Reductive Logic, Proof-search, and Coalgebra

Alexander V. Gheorghiu David J. Pym

Department of Computer Science
University College London

CMCS
2022

Deductive Logic vs. Reductive Logic

$$\frac{\text{Established Premiss}_1 \quad \ldots \quad \text{Established Premiss}_n}{\text{Established Conclusion}} \Downarrow$$

Deductive Logic vs. Reductive Logic

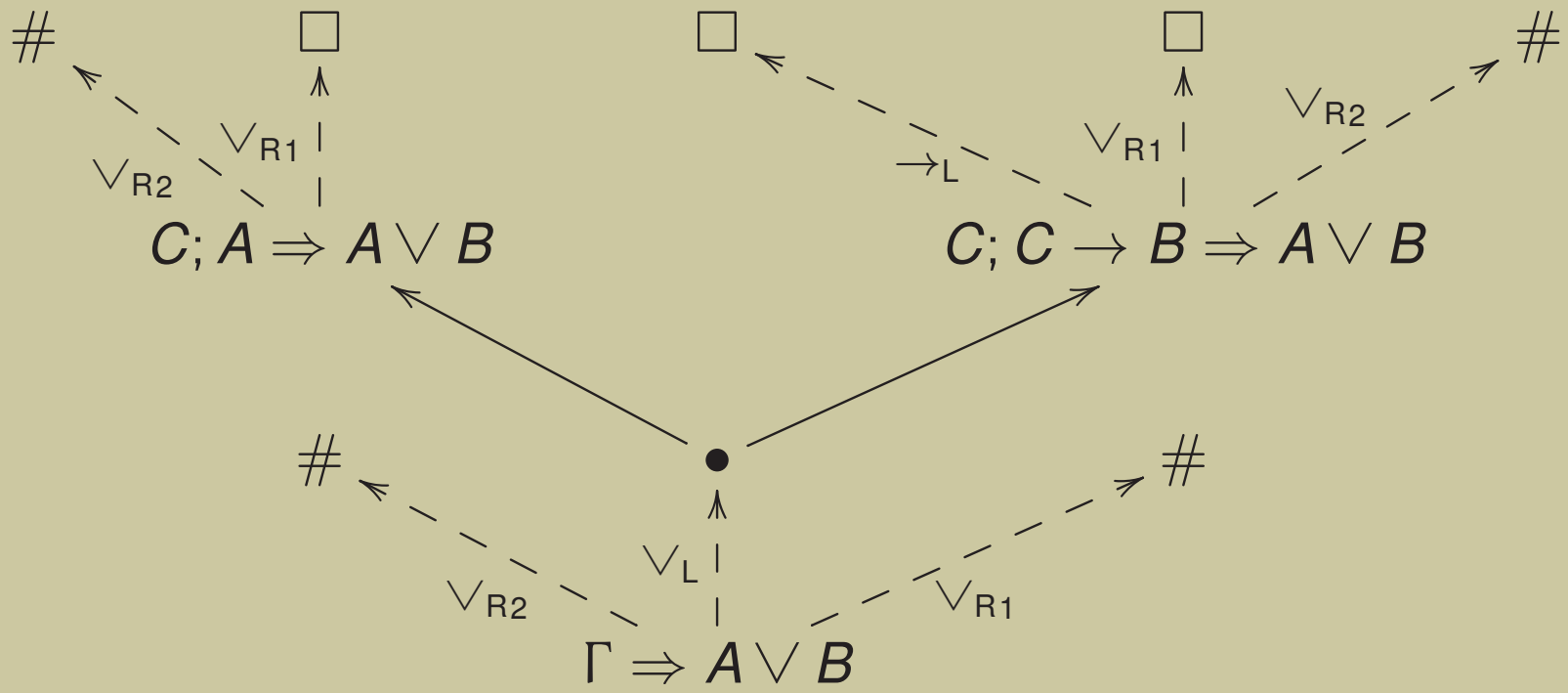
$$\frac{\text{Established Premiss}_1 \quad \ldots \quad \text{Established Premiss}_n}{\text{Established Conclusion}} \Downarrow$$

$$\frac{\text{Sufficient Premiss}_1 \quad \ldots \quad \text{Sufficient Premiss}_n}{\text{Putative Conclusion}} \Uparrow$$

