Resources, process calculi and Gödel-Dummett logics

Dominique Larchey

LORIA – CNRS

Nancy, France

Gödel-Dummett logics LC

- Most studied intermediate logic $\mathsf{IL} \subset \mathsf{LC} \subset \mathsf{CL}$
- Proof theory, proof-search
  - IL (Dyckhoff & Hudelmair, Weich, Larchey & Galmiche)
  - Intermediate logics (Avellone et al. and Fiorino)
  - LC (Dyckhoff, Avron, Larchey)
- Calculi
  - Hyper-sequent calculi (Avron,Baaz, Fermüller)
  - Sequent calculus (Dyckhoff)
  - Counter-model search (CADE'02)
  - Decision through graph/matrix computation (IJCAR'04)

Process calculi, resource consumption

- Specification logics for resources
  - Production/consumption (linear logics, petri nets)
  - Distribution/sharing (spatial logics, BI)
  - Concurrency/mobility (ambients)
- LC not a general resource specification logic
- Characterization of LC w.r.t. a class of resource properties
  - Recursive non-deterministic processes, conditional branching
  - Boundability of resource consumption / counter-models in LC

A simple resource calculus

- Only one resource denoted $\bullet$
- Resource is only consumed, not produced
- Processes features
  - Non determinism $[P + Q + R]$
  - (External) conditional branching $<f> P$
  - Resource consumption $\bullet P$
  - Recursion through equations $X = [\cdots + \bullet X + \cdots]$
- X can consume one $\bullet$ and then becomes X again

Resource consumption semantics

- Define a relation $P \dashv\!\![n]\!\!\vdash\!\bullet\; Q$
  - means P can become Q after having consumed n times $\bullet$
- Resource consumption for equations system $\mathcal{E}$ and context σ

$$\frac{}{P \dashv\!\![0]\!\!\vdash\!\bullet\; P}\ [\mathrm{Id}]$$

$$\frac{P \dashv\!\![n]\!\!\vdash\!\bullet\; Q \qquad X = P \in \mathcal{E}}{X \dashv\!\![n]\!\!\vdash\!\bullet\; Q}\ [\mathrm{Eq}]$$

$$\frac{P_i \dashv\!\![n]\!\!\vdash\!\bullet\; Q}{[\cdots + P_i + \cdots] \dashv\!\![n]\!\!\vdash\!\bullet\; Q}\ [\mathrm{Sum}]$$

$$\frac{P \dashv\!\![n]\!\!\vdash\!\bullet\; Q \qquad [\![f]\!]_\sigma = 1}{<\!f\!>P \dashv\!\![n]\!\!\vdash\!\bullet\; Q}\ [\mathrm{Con}]$$

$$\frac{P \dashv\!\![n]\!\!\vdash\!\bullet\; Q}{\bullet\, P \dashv\!\![n+1]\!\!\vdash\!\bullet\; Q}\ [\mathrm{Res}]$$

Bounding resources consumption

- Basic idea
  - Is n in $P \dashv\!\![n]\!\!\vdash\!\bullet\, Q$ bounded for any P, Q ?
  - $X = \bullet X$ has unboundable resource use
  - $X = [\,]$ or $X = [X]$ have resource use bounded by 0
- Definition
  - Given a fixed process equation system $\mathcal{E}$
  - $\mathcal{E}$ has resource use boundable by n if

there exists a context σ s.t. for any process variables X, Y of $\mathcal{E}$, $X \dashv\!\![k]\!\!\vdash\!\bullet\, Y$ holds for no k greater than n

Normalized processes vs bi-colored graphs (1)

- Process system is normalized if no nested constructs
- Normalization = add (new) intermediate equations

$$X = \bullet\bullet X \qquad \text{split into} \qquad X = \bullet Y,\ Y = \bullet X$$

- Normalization achieved while preserving resource consumption
- Associated bi-colored conditional graph
  - $X = [Y_1 + Y_2]$: (unconditional) green arrows $X \longrightarrow Y_1$, $X \longrightarrow Y_2$
  - $X = <f> Y$: conditional green arrow $X \longrightarrow_f Y$
  - $X = \bullet Y$: red arrow $X \Rightarrow Y$

Normalized processes vs bi-colored graphs (2)

- Normalized process equation system

$$
\begin{array}{lllll}
X = [K_6] & Z = [K_5] & K_2 = <b> X & K_4 = \bullet\, Z & K_1 = [K_2 + K_7 + Z] \\
Y = [K_1] & K_6 = \bullet\, K_3 & K_3 = <a> Y & K_5 = [\;] & K_7 = <\neg a> K_4
\end{array}
$$

