A Lambda Calculus for Gödel–Dummett Logic Capturing Waitfreedom

Yoichi Hirai

Univ. of Tokyo, JSPS research fellow

2012-05-23, FLOPS 2012, Kobe

Curry–Howard Correspondence

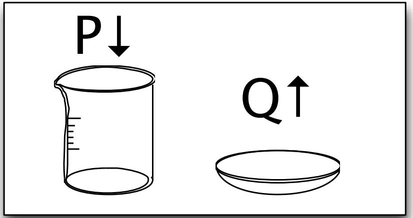
Logic	Computation
Intuitionistic propositional logic	Simply-typed λ each term has a unique normal form
$+ \quad (\varphi \to \psi) \vee (\psi \to \varphi)$ $= \quad$ Gödel–Dummett Logic	λ-GD Normal form not unique in general: utilize as distributed computation
$+ \quad \varphi \vee \neg\varphi$ $= \quad$ Classical propositional logic	many calculi Normal form not unique in general: fight with evaluation strategy

This correspondence is based on
Brouwer–Heyting–Kolmogorov interpretation

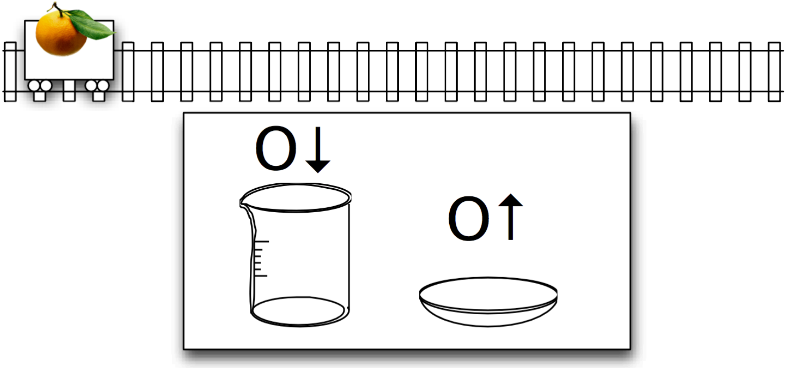
Brouwer–Heyting–Kolmogorov Interpretation

explains logical connectives in intuitionistic logic

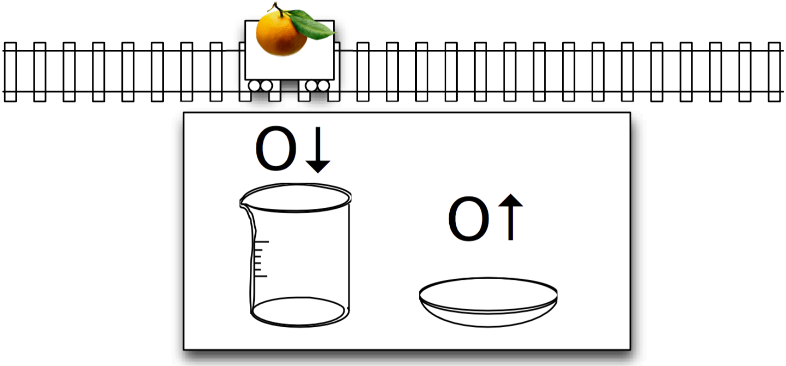
- A proof of $P \land Q$ is a pair $\langle x, y \rangle$
  where x is a proof of P and y is a proof of Q
- A proof of $P \lor Q$ is a pair $\langle i, x \rangle$
  where $i = 0$ and x is a proof of P, or $i = 1$ and x is a proof of Q
- A proof of $P \to Q$ is a construction which permits us to transform any proof of P into a proof of Q.



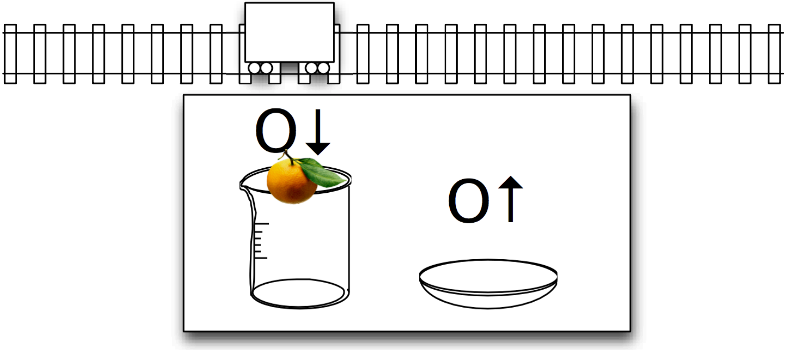
Computational Interpretation of Implication O $\rightarrow$ O



