What is computation?

Daniel Murfet
October 2014
therisingsea.org

1407.2650 “Logic and linear algebra”
1406.5749 “On Sweedler’s cofree cocommutative coalgebra”
1402.4541 “Computing with cut systems”
• Turing machines, Lambda calculus, logic…
• Semantics : syntax :: representations : group
• Homotopy type theory (Awodey, Voevodsky 2012)
• Girard “Towards a Geometry of Interaction” (1989)
Sense & Denotation

• Frege “On sense and denotation” (1892)

• A sentence denotes or refers to some external object, and expresses its sense, which is the ‘mode of presentation’ of its denotation.

  \[2 \times 2 = 4\]

  same denotation, different sense
Sense as algorithm

\[ 2 \times 2 = 4 \]

\[ \text{mult}_2(2) = 4 \]

Turing machine  \quad \text{input}  \quad \text{output}

\[ \begin{array}{c}
\cdots \hline 2 \hline \cdots \\
\triangle \text{mult}_2
\end{array} \quad \rightarrow \quad \begin{array}{c}
\cdots \hline \text{computation} \\
\cdots \hline 4 \hline \cdots \\
\triangle
\end{array} \]
Sense as topology

\[2 \times 2 = 4\]

proof-nets = diagrammatics of linear logic
Sense as algebra

\[ \mathcal{T} = \mathbb{Z}_2\text{-graded triangulated category} \quad [1] \circ [1] = \text{id} \]

\[ \text{End}^*_\mathcal{T}(Y) = \text{Hom}_\mathcal{T}(Y, Y) \oplus \text{Hom}_\mathcal{T}(Y, Y[1]) \]

\[ C = \mathbb{Z}_2\text{-graded algebra} \]

A \( C \)-module in \( \mathcal{T} \) is a morphism \( C \rightarrow \text{End}^*_\mathcal{T}(Y) \)
\( \mathcal{T} = \mathbb{Z}_2\)-graded triangulated category \quad [1] \circ [1] = \text{id} \\

**Example**

\[
C_1 = \text{End}_k(k \oplus k[1])
\]

\[
a = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad a^\dagger = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}
\]

\[
= k\langle a, a^\dagger \rangle \text{ with } a^2 = (a^\dagger)^2 = 0, aa^\dagger + a^\dagger a = 1
\]

\[C_n = k\langle a_1, \ldots, a_n, a_1^\dagger, \ldots, a_n^\dagger \rangle \text{ with Clifford relations}
\]

\( \mathcal{T}^\bullet = C_n\)-modules in \( \mathcal{T} \) for \( n \geq 0 \)

\[C_0 = k\]
Sense as algebra

\[ \mathcal{T} = \mathbb{Z}_2\text{-graded triangulated category} \]

A \( C_1 \text{-module in } \mathcal{T} \) is \( (Y, a, a^\dagger) \)

\[ Y \cong X \oplus X[1] \quad X = \text{Im}(aa^\dagger) \]

\( (Y, a, a^\dagger) \cong X \quad 2 \times 2 = 4 \)

same denotation, different sense
• A bicategory has objects, 1-morphisms and 2-morphisms, and composition functors

\[ \mathcal{B}(b, c) \times \mathcal{B}(a, b) \to \mathcal{B}(a, c) \]

\[(\text{mult}_2, 2) \mapsto 4 = X\]

\[\text{mult}_2(2)\]

• A cut system is similar, except it has \textit{cut functors}

\[ \mathcal{B}(b, c) \times \mathcal{B}(a, b) \to \mathcal{B}(a, c)^\bullet \]

\[(\text{mult}_2, 2) \mapsto (Y, a, a^\dagger)\]

\[\text{mult}_2\]
Theorem

• There is a bicategorical semantics of intuitionistic propositional linear logic in the cocompletion of a cut system $\mathcal{B}$ defined on the bicategory of Landau-Ginzburg models (hypersurface singularities and matrix factorisations).

• Lambda calculus embeds in intuitionistic linear logic

• The Clifford actions are derived from Atiyah classes of matrix factorisations (homological perturbation lemma under the hood).
• Universal examples of same denotation, different sense: Turing machines, proof-nets, Clifford representations in triangulated categories (?)

\[(Y, a, a^\dagger) \cong X \quad 2 \times 2 = 4\]
The adjoint $Y \to X \to Z$ of a morphism $\phi : X \otimes Y \to Z$ is depicted

![$V = \text{universal coalgebra over } V$](image)
\[ \text{int}_A = !(A \rightarrow A) \rightarrow (A \rightarrow A) \quad \alpha \mapsto \alpha^2 \]
\[ 2 : !(A \to A) \to (A \to A) \]

\[ \tilde{2} : !(A \to A) \to !(A \to A) \]
We now enumerate these diagrammatic transformations. From $(4.5) = (4.3)$, with this example is a trivial one, the reader can easily imagine a similar calculation whose explicit truth. Although some work was necessary to convert this implicit truth into explicit algorithm – cut-elimination – the knowledge is certainly the models of computation listed above is a tension between the essence of these processes? This is a question at least as deep as "What is space?" and it mental concept in mathematics. But what is it, really? One answer is that computation is $\text{int}_A$. The purpose of this section is to explain some of his ideas and how they have motivated the author's search for geometric models of computation. One of the most interesting aspects of linear logic is Girard's program to study the semantics of the cut-elimination process. He calls this the geometry of interaction; see \[5 \text{ The geometry of interaction}\]. 

Consider the proofs $\text{mult}$. Since this answer is derived from a deterministic $\text{cut}$ $\text{mult}$, so we conclude that (at least at the level of the denotations) the output of the program $\text{mult}$ as input to $\text{mult}$.

At this point the promotions cancel with the derelictions by the identity (4.4) = (4.6). The purpose of this section is to explain some of his ideas and how when a Turing machine is iterated, or what happens during the $\text{cut}$ $\text{mult}$, with $|A| = 4.1$. We feed 2

\[\text{int}_A\] to

\[\text{int}_A\]

\[\text{int}_A\]

\[\text{int}_A\]

\[\text{int}_A\]