- Associated conditional bi-colored graph

$$
\begin{tikzcd}
X \arrow[d] & K_2 \arrow[l, "b"'] & K_1 \arrow[l] \arrow[r] \arrow[drr] & K_7 \arrow[r, "\overline{a}"] & K_4 \arrow[d, Rightarrow] \\
K_6 \arrow[r, Rightarrow] & K_3 \arrow[r, "a"] & Y \arrow[u] & K_5 & Z \arrow[l]
\end{tikzcd}
$$

- Size of graph linear in the size of process system

Instance graphs

- *Conditional bi-colored graphs* contains
  - green arrows: conditional $\longrightarrow_f$, unconditional $\longrightarrow$
  - red arrows: $\Rightarrow$
- Conditions can be instanciated with a context σ
  - keep unconditional arrows ($\longrightarrow$ or $\Rightarrow$)
  - keep arrows $\longrightarrow_f$ for which $[\![f]\!]_\sigma = 1$
- From a graph $\mathcal{G}$, we obtain a family of *instance graphs* $\mathcal{G}_\sigma$

Alternating chains and resource use bounding

- A *n-alternating chain*
  - of the form $(\longrightarrow^{\star}\Longrightarrow)^n$, exactly n red arrows $\Rightarrow$
- Resource use for a system $\mathcal{E}$ with associated graph $\mathcal{G}$

$$\boxed{\begin{array}{l}\mathcal{E} \text{ has resource use boundable by } n \text{ if and only if } \mathcal{G} \text{ has} \\ \text{an instance } \mathcal{G}_\sigma \text{ containing no } (n+1)\text{-alternating chain}\end{array}}$$

$$\begin{tikzcd}
X \arrow[d] & K_2 \arrow[l, "b"'] & K_1 \arrow[l] \arrow[r] \arrow[drr] & K_7 \arrow[r, "\overline{a}"] & K_4 \arrow[d, Rightarrow] \\
K_6 \arrow[r, Rightarrow] & K_3 \arrow[r, "a"] & Y \arrow[u] & K_5 & Z \arrow[l]
\end{tikzcd}
\quad - \begin{bmatrix} a = 0 \\ b = 0 \end{bmatrix} \longrightarrow \quad
\begin{tikzcd}
X \arrow[d] & K_2 & K_1 \arrow[l] \arrow[r] \arrow[drr] & K_7 \arrow[r] & K_4 \arrow[d, Rightarrow] \\
K_6 \arrow[r, Rightarrow] & K_3 & Y \arrow[u] & K_5 & Z \arrow[l]
\end{tikzcd}$$

- Graphs and process equations are equivalent representations

Gödel-Dummett logic LC

- Intermediate logic: $\boxed{\mathsf{IL} \subset \mathsf{LC} \subset \cdots \subset \mathsf{LC}_n \subset \cdots \subset \mathsf{LC}_1 = \mathsf{CL}}$
- Syntactic characterization: $\mathsf{LC} = \mathsf{IL} + (X \supset Y) \vee (Y \supset X)$
- Semantic models:
  - Linear Kripke trees or the lattice $\overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$
  - For finitary LC_n, $\overline{[0, n)} = [0, \ldots, n[\cup \{\infty\}$
  - Lattice structure: min, max, . . .
- Complexity:
  - LC (and CL) are NP-complete
  - IL is PSPACE-complete

Instance graphs and implicational sequents

- Implicational sequent associated to a bi-colored (instance) graph
  - for an arrow $X \rightarrow Y$: add $X \supset Y$ on the left of $\vdash$
  - for an arrow $X \Rightarrow Y$: add $Y \supset X$ (reverse) on the right of $\vdash$

$$\begin{tikzcd} B \arrow[r] & C \\ A \arrow[u] \arrow[ur, Rightarrow] & D \arrow[u] \end{tikzcd} \qquad A \supset B, B \supset C, D \supset C \vdash C \supset A$$

- A counter-model in LC_n iff no $(n+1)$-alternating chain in graph
- If no $(n+1)$-alternating chain, draw graph with $(n+1)$ levels :
  - Red arrows $\Rightarrow$ go up (strictly)
  - Green arrows $\rightarrow$ never go down
- $\boxed{\text{Counter model given by the level}}$ $(A = D = 0$ and $B = C = 1)$

Conditional bi-colored graphs to **LC**

- Build a sequent $\Gamma \vdash \Delta$ from a graph $\mathcal{G}$
  - $v \rightarrow w$ in $\mathcal{G}$: add $X_v \supset Y_w$ to Γ;
  - $v \rightarrow_f w$: add $(\neg\neg f) \supset (X_v \supset X_w)$ to Γ;
  - $v \Rightarrow w$: add $X_w \supset X_v$ to Δ.
- Counter-model of built sequent $\Gamma \vdash \Delta$?

