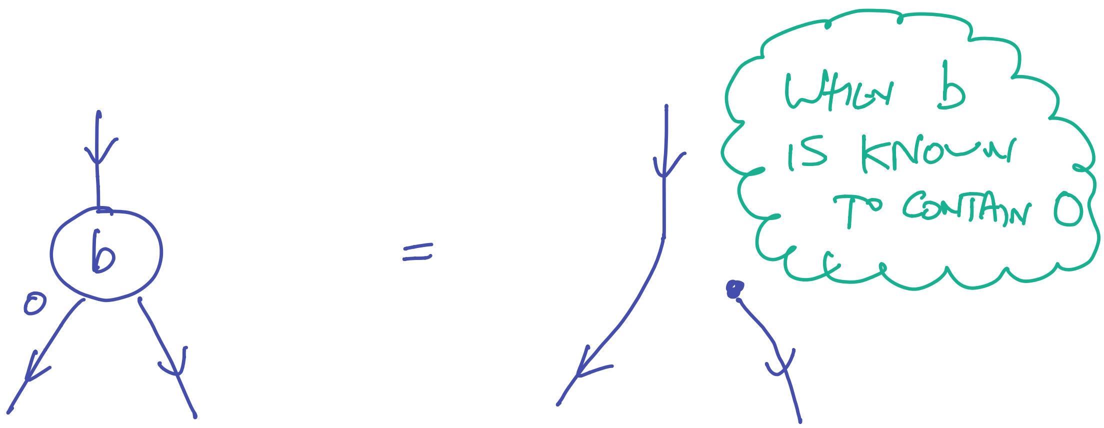
EXAMPLE

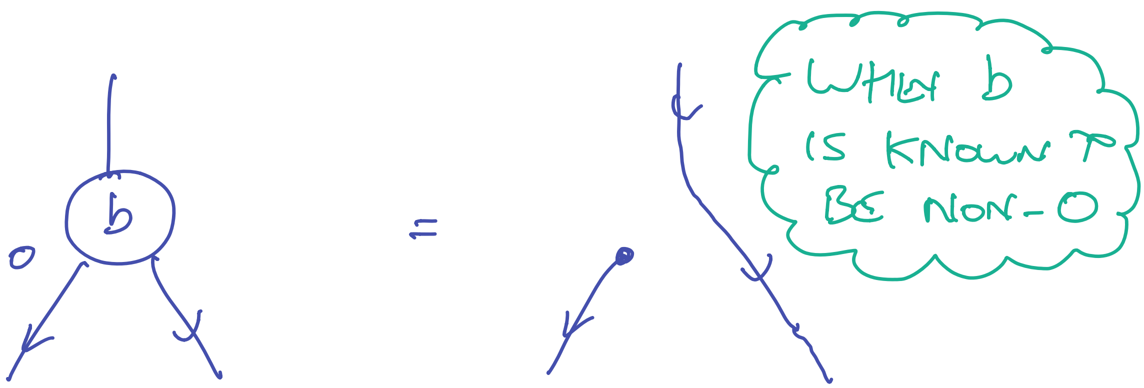


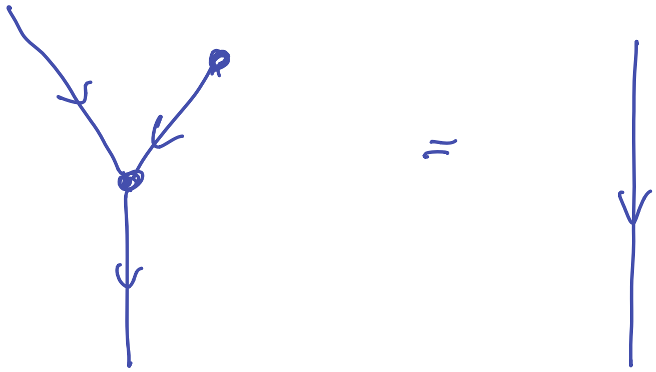
SIDE CONDITION

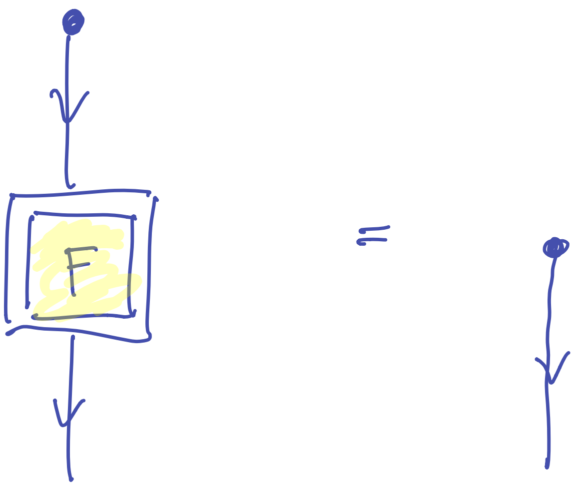
THE FLOWCHARTS F AND G DON'T INVOLVE ANY OPERATIONS THAT "TOUCH" REGISTER b

SOME DERIVABLE COMPUTATIONAL PRINCIPLES.







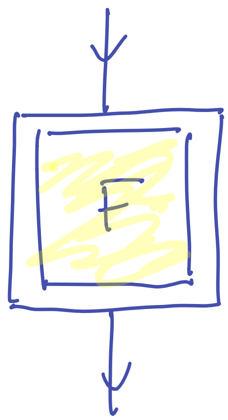


COMPILER WRITERS CALL THIS LAST EQUATION:

DEAD CODE ELIMINATION

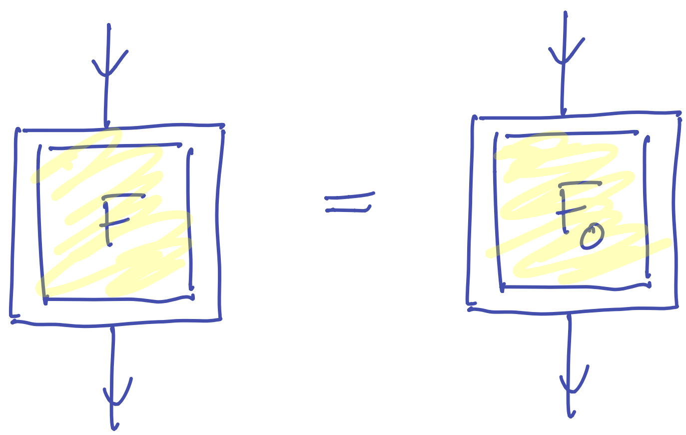
FINALLY: ARGUMENT BY CASES

IF



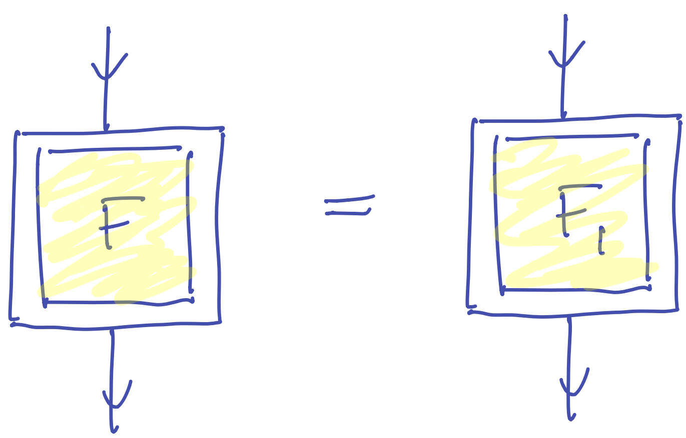
IS KNOWN NOT TO TOUCH THE VALUE OF b

AND



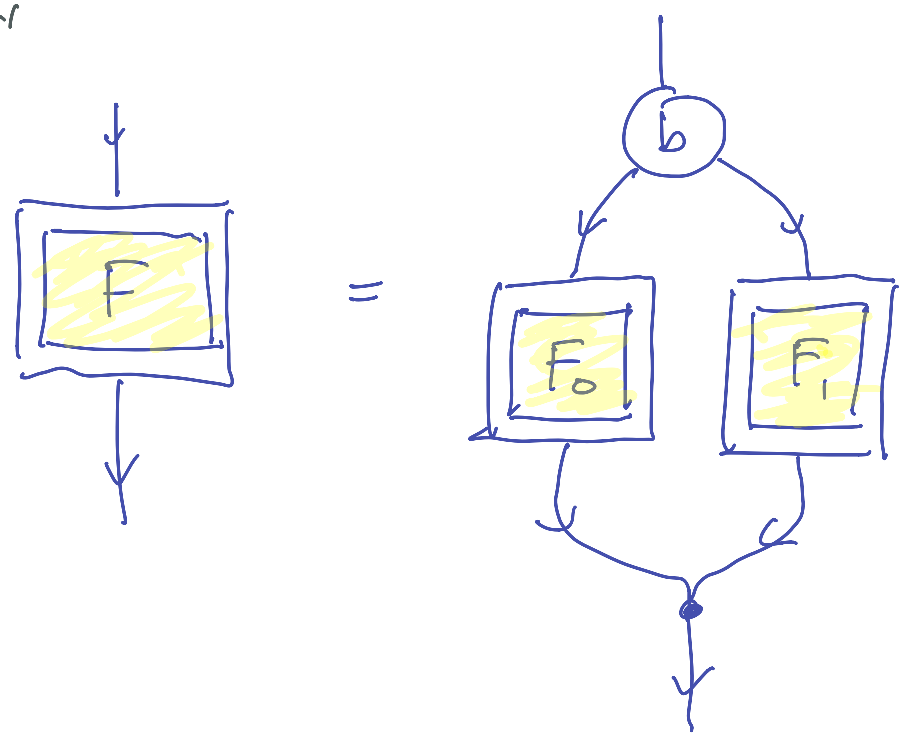
WHEN b IS KNOWN TO BE 0

AND

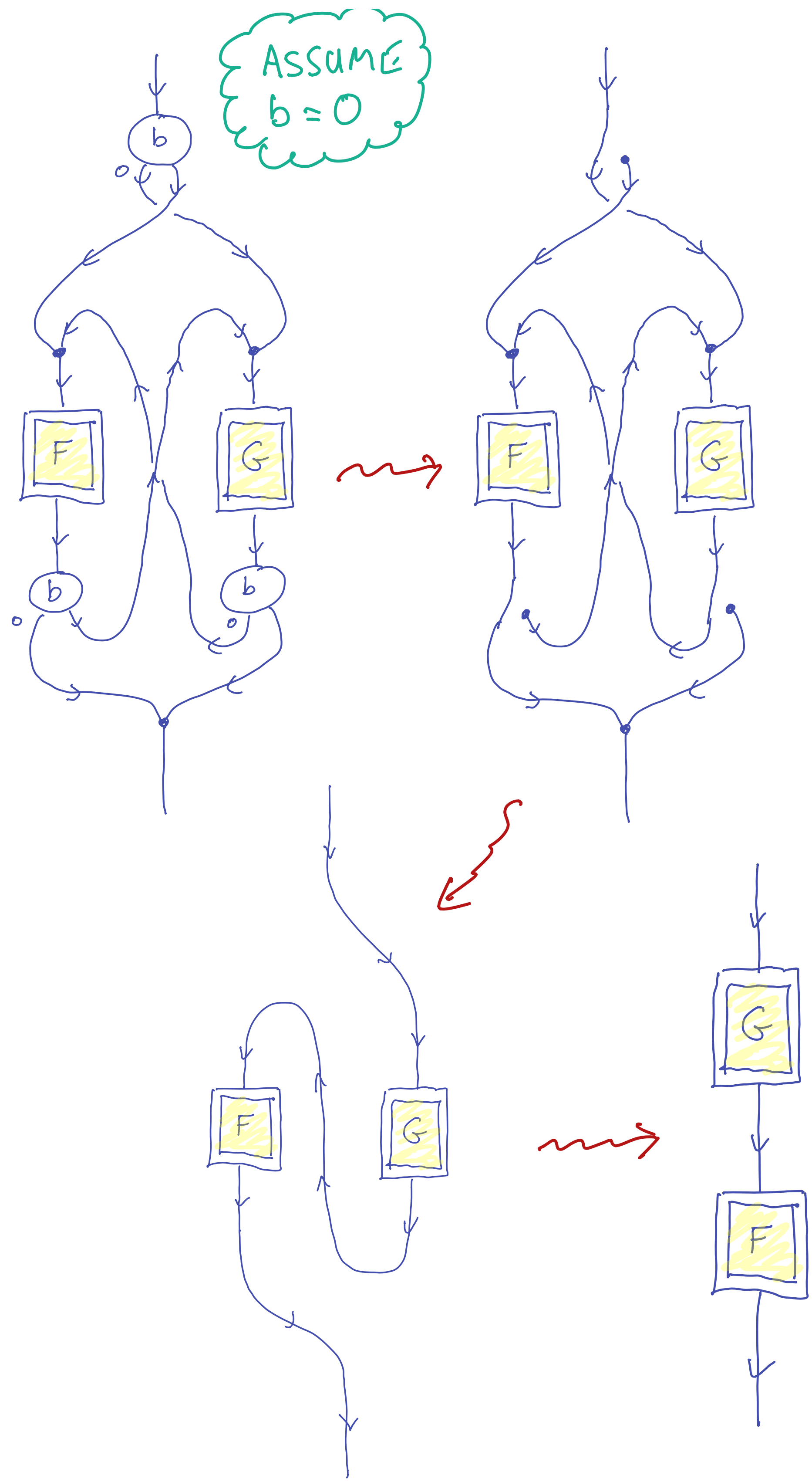


WHEN b IS KNOWN TO BE NON-0

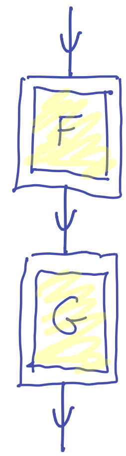
Theor



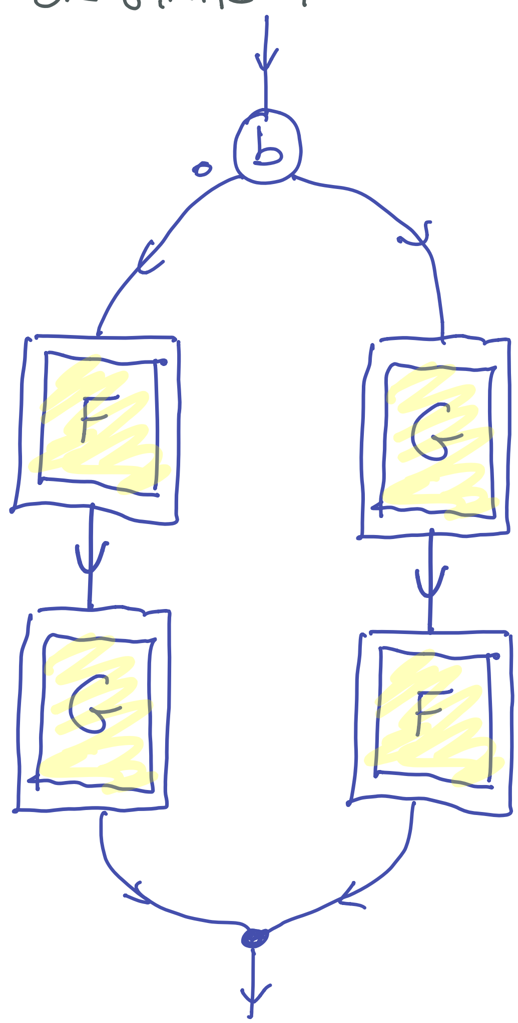
IN Flow