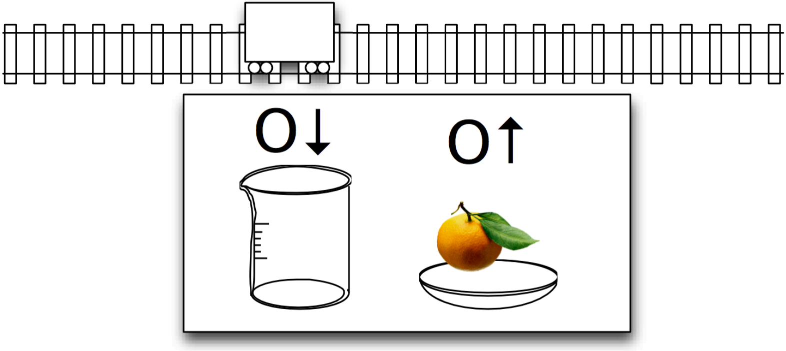
Computational Interpretation of Implication $O \to O$



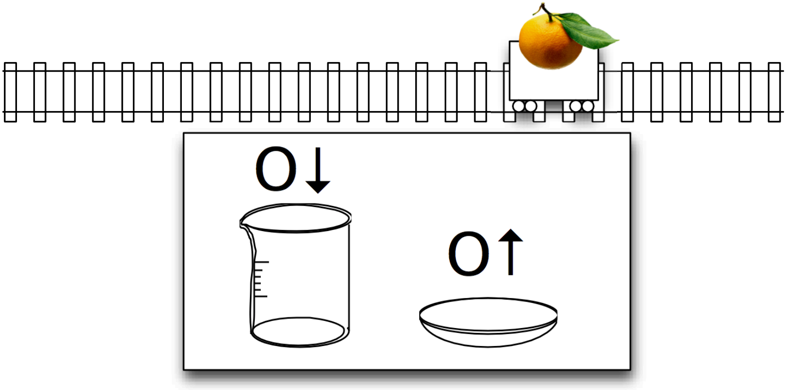
Computational Interpretation of Implication O → O



Computational Interpretation of Implication $O \rightarrow O$

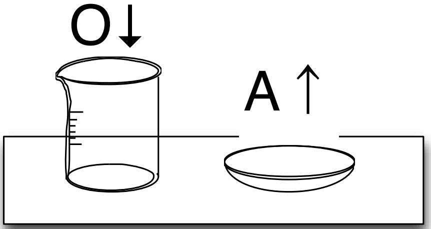


Computational Interpretation of Implication O → O

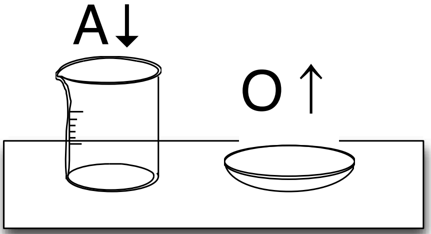


Computational Interpretation of Dummett Axiom $(O \to A) \vee (A \to O) \ldots ?$

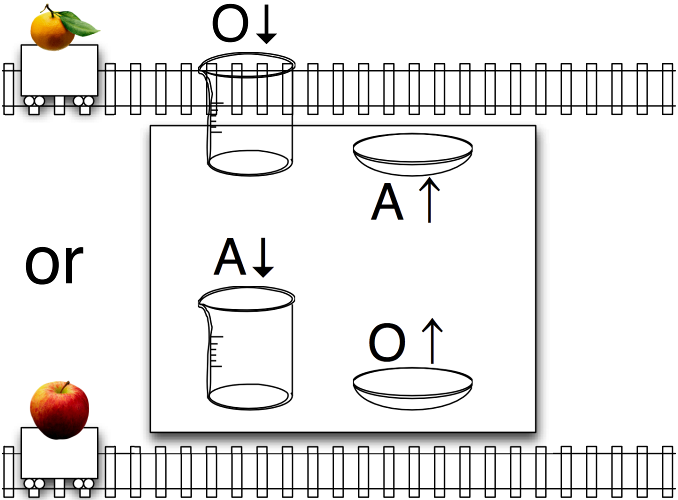
Gödel–Dummett logic has Dummett Axiom $(\varphi \to \psi) \vee (\psi \to \varphi)$, which must be realized for any φ and ψ.