$\Gamma \vdash \Delta$ has a counter-model in LC_n if and only if $\mathcal{G}$ has an instance $\mathcal{G}_\sigma$ containing no $(n+1)$-alternating chain

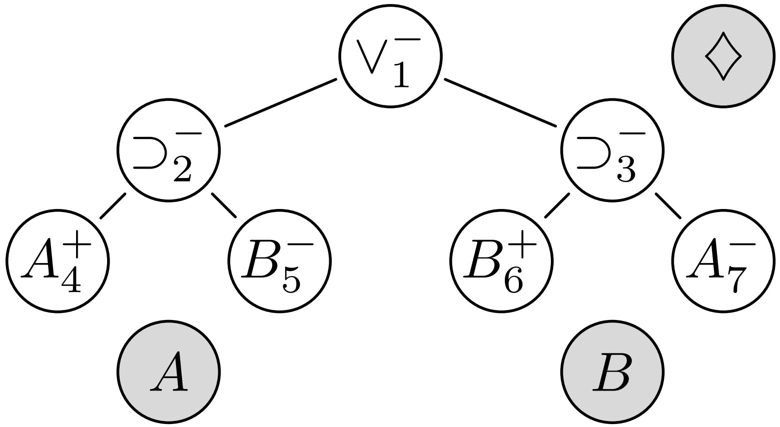
- Size of $\Gamma \vdash \Delta$ linear in size of $\mathcal{G}$

Resource bounding with the **LC** logic

- Resource bounding problem for process equations
- Linear transformation from process equations to a graph
- Linear transformation from graph to formula (or sequent) of LC
- Resource bounding problem linearly transformed into a decision (counter-model) problem for LC_n
- Is there a reverse transformation ?
- Yes, by our previous results (IJCAR'04)

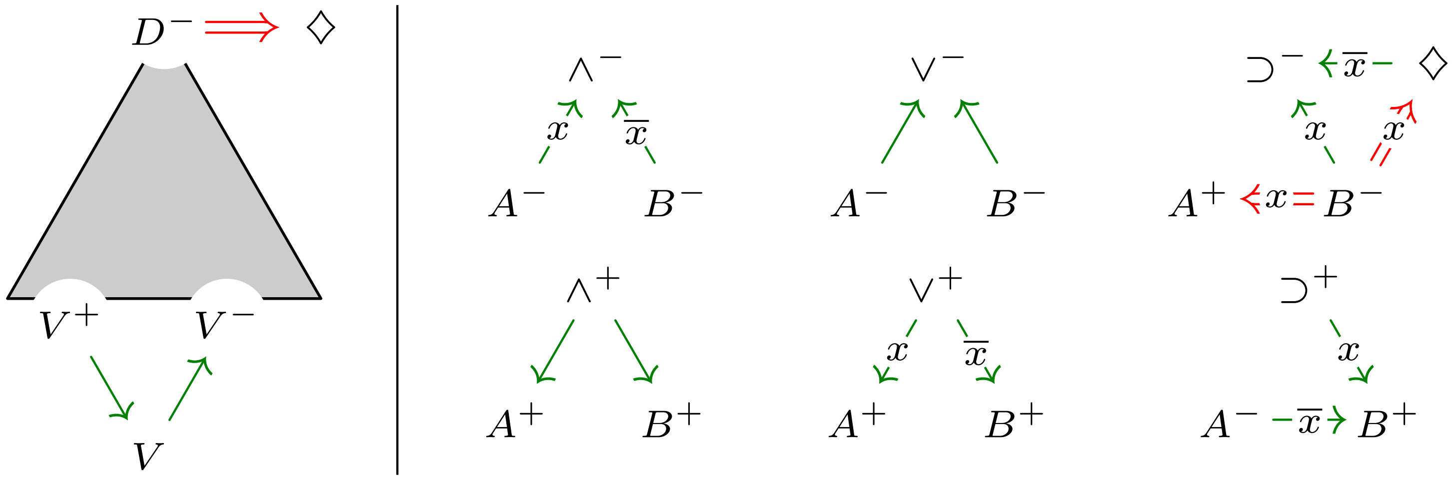
From **LC** to graphs: indexing and polarizing

