Linear Logic: Resources, Dialogue, Narrative, and Geometry

N-20221005

§1.1 First orientation

This note is reconstructed from a ten-page PDF titled *The Geometry of Dialogue and the Logic of Narrative: A Taste of Linear Logic*, dated 2022-10-05. The original text was not a conventional mathematical note. It was closer to a reflective synthesis: a recent immersion into linear logic, several remarks attributed to Dr. Keyao Peng, an interest in game design and emergent narrative, and a set of visual fragments from Ludics, sequent calculus, and Geometry of Interaction.

The background matters. The original memo was written at the moment when logic stopped looking like a static taxonomy of truth values and began to look like a theory of interaction. The guiding words were:

$$\text{interaction}, \qquad \text{resource}, \qquad \text{game}, \qquad \text{narrative}.$$

The reconstruction below keeps this origin, but changes the form. Instead of an essay moving from metaphor to metaphor, this note is organized as a usable mathematical notebook: definitions first, then small calculi, then code-like models, and finally the more philosophical interpretation.

> **Remark 1.1.1 —** The original PDF repeatedly tried to connect three worlds: a child's fable, a philosophical debate, and high-level topology or geometry of interaction. The stable mathematical bridge between them is linear logic's treatment of resources and duality.

§1.2 From truth to resources

Classical logic treats propositions as reusable truths. If A is available, then one may use A as often as needed. This is encoded by the structural rules:

$$\text{weakening: ignore a hypothesis}, \qquad \text{contraction: duplicate a hypothesis}.$$

Linear logic removes these structural rules by default. A hypothesis is a resource. If it is consumed by an inference, it is no longer freely available.

Definition 1.2.1 (Linear implication)**.** The linear implication

$$A \multimap B$$

means: given one resource of type A, one may consume it to produce one resource of type B. It is not the same as the classical implication $A \to B$, because A is not automatically reusable.

The exponential modality ! marks reusable resources. A useful slogan is:

$$A \to B \quad \text{behaves like} \quad !A \multimap B.$$

The ordinary implication first makes A reusable, then spends it linearly.

classical reading	linear reading	informal meaning
$A \to B$	$!A \multimap B$	reusable assumption produces B
$A \otimes B$	tensor	both resources are present together
$A \& B$	additive conjunction	one must be ready for either choice
$A \oplus B$	additive disjunction	one side is chosen by the prover

The essential change is not symbolic. It is operational. Linear logic asks what a proof *does* to its premises.

§1.3 Narrative as a resource pipeline

The PDF's clearest concrete example is the story of the Three Little Pigs. The story is treated as a multiset-rewriting system. Characters and materials are resources. Events are linear implications.

Let the initial state be

$$\Delta_0 = \{\texttt{pig}, \texttt{pig}, \texttt{pig}, \texttt{straw}, \texttt{sticks}, \texttt{bricks}, \texttt{wolf}\}.$$

The story rules are:

$$\begin{aligned}
r_1 &: \texttt{pig} \otimes \texttt{straw} \multimap \texttt{straw_house},\\
r_2 &: \texttt{pig} \otimes \texttt{sticks} \multimap \texttt{stick_house},\\
r_3 &: \texttt{pig} \otimes \texttt{bricks} \multimap \texttt{brick_house},\\
r_4 &: \texttt{wolf} \otimes \texttt{straw_house} \multimap \texttt{wolf},\\
r_5 &: \texttt{wolf} \otimes \texttt{stick_house} \multimap \texttt{wolf},\\
r_6 &: \texttt{wolf} \otimes \texttt{brick_house} \multimap \texttt{brick_house}.
\end{aligned}$$

The canonical ending of the fable is the sequent

$$\Delta_0 \vdash \texttt{brick\_house}.$$

This says: starting from the initial resources, there is a linear proof whose final surviving resource is the brick house.

Example 1.3.1 (A code-level model of the fable)

The following pseudo-Python is often clearer than a large proof tree.

```
from collections import Counter

state = Counter({
    "pig": 3,
    "straw": 1,
    "sticks": 1,
    "bricks": 1,
    "wolf": 1,
})

rules = [
    (("pig", "straw"),        "straw_house"),
    (("pig", "sticks"),       "stick_house"),
    (("pig", "bricks"),       "brick_house"),
    (("wolf", "straw_house"), "wolf"),
    (("wolf", "stick_house"), "wolf"),
    (("wolf", "brick_house"), "brick_house"),
]

def apply(rule, state):
    inputs, output = rule
    new_state = state.copy()
    for x in inputs:
        if new_state[x] <= 0:
            return None
        new_state[x] -= 1
    new_state[output] += 1
    return +new_state

story = [rules[0], rules[1], rules[2], rules[3], rules[4], rules[5]]
for rule in story:
    state = apply(rule, state)

assert state == Counter({"brick_house": 1})
```

This is exactly the resource-sensitive reading: a pig used to build the straw house is consumed; it cannot simultaneously remain an unused pig. The wolf survives the destruction of the straw and stick houses, but not the encounter with the brick house.