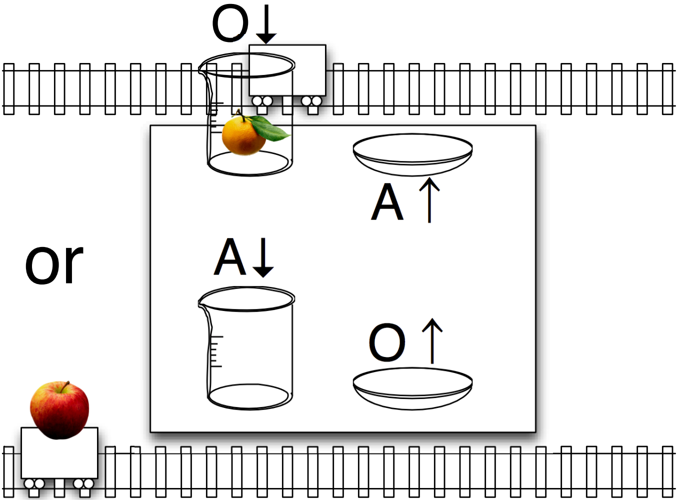
or



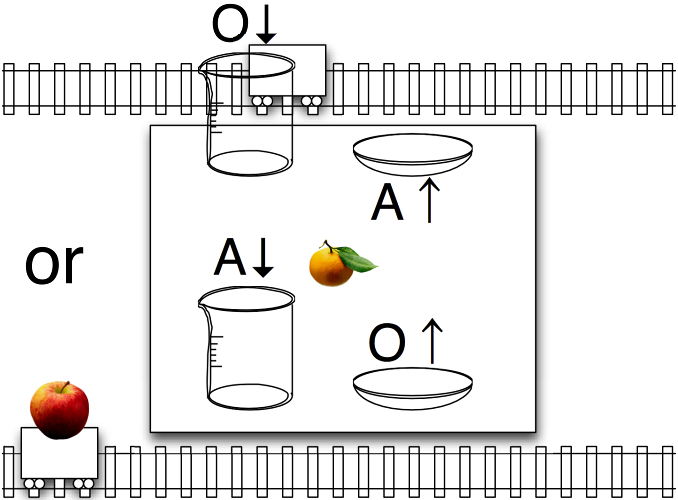
Computational Interpretation of Dummett Axiom $(O \to A) \vee (A \to O)$



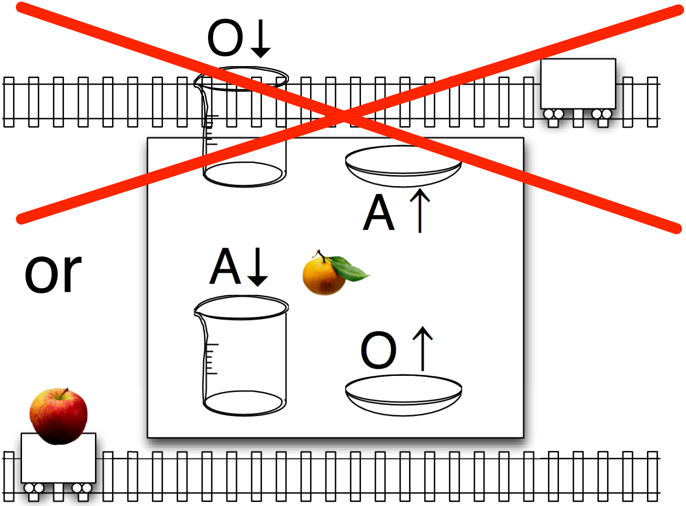
Computational Interpretation of Dummett Axiom $(O \to A) \vee (A \to O)$



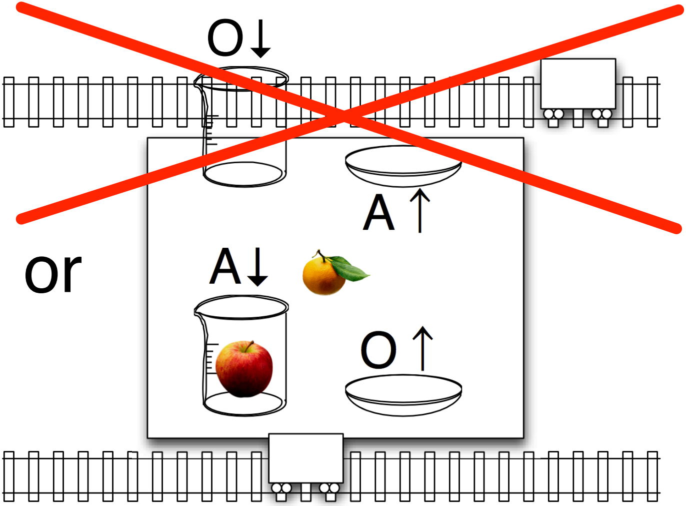
Computational Interpretation of Dummett Axiom $(O \to A) \vee (A \to O)$