ASSUME
b = 0
b
0
F
G
b
0
b
0
F
G
F
G
G
F

DUALLY WHEN $b \neq 0$ OUR FLOWCHART REDUCES TO



SO APPLYING THE ARGUMENT BY CASES RULE OUR ORIGINAL FLOWCHART REDUCES TO:



NOTE: THE ORIGINAL FLOWCHART INVOLVED FEEDBACK, BUT THIS CALCULATION SHOWS THAT IT DIDN'T ACTUALLY GIVE RISE TO AN ITERATIVE ALGORITHM

FALSE FEEDBACK

- ARTHAN, MARTIN, MATHIESEN + OLIVA
  USE THIS FRAMEWORK TO SHOW THAT FLOYD-HOARE LOGICS ARISE FROM CERTAIN TRACE PRESERVING FUNCTORS

"A GENERAL FRAMEWORK FOR SOUND + COMPLETE FLOYD-HOARE LOGICS".
ACM TRANSACTIONS OCT 2009

- HYLAND
  PROVIDES AN ABSTRACT PROOF OF KLEENE'S THEOREM IN TERMS OF A TRACE PRESERVING FUNCTOR FROM
  FINITE STATE AUTOMATA
  TO REGULAR LANGUAGES

"ABSTRACT + CONCRETE MODELS FOR RECURSION" 2008

- GHICA + JUNG
  VARIOUS APPLICATIONS TO THE MODELLING OF DIGITAL CIRCUITS.
  "THE GEOMETRY OF SYNTHESIS"