The proof tree in the original PDF should therefore be read as a storyboard. Each inference step is not only a logical step but a plot event. The proof search is the narrative search.

§1.4 Tensor, additives, and branching stories

The tensor $A \otimes B$ means that both resources are available together. In the fable,

$$\texttt{pig} \otimes \texttt{straw}$$

means that a pig and straw must both be present for the event r_1 to fire.

Additives model choice. The connective $A \& B$ means that the environment may choose which side to test; the proof must be ready for both. This is useful for games and branching narratives.

A compact narrative grammar is:

linear-logic object	narrative reading
Δ	current multiset of actors and props
$A \otimes B$	co-presence of resources
$A \multimap B$	event consuming A and producing B
$A \& B$	branch where the environment chooses
$A \oplus B$	branch where the system chooses
$!A$	persistent rule or reusable condition

This explains why linear logic is attractive for game design. It tracks what is actually available after each action.

§1.5 Dialogue as a game

The second major theme of the PDF is Dr. Peng's comment that linear logic opened a "new world" because logic ascends to the level of interaction, or even to the level of a game. In game semantics:

$$\text{propositions are games,} \qquad \text{proofs are strategies.}$$

A proof of A is not merely a certificate that A is true. It is a strategy for the Proponent to win the game associated with A, no matter how the Opponent plays.

Definition 1.5.1 (Game-semantical slogan)**.** A formula specifies a protocol of interaction. A proof specifies a winning strategy in that protocol.

The cut rule has a dynamic meaning. If one proof produces A and another proof consumes A, cutting them together makes the two strategies interact. Cut elimination is the execution of that interaction until no internal communication remains.

$$\begin{gathered} \text{proof of } A \\ \Downarrow \\ \text{strategy producing moves of type } A \\ \Downarrow \\ \text{cut with a counter-strategy} \\ \Downarrow \\ \text{interaction / normalization / play} \end{gathered}$$

§1.6 Schopenhauer's stratagem as a ludic interaction

The PDF's second example uses a Schopenhauer-style debate. The Opponent says:

> “He is a child, so you must make allowance for him.”

The Proponent answers:

> “Precisely because he is a child, I must correct him; otherwise he will persist in bad habits.”

The content of the debate is less important than its structure. The Proponent accepts one premise and flips the other.

A code-like representation is:

```
O = {
    "P1": "he is a child",
    "P2": "one must make allowance for children",
    "claim": "do not correct him",
}

P = {
    "accept": "P1",
    "negate": "P2",
    "counterclaim": "because he is a child, correct him",
    "support": "otherwise he will persist in bad habits",
}

def retorsio_argumenti(opponent, proponent):
    assert proponent["accept"] == "P1"
    return {
        "kept": opponent["P1"],
        "reversed": opponent["P2"],
        "new_claim": proponent["counterclaim"],
    }
```

This is close to the Ludics interpretation. A debate is a tree of possible moves. A winning strategy is a path through that tree that answers every possible continuation by the opponent.

§1.7 Ludics: loci instead of signifieds

The PDF’s phrase “the signifier chain without the signifier” should not be read merely as literary theory. In Girard’s Ludics, formulas are replaced by loci: addresses at which interaction takes place.

Definition 1.7.1 (Locus, informal)**.** A locus is an address in a design. Instead of saying that a proof manipulates formulas with fixed semantic content, Ludics says that designs interact at addresses.

The meaning of a proposition is therefore not hidden behind the symbol as a static signified. Meaning is generated by interaction: by how a design responds to its dual.

This gives a cleaner version of the PDF’s Lacanian metaphor:

meaning is not stored inside a sign;
meaning appears through interaction with its negation.

§1.8 Orthogonality and negation

A central operation in Ludics and linear logic is orthogonality. Very roughly, two designs are orthogonal if their interaction terminates properly.

Definition 1.8.1 (Orthogonality, notebook version)**.** Write $D \perp E$ when the interaction between designs D and E converges successfully. Then the orthogonal of a collection A is

$$A^{\perp} = \{E : \forall D \in A,\ D \perp E\}.$$

A behavior is often recovered by biorthogonal closure:

$$A = (A^{\perp})^{\perp}.$$

This is the mathematically controlled form of the PDF's claim that a proposition is defined by its negation. One does not understand A by staring at A alone. One understands A through the class of all counter-designs with which it can interact.

§1.9 Geometry of interaction

Geometry of Interaction is the diagrammatic and dynamical side of the same idea. A proof is not a static derivation tree but a flow of information. The PDF's single most useful visual artifact is a colored morphism diagram from Melliès-style topologies.