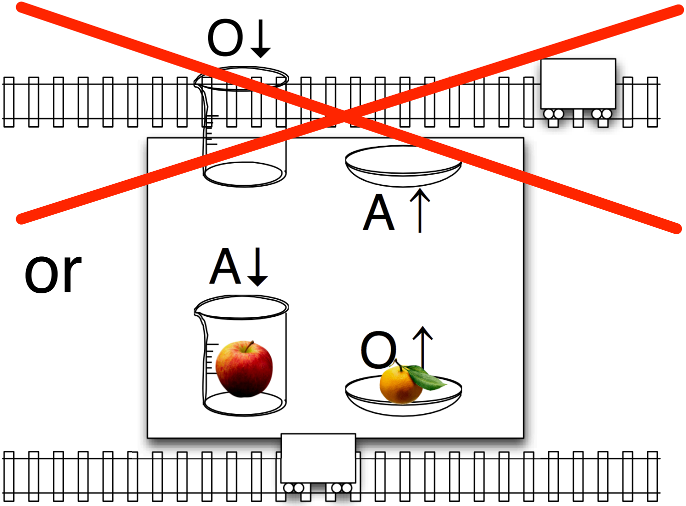
Computational Interpretation of Dummett Axiom $(O \to A) \vee (A \to O)$



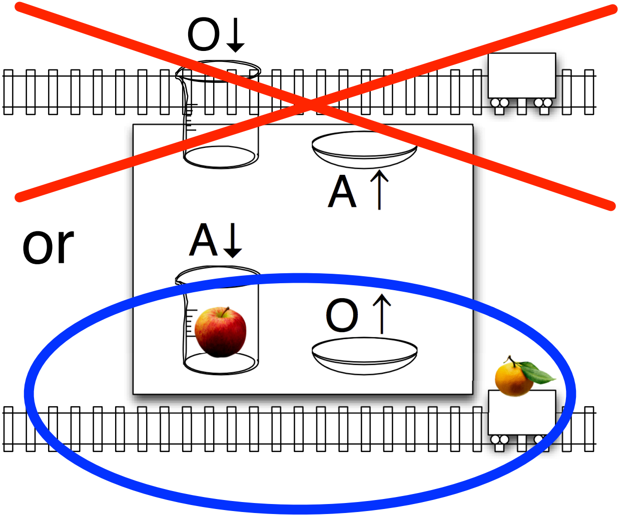
Computational Interpretation of Dummett Axiom $(O \to A) \vee (A \to O)$



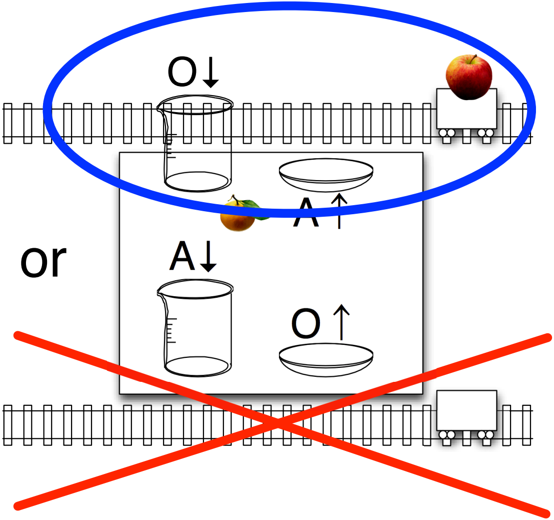
Computational Interpretation of Dummett Axiom
$(O \to A) \lor (A \to O)$