- Build a graph based on sub-formulae tree
- Each occurrence of a sub-formulae has an **index** and a **polarity**
- Example: $(A \supset B) \vee (B \supset A)$



- Add one node $\diamondsuit$
- Add one further node for each variable V

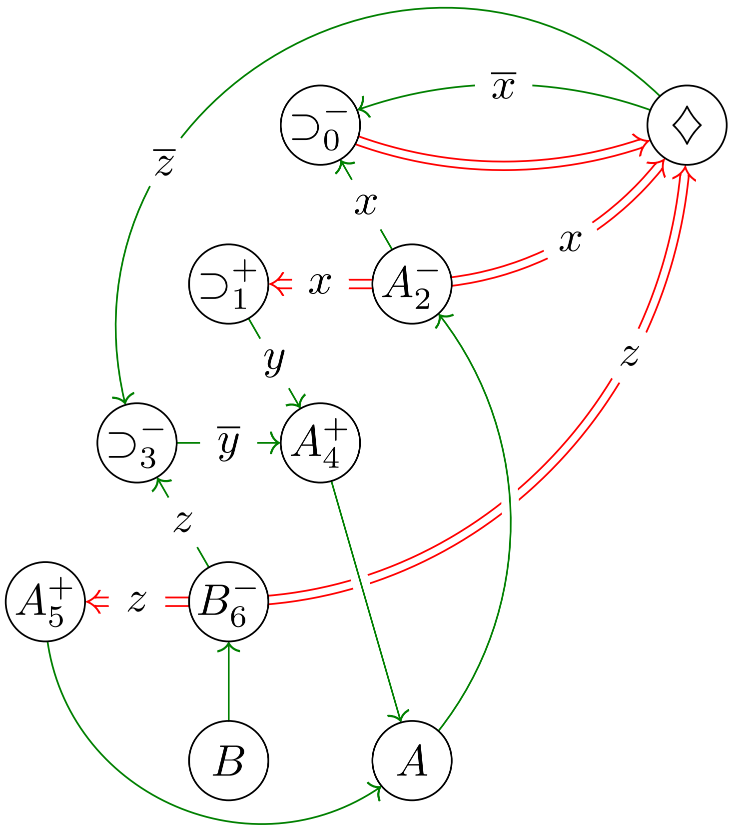
From $\mathbf{LC}_n$ to graphs: building arrows



- Given D^-, compute its graph $\mathcal{G}$ with boolean selectors
- $\mathcal{G}$ has size linear in the size of the formula D

D has counter-model in LC_n iff there exists an instance graph $\mathcal{G}_\sigma$ with no $(n+1)$-alternating chain

Example: Peirce's formula (1)



$$((A_5^+ \supset_3^- B_6^-) \supset_1^+ A_4^+) \supset_0^- A_2^-$$

- Build the bi-colored graph
- break cycles containing $\Rightarrow$

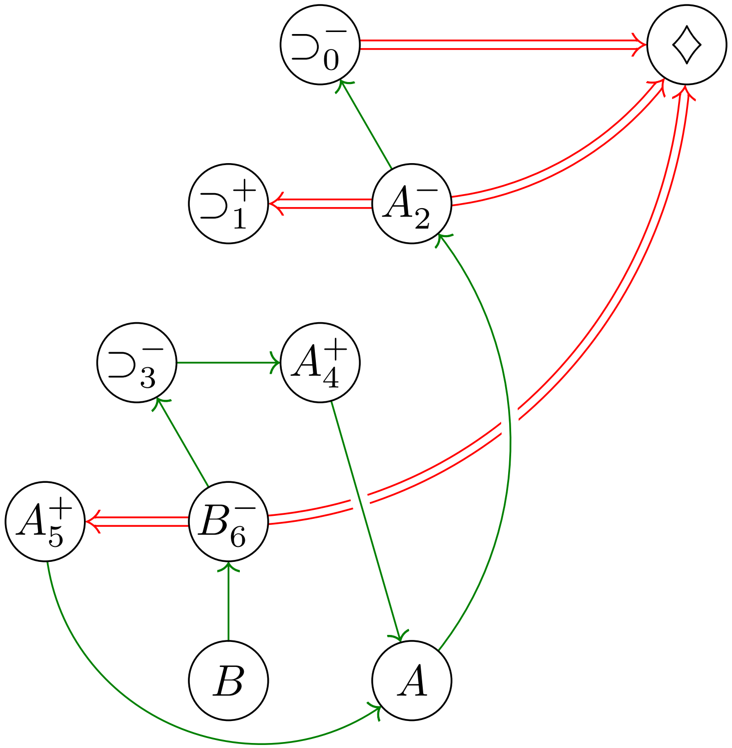
$0 \Rightarrow \Diamond \rightarrow 0$	x
$0 \Rightarrow \Diamond \rightarrow 3 \rightarrow 4 \rightarrow A \rightarrow 2 \rightarrow 0$	$z + y + \overline{x}$
$2 \Rightarrow \Diamond \rightarrow 3 \rightarrow 4 \rightarrow A \rightarrow 2$	$\overline{x} + z + y$
$2 \Rightarrow 1 \rightarrow 4 \rightarrow A \rightarrow 2$	$\overline{x} + \overline{y}$