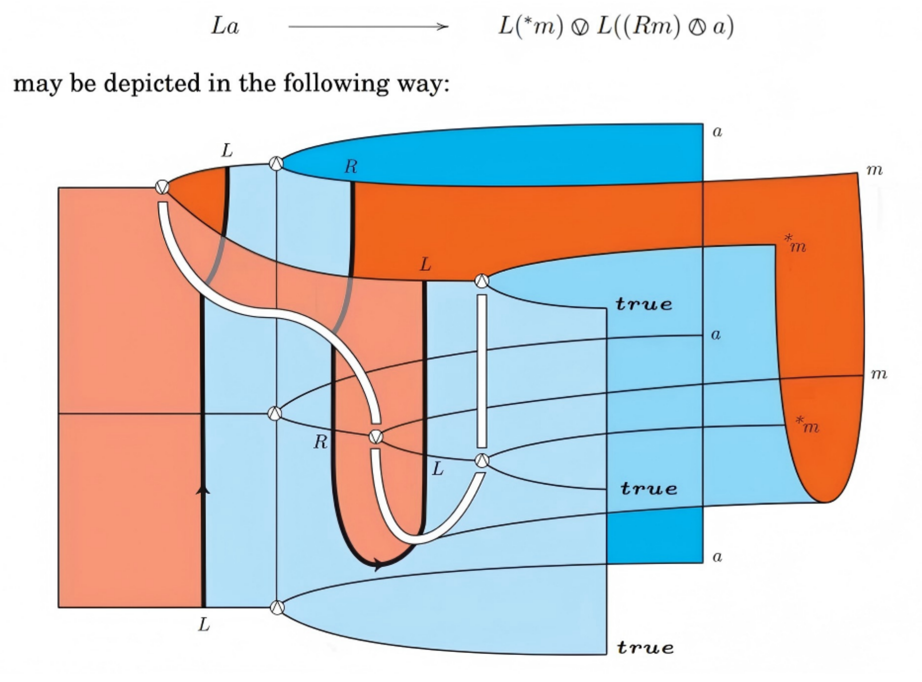

Figure 1.1: Geometry-of-Interaction morphism.

The visible morphism has the form

$$La \longrightarrow L({}^{*}m) \otimes L((Rm) \otimes a).$$

For a first reading, the notation is less important than the picture. The colored ribbons behave like wires. They route input types into output types, bend around each other, and meet at nodes. This is why the original PDF compares human interaction to Feynman diagrams: logical equivalence becomes a kind of topological deformation of processes.

Remark 1.9.1 — The diagram should not be treated as a decorative metaphor. In Geometry of Interaction, cut elimination is modeled by the circulation of information through a network. The graphical form is an attempt to make proof normalization visible.

§1.10 From narrative to social relations

The original PDF ends by extending the same logic to narrative generation and social relations. This is speculative, but it has a precise core. A social or narrative process can be modeled as a resource transformation system:

$$\text{inputs} \otimes \text{labor/process} \multimap \text{outputs}.$$

The important constraint is that resources do not duplicate for free. If a theory of narrative or society assumes arbitrary reuse of agents, time, attention, or material goods, it is silently using weakening and contraction. Linear logic forces these assumptions into the open.

A minimal production rule looks like:

```
production_rule = {
    "inputs":  ["labor", "wood", "tools"],
    "output":  "table",
    "logic":   "labor tensor wood tensor tools -o table",
}
```

The same pattern covers the fable rule

$$\texttt{pig} \otimes \texttt{bricks} \multimap \texttt{brick\_house}.$$

This is why the PDF's jump from the Three Little Pigs to social production is not arbitrary. Both are resource-sensitive narratives.

§1.11 The general pattern

The reconstructed message is simple:

$$\begin{gathered}\text{logic is not only truth classification}\\ \Downarrow\\ \text{a proof is a resource-sensitive process}\\ \Downarrow\\ \text{a process can be read as a game}\\ \Downarrow\\ \text{a game can be read as dialogue}\\ \Downarrow\\ \text{dialogue can be visualized as Geometry of Interaction.}\end{gathered}$$

The first version of the PDF used large philosophical language: signifier chains, narrative, game, topology, dialectics, social relations. The cleaned-up mathematical version keeps the ambition but anchors it in four concrete devices:

$$A \multimap B, \qquad A \otimes B, \qquad A^{\perp}, \qquad A = (A^{\perp})^{\perp}.$$

Linear logic is the logic of finite resources; Ludics is the logic of interaction at addresses; game semantics is the logic of strategies; Geometry of Interaction is the logic of information flow. Together they explain why a proof can look like a story, a debate, or a diagram of motion.