Computational Interpretation of Dummett Axiom $(O \to A) \lor (A \to O)$

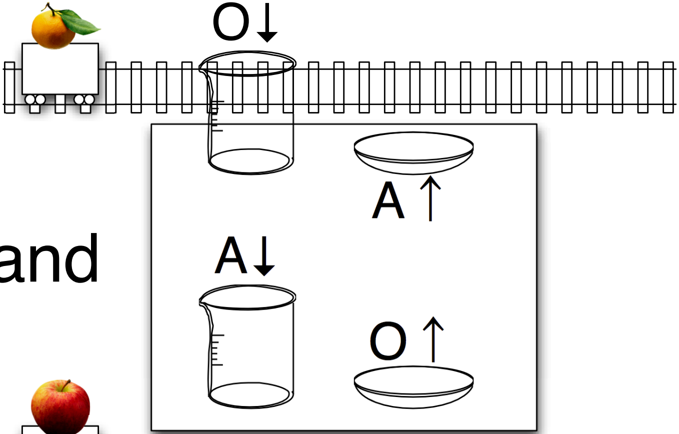


Computational Interpretation of Dummett Axiom $(O \to A) \lor (A \to O)$

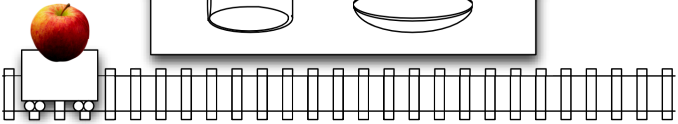


$(O \to A) \wedge (A \to O)$ is Not a Theorem of G.D. Logic

But, if rendez-vous was possible, the formula is realizable

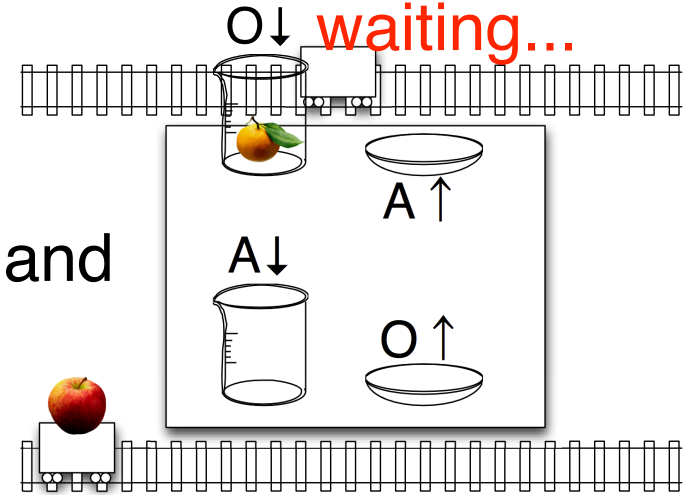


and



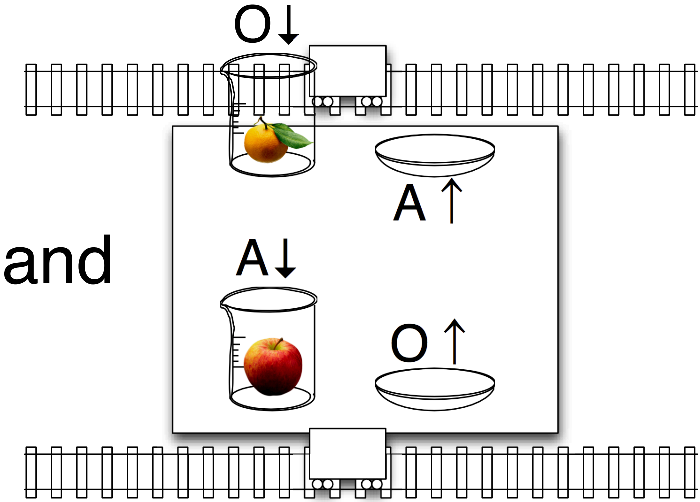
$(O \to A) \wedge (A \to O)$ is Not a Theorem of G.D. Logic

But, if rendez-vous was possible, the formula is realizable



$(O \rightarrow A) \wedge (A \rightarrow O)$ is Not a Theorem of G.D. Logic

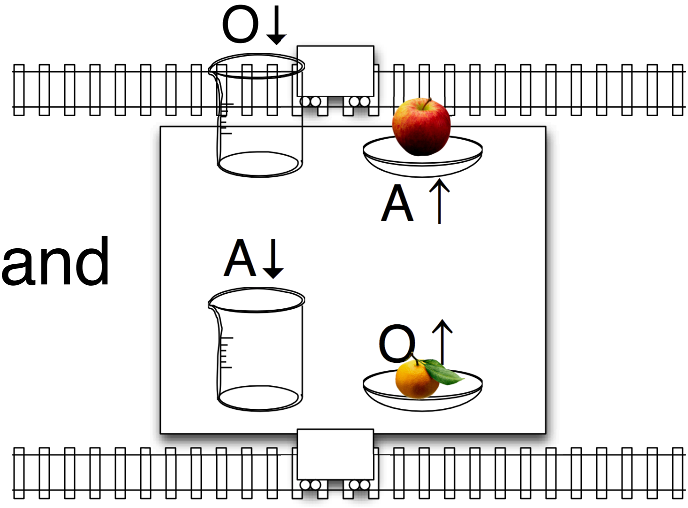
But, if rendez-vous was possible, the formula is realizable



$(O \to A) \wedge (A \to O)$ is Not a Theorem of G.D. Logic

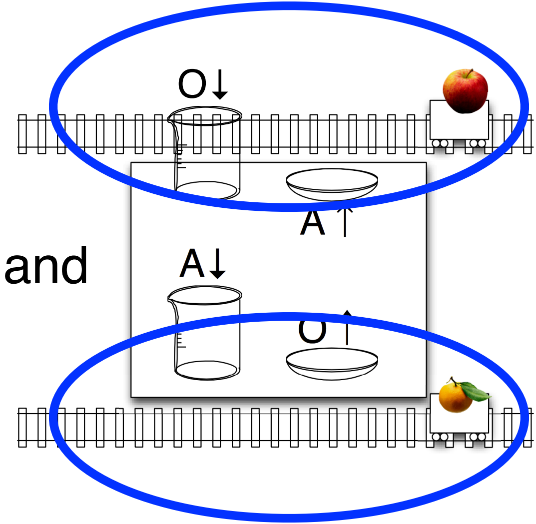
But, if rendez-vous was possible, the formula is realizable

and



$(O \to A) \wedge (A \to O)$ is Not a Theorem of G.D. Logic

But, if rendez-vous was possible, the formula is realizable



Waitfree = “impossible to wait for other processes”

(free as in tax-free and alcohol-free)
A task can be waitfreely solvable or not.