- Constraint for no $\Rightarrow$-cycle:

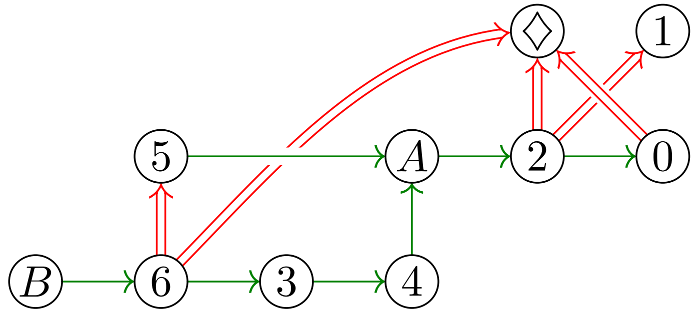
$$x.(z + y + \overline{x}).(\overline{x} + z + y).(\overline{x} + \overline{y})$$

- Solution: $x = z = 1$ and $y = 0$

Example: instance graph (2)



- Intanciate with $x = z = 1$, $y = 0$
- Of course, there is no $\Rightarrow$-cycle!
- Redraw the graph by levels:



- Counter-model: $[\![A]\!] = 1$, $[\![B]\!] = 0$
- In LC and LC_2 but not in CL $= LC_1$

From LC_n to resource bounding

- Linear transformation of a formula of LC into conditional graph
- Linear transformation of graph into a process system
- Decision problem for LC_n transformed into resource bounding problem (of linear size)
- Deciding LC_n characterized by resource bounding
- How to decide LC_n in practice ?

$\Rightarrow$-cycle detection and matrix computation (1)

- Bi-colored graph = sparse **matrices of boolean functions**

$\rightarrow$	0	1	2	3	4	5	6	A	B	$\diamondsuit$
0										
1					y					
2	x									
3					$\overline{y}$					
4								1		
5								1		
6				z						
A			1							
B							1			
$\diamondsuit$	$\overline{x}$			$\overline{z}$						

$\Rightarrow$	0	1	2	3	4	5	6	A	B	$\diamondsuit$
0										1
1										
2		x								x
3										
4										
5										
6						z				z
A										
B										
$\diamondsuit$										

- Shared ROBDDs, and matrix operations:
  - $+ = \vee$, $\times = \wedge$, $M^\star = I + M^2 + M^3 + \cdots$
  - $\mathrm{tr}(M) = \sum_x M_{x,x}$, $\sum(M) = \sum_{x,y} M_{x,y}$

$\Rightarrow$-cycle detection and matrix computation (2)

- Result: provable in LC iff $\boxed{\mathrm{tr}\big((\rightarrow + \Rightarrow)^{\star}\Rightarrow\big) = 1}$
- If the trace not a tautology:
  - Extract an instance with no $\Rightarrow$-cycle
  - Draw the instance by levels
  - Counter-model given by level
- Result: invalid in LC_n iff $\boxed{\sum(\rightarrow^{\star}\Rightarrow)^{n+1} < 1}$
- Compute this sequence:
  - n bounded by number of $\supset^{-}$+1
  - Search the first non-tautology
  - Obtain the minimal counter-model

Conclusion and perspectives

- A process calculus with resource consumption
- Decision problem in LC_n = resource consumption bounding
- Solved through matrix computation with BDDs

`http://www.loria.fr/~larchey/LC`

- Use this process calculus is an abstraction calculus to effectively bound resource use in more complex systems