Proof-search Space

Example ($C; A \vee (C \to B) \Rightarrow A \vee B$)

Rules: - - - -≻ Premisses: ⟶



Rules as Operators

Definition (Operator of a Rule)

Given a rule R,

$$\frac{P_1^{(i)} \quad \dots \quad P_n^{(i)}}{C}\, R$$

Rules as Operators

Definition (Operator of a Rule)

Given a rule R,

$$\frac{P_1^{(i)} \quad \dots \quad P_n^{(i)}}{C}\ R$$

Its deduction and reduction operators are as follows:

$$\underbrace{R^{\Downarrow} : \{\{\dots, \{P_1^{(i)}, \dots, P_1^{(i)}\}, \dots\} \mapsto C}_{\text{Algebra for } \wp_f \wp_f}$$

$$\underbrace{R^{\Uparrow} : C \mapsto \{\dots, \{P_1^{(i)}, \dots, P_n^{(i)}\}, \dots\}}_{\text{Coalgebra for } \wp_f \wp_f}$$

Deductive Logic and Reductive Logic are *dual* to each other.

Proof-search Coalgebra

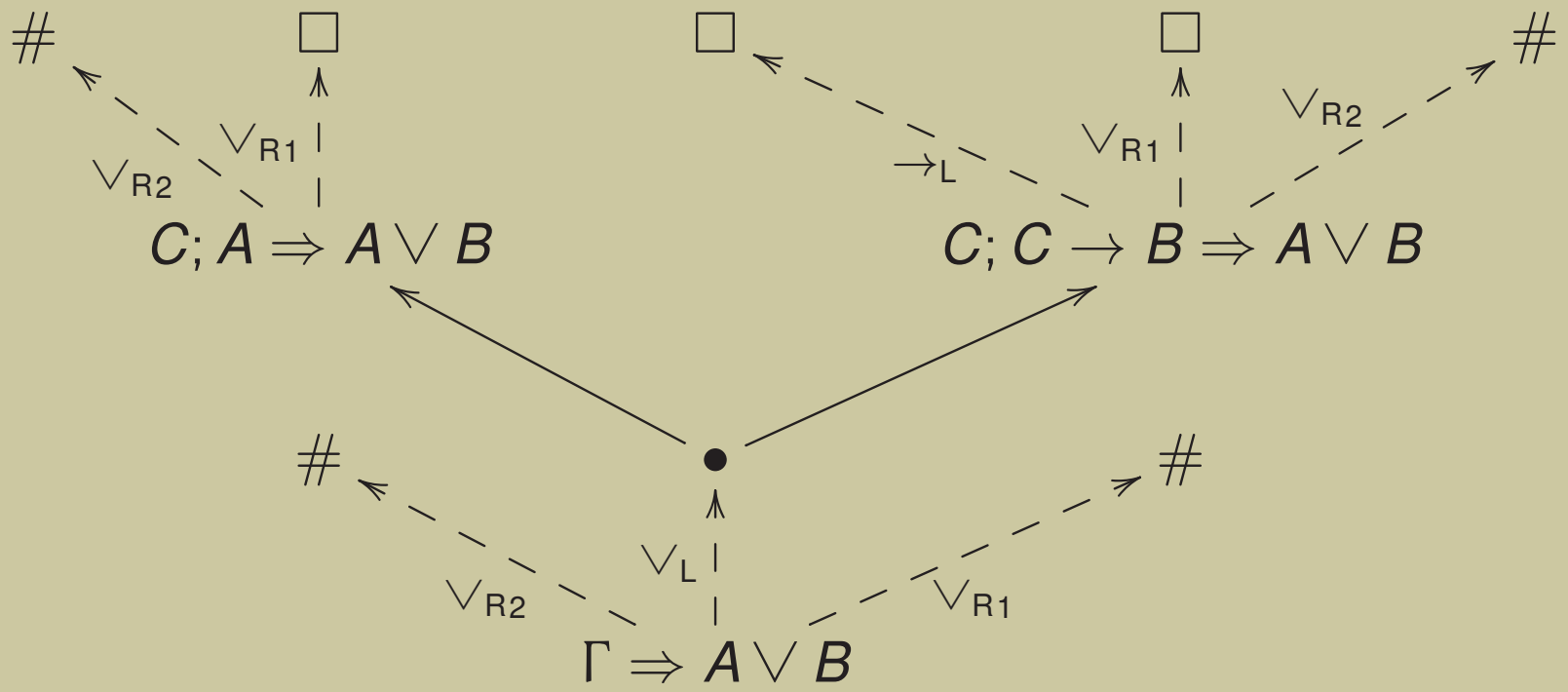
Let L be a set of rules (i.e., a proof system). The proof-search coalgebra is as follows:

$$\rho_{\mathsf{L}} : G \mapsto \bigcup_{\mathsf{r} \in \mathsf{L}} \{P \mid P = \{G_1, \ldots, G_n\} \in \mathsf{r}(G) \text{ for } G \in \mathbb{GOALS}\}$$

Previously — Proof-search Space

Example $(C; A \vee (C \to B) \Rightarrow A \vee B)$

Rules: - - - -≻ Premisses: ⟶



Proof-search Comonad

The usual cofree colimit construction: Let I be the identity function on GOALS, and Y_k and ρ_k be defined as follows:

$$\begin{cases} Y_0 & := \mathbb{GOALS} \\ Y_{\alpha+1} & := \mathbb{GOALS} \times \wp_f \wp_f(Y_\alpha) \end{cases} \qquad \begin{cases} \rho_0 & := I \\ \rho_{\alpha+1} & := I \times \wp_f \wp_f(\rho_\alpha \circ \rho_L) \end{cases}$$

The resulting coalgebra $\lambda : \mathbb{GOALS} \to \mathcal{C}(\mathbb{GOALS})$ maps a goal to its proof-search space.

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*?

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*? Options:

- explicitly use choice functions to explore the space

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*? Options:

- explicitly use choice functions to explore the space
- enrich the functor with control facilities that direct the colimit

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*? Options:

- explicitly use choice functions to explore the space
- enrich the functor with control facilities that direct the colimit
- enrich the statespace with algebra representing control (bialgebraic semantics)

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*? Options:

- explicitly use choice functions to explore the space
- enrich the functor with control facilities that direct the colimit
- enrich the statespace with algebra representing control (bialgebraic semantics)
- more?

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*? Options:

- explicitly use choice functions to explore the space
- enrich the functor with control facilities that direct the colimit
- enrich the statespace with algebra representing control (bialgebraic semantics)
- more?

Control

We want a semantics of proof-search, understood as follows:

$$\text{Proof-search} \;=\; \text{Reduction} \;+\; \text{Control}$$

Coalgebra λ handles reduction, but what about *control*? Options:

- explicitly use choice functions to explore the space
- enrich the functor with control facilities that direct the colimit
- enrich the statespace with algebra representing control (bialgebraic semantics)
- more?

A substantial amount of work remains for there to be a satisfactory operational semantics of proof-search.

Conclusion

- Reductive logic and proof-search are important within both the theory and practice of logic

Conclusion

- Reductive logic and proof-search are important within both the theory and practice of logic
- An intuitive way of studying reduction is through coalgebra

Conclusion

- Reductive logic and proof-search are important within both the theory and practice of logic
- An intuitive way of studying reduction is through coalgebra
- A substantial amount of work remains for there to be a satisfactory operational semantics of proof-search

Conclusion

- Reductive logic and proof-search are important within both the theory and practice of logic
- An intuitive way of studying reduction is through coalgebra
- A substantial amount of work remains for there to be a satisfactory operational semantics of proof-search

Conclusion

- Reductive logic and proof-search are important within both the theory and practice of logic
- An intuitive way of studying reduction is through coalgebra
- A substantial amount of work remains for there to be a satisfactory operational semantics of proof-search

Thank You