A task $\subseteq \mathbf{I}^{\mathbb{P}} \times \mathbf{O}^{\mathbb{P}}$ where $\mathbb{P}$ is processes, $\mathbf{I}$ is inputs and $\mathbf{O}$ is outputs

A watifreely unsolvable task: $\{((x, y), (y, x)) \mid x, y \in \{0, 1\}\}$

A waitfreely solvable task:
$\{((x, y), (x, x + y)), ((x, y), (x + y, y)), ((x, y), (x + y, x + y)) \mid x, y \in \{0, 1\}\}$

The definition of waitfreedom is long: involving a virtual machine
Saks and Zaharoglou (2000)

A task is

waitfreely solvable

$\Longleftrightarrow$

solvable by a typable λ-GD-term

For a typed lambda calculus λ-GDfor Gödel–Dummett logic.

Formulas, Sequents, Hypersequents

Local Types

$\varphi ::= \bot \mid \mathbf{P} \mid \varphi \to \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$

Global Types

$\varphi^+ ::= [\mathbf{i}]\varphi \mid \varphi^+ \wedge \varphi^+ \mid \varphi^+ \vee \varphi^+$
where **i** is a process

Sequent

$\mathcal{S} ::= \Gamma \vdash \varphi^+$ whereΓ is a finite set of global formulas

Hypersequent

$\mathcal{H} ::= \mathcal{S} \mid (\mathcal{S} \mathbin{\Big|} \mathcal{H})$

Hyper-Natural Deduction (Based on Avron's Hypersequents)

External Rules

$$\text{com'}\ \frac{\mathcal{H} \mid \Gamma, \Delta \vdash [i]\varphi \qquad \mathcal{H} \mid \Gamma, \Delta \vdash [j]\psi}{\mathcal{H} \mid \Gamma \vdash [i]\psi \mid \Delta \vdash [j]\varphi}$$ and structural rules

Inner Global Rules

$$\wedge\mathcal{E}_0\ \frac{\mathcal{H} \mid \Gamma \vdash \varphi^+ \wedge \psi^+}{\mathcal{H} \mid \Gamma \vdash \varphi^+} \qquad \wedge\mathcal{E}_1\ \frac{\mathcal{H} \mid \Gamma \vdash \varphi^+ \wedge \psi^+}{\mathcal{H} \mid \Gamma \vdash \psi^+}$$

and $\wedge\mathcal{I}, \vee\mathcal{E}, \vee\mathcal{I}$

Inner Local Rules

$$[i]\text{Ax}\ \frac{}{[i]\varphi, \Gamma \vdash [i]\varphi} \qquad [i] \to \mathcal{I}\ \frac{\mathcal{H} \mid [i]\varphi, \Gamma \vdash [i]\psi}{\mathcal{H} \mid \Gamma \vdash [i](\varphi \to \psi)}$$

$$[i] \to \mathcal{E}\ \frac{\mathcal{H} \mid \Gamma \vdash [i](\varphi \to \psi) \qquad \mathcal{H} \mid \Gamma \vdash [i]\varphi}{\mathcal{H} \mid \Gamma \vdash [i]\psi}$$

and $\bot\mathcal{E}, \wedge\mathcal{E}, \wedge\mathcal{I}, \vee\mathcal{E}, \vee\mathcal{I}$

Example Derivation of $[0](O \to A) \vee [1](A \to O)$

$$
\begin{prooftree}
\AxiomC{$[0]O, [1]A \vdash [0]O$}
\AxiomC{$[0]O, [1]A \vdash [1]A$}
\LeftLabel{com'}
\BinaryInfC{$[0]O \vdash [0]A \mid [1]A \vdash [1]O$}
\LeftLabel{$[i] \to \mathcal{I}$}
\doubleLine
\UnaryInfC{$\vdash [0](O \to A) \mid \vdash [1](A \to O)$}
\LeftLabel{$[i] \vee \mathcal{I}$}
\doubleLine
\UnaryInfC{$\vdash [0](O \to A) \vee [1](A \to O) \mid \vdash [0](O \to A) \vee [1](A \to O)$}
\LeftLabel{EC}
\UnaryInfC{$\vdash [0](O \to A) \vee [1](A \to O)$}
\end{prooftree}
$$

The conclusion

- as a type, represents the orange–apple protocol
- as a formula, without modalities [i], is the Dummett axiom.

Typing Terms

External Rules

$$\frac{\mathcal{G}_0 \mid \Gamma, \Delta \rhd M : [i]\varphi \qquad \mathcal{G}_1 \mid \Gamma, \Delta \rhd N : [j]\psi}{[\mathcal{G}_0, \mathcal{G}_1] \mid \Gamma \rhd \overrightarrow{\ell^{i}_{\Delta}}(M) : [i]\psi \mid \Delta \rhd \overleftarrow{\ell^{j}_{\Gamma}}(N) : [j]\varphi}$$

Inner Global Rules

$$\frac{\mathcal{G} \mid \Gamma \rhd M : \varphi^+ \wedge \psi^+}{\mathcal{G} \mid \Gamma \rhd \pi^{g}_{l}(M) : \varphi^+} \qquad \frac{\mathcal{G} \mid \Gamma \rhd M : \varphi^+ \wedge \psi^+}{\mathcal{G} \mid \Gamma \rhd \pi^{g}_{r}(M) : \psi^+}$$

Inner Local Rules

$$\frac{}{x : [i]\varphi, \Gamma \rhd x : [i]\varphi} \qquad \frac{\mathcal{G} \mid x : [i]\varphi, \Gamma \rhd M : [i]\psi}{\mathcal{G} \mid \Gamma \rhd \lambda x.M : [i](\varphi \to \psi)}$$

$$\frac{\mathcal{G}_0 \mid \Gamma \rhd M : [i](\varphi \to \psi) \qquad \mathcal{G}_1 \mid \Gamma \rhd N : [i]\varphi}{[\mathcal{G}_0, \mathcal{G}_1] \mid \Gamma \rhd MN : [i]\psi}$$

where $\mathcal{G}_0$ and $\mathcal{G}_1$ has the same types.

Example Typed Term

$$
\begin{array}{c}
\mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \mathsf{x}\colon [0]\mathsf{O} \qquad \mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \mathsf{y}\colon [1]\mathsf{A} \\
\hline
\mathsf{x}\colon [0]\mathsf{O} \rhd \overrightarrow{\ell}\,(\mathsf{x})\colon [0]\mathsf{A} \mid \mathsf{y}\colon [1]\mathsf{A} \rhd \overleftarrow{\ell}\,(\mathsf{y})\colon [1]\mathsf{O} \\
\hline\hline
\mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \overrightarrow{\ell}\,(\mathsf{x})\colon [0]\mathsf{A} \mid \mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \overleftarrow{\ell}\,(\mathsf{y})\colon [1]\mathsf{O} \\
\hline\hline
\mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right)\colon \varphi \mid \mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)\colon \varphi \\
\hline
\mathsf{x}\colon [0]\mathsf{O}, \mathsf{y}\colon [1]\mathsf{A} \rhd \left[\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)\right]\colon (\varphi :=)[0]\mathsf{A} \vee [1]\mathsf{O}
\end{array}
$$

Example Reduction 1

From the previously typed term:
write reduction

$$(\{\qquad\}, \{\qquad\}, [\mathbf{inl^g}\left(\overrightarrow{\ell}\,(\mathbf{x})\right), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)])$$
$$\rightsquigarrow(\{\ell \mapsto \mathbf{x}\}, \{\qquad\}, [\mathbf{inl^g}\left(\overrightarrow{\ell}\,(\mathbf{x})\right), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)])$$

Example Reduction 1

From the previously typed term:
read reduction

$$
\begin{aligned}
&(\{\qquad\quad\}, \{\qquad\quad\}, [\mathbf{inl^g}\left(\overrightarrow{\ell}\,(\mathbf{x})\right), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)]) \\
\rightsquigarrow &(\{\ell \mapsto \mathbf{x}\}, \{\qquad\quad\}, [\mathbf{inl^g}\left(\overrightarrow{\ell}\,(\mathbf{x})\right), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)]) \\
\rightsquigarrow &(\{\ell \mapsto \mathbf{x}\}, \{\qquad\quad\}, [\mathbf{inl^g}\,(\mathbf{abort}\,), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)])
\end{aligned}
$$

Example Reduction 1

From the previously typed term:
write reduction

$$
\begin{aligned}
&(\{\qquad\}, \{\qquad\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\qquad\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\qquad\}, [\mathsf{inl^g}\,(\mathsf{abort}), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl^g}\,(\mathsf{abort}), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])
\end{aligned}
$$

Example Reduction 1

From the previously typed term:
read reduction

$$
\begin{aligned}
&(\{\qquad\quad\}, \{\qquad\quad\}, [\mathbf{inl^g}\left(\overrightarrow{\ell}\,(\mathbf{x})\right), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathbf{x}\}, \{\qquad\quad\}, [\mathbf{inl^g}\left(\overrightarrow{\ell}\,(\mathbf{x})\right), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathbf{x}\}, \{\qquad\quad\}, [\mathbf{inl^g}\,(\mathbf{abort}\,), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathbf{x}\}, \{\ell \mapsto \mathbf{y}\}, [\mathbf{inl^g}\,(\mathbf{abort}\,), \mathbf{inr^g}\left(\overleftarrow{\ell}\,(\mathbf{y})\right)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathbf{x}\}, \{\ell \mapsto \mathbf{y}\}, [\mathbf{inl^g}\,(\mathbf{abort}\,), \mathbf{inr^g}\,(\mathbf{x})])
\end{aligned}
$$

Example Reduction 1

From the previously typed term:
abort propagation

$$
\begin{aligned}
&(\{\quad\quad\}, \{\quad\quad\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\quad\quad\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\quad\quad\}, [\mathsf{inl^g}\,(\mathsf{abort}\,), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl^g}\,(\mathsf{abort}\,), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl^g}\,(\mathsf{abort}\,), \mathsf{inr^g}\,(\mathsf{x})])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{abort}\,, \mathsf{inr^g}\,(\mathsf{x})])
\end{aligned}
$$

Example Reduction 1

From the previously typed term:
abort propagation

$$
\begin{aligned}
&(\{\quad\quad\},\{\quad\quad\},[\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right),\mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell\mapsto\mathsf{x}\},\{\quad\quad\},[\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right),\mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell\mapsto\mathsf{x}\},\{\quad\quad\},[\mathsf{inl}^{\mathsf{g}}\,(\mathsf{abort}\,),\mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell\mapsto\mathsf{x}\},\{\ell\mapsto\mathsf{y}\},[\mathsf{inl}^{\mathsf{g}}\,(\mathsf{abort}\,),\mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell\mapsto\mathsf{x}\},\{\ell\mapsto\mathsf{y}\},[\mathsf{inl}^{\mathsf{g}}\,(\mathsf{abort}\,),\mathsf{inr}^{\mathsf{g}}\,(\mathsf{x})])\\
\rightsquigarrow&(\{\ell\mapsto\mathsf{x}\},\{\ell\mapsto\mathsf{y}\},[\mathsf{abort}\,,\mathsf{inr}^{\mathsf{g}}\,(\mathsf{x})])\\
\rightsquigarrow&(\{\ell\mapsto\mathsf{x}\},\{\ell\mapsto\mathsf{y}\},\mathsf{inr}^{\mathsf{g}}\,(\mathsf{x}))
\end{aligned}
$$

Example Reduction 2

From the same term:

$$
\begin{aligned}
&(\{\qquad\}, \{\qquad\}, [\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)]) \\
\rightsquigarrow&(\{\qquad\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr}^{\mathsf{g}}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)]) \\
\rightsquigarrow&(\{\qquad\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr}^{\mathsf{g}}\,(\mathsf{abort}\,)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl}^{\mathsf{g}}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr}^{\mathsf{g}}\,(\mathsf{abort}\,)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl}^{\mathsf{g}}\,(\mathsf{y}), \mathsf{inr}^{\mathsf{g}}\,(\mathsf{abort}\,)]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl}^{\mathsf{g}}\,(\mathsf{y}), \mathsf{abort}\,]) \\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, \mathsf{inl}^{\mathsf{g}}\,(\mathsf{y}))
\end{aligned}
$$

Example Reduction 3

Still from the same term:

$$
\begin{aligned}
&(\{\quad\}, \{\quad\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\quad\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl^g}\left(\overrightarrow{\ell}\,(\mathsf{x})\right), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl^g}\,(\mathsf{y}), \mathsf{inr^g}\left(\overleftarrow{\ell}\,(\mathsf{y})\right)])\\
\rightsquigarrow&(\{\ell \mapsto \mathsf{x}\}, \{\ell \mapsto \mathsf{y}\}, [\mathsf{inl^g}\,(\mathsf{y}), \mathsf{inr^g}\,(\mathsf{x})])
\end{aligned}
$$

Results

Strong normalization No typed term can reduce infinitely often

Non-abortfullness No typed term can reduce into **abort**

Waitfree characterization

A task is waitfreely solvable $\Leftrightarrow$ solvable by a typed term

History

Lamport (1977) introduced what is now called waitfree computing

Avron (1991) introduced a hypersequent calculus for Gödel–Dummett logic and showed cut-elimination

Gafni and Koutsoupias (1999)
showed that it is undecidable whether a task can be solved waitfreely

Herlihy and Shavit (1999), Saks and Zaharoglou (2000)
gave a topological characterization of waitfree computation (Gödel Prize 2004)

I found waitfreedom when I was looking at past Gödel Prizes

Avron's Question and My Answer

Avron (1991): it seems to us extremely important to determine the exact computational content of [the intermediate logics with cut-elimination for hypersequents] — and to develop corresponding "λ-calculi."

I answer:

- The computational content of Gödel–Dummett logic is waitfreedom
- λ-GD is the λ-calculus for it

Fermüller (2003) gave a natural deduction but not reductions

Future Work:

- generalizing to more logics and investigating classical logic
- adapting to weaker shared memory consistency