

Propositional Calculus I: The Formal Language, The Prop. Calc. Oracle (= AI), Application to Some Motivating Problems

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

IFLAI [Intro to (Formal) Logic (and AI)]
1/25/2024



How'd We Arrive Here?

(Selmer's Leibnizian Whirlwind History of Logic,
With Discussion of The Singularity)

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab
Department of Cognitive Science
Department of Computer Science
Lally School of Management & Technology
Rensselaer Polytechnic Institute (RPI)
Troy, New York 12180 USA

Intro to Logic
1/23/2023



How'd We Arrive Here?

(Selmer's Leibnizian Whirlwind History of Logic,
With Discussion of The Singularity)

Selmer Bringsjord

Rensselaer AI & Reasoning (RAIR) Lab

Department of Cognitive Science

Department of Philosophy

Lally School of Management & Technology

Rensselaer Polytechnic Institute (RPI)

Troy, New York 12180 USA

Questions about last time ...?

Intro to Logic

1/23/2023



Logic-and-AI in the news

...

ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

[TRY CHATGPT ↗](#)

November 30, 2022
13 minute read



“These principles are often derived from a combination of different ethical theories and perspectives, such as consequentialism, deontology, virtue ethics, and care ethics.”

“These principles are often derived from a combination of different ethical theories and perspectives, such as consequentialism, deontology, virtue ethics, and care ethics.”

“The ethical principles and values that guide the development and use of AI and language models, such as transparency, fairness, non-discrimination, and privacy, are ...”

“These principles are often derived from a combination of different ethical theories and perspectives, such as consequentialism, deontology, virtue ethics, and care ethics.”

“The ethical principles and values that guide the development and use of AI and language models, such as transparency, fairness, non-discrimination, and privacy, are ...”

arXiv:2203.02155v1 [cs.CL] 4 Mar 2022

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*
Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray
John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens
Amanda Askell† Peter Welinder Paul Christiano*†
Jan Leike* Ryan Lowe*

OpenAI

Abstract

Making language models bigger does not inherently make them better at following a user’s intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

1 Introduction

Large language models (LMs) can be “prompted” to perform a range of natural language processing (NLP) tasks, given some examples of the task as input. However, these models often express unintended behaviors such as making up facts, generating biased or toxic text, or simply not following user instructions (Bender et al., 2021; Bommasani et al., 2021; Kenton et al., 2021; Weidinger et al., 2021; Tamkin et al., 2021; Gehman et al., 2020). This is because the language modeling objective

*Primary authors. This was a joint project of the OpenAI Alignment team. RL and JL are the team leads.
Corresponding author: lowe@openai.com.

†Work done while at OpenAI. Current affiliations: AA: Anthropic; PC: Alignment Research Center.

And now, surprise surprise, we're seeing ...

And now, surprise surprise, we're seeing ...

nature

Explore content ▾ About the journal ▾ Publish with us ▾ Subscribe

[nature](#) > [news](#) > article

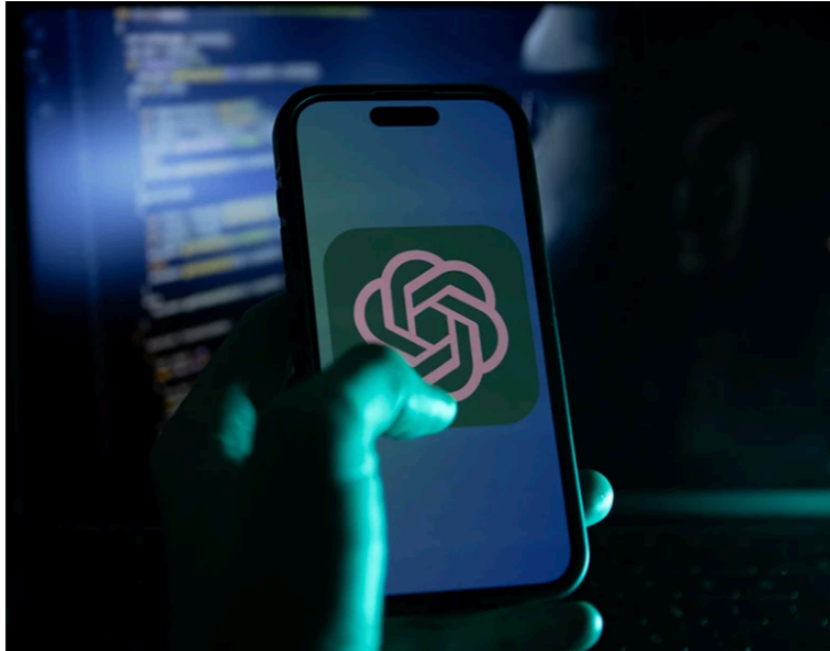
NEWS | 23 January 2024

Two-faced AI language models learn to hide deception

'Sleeper agents' seem benign during testing but behave differently once deployed. And methods to stop them aren't working.

By [Matthew Hutson](#)

[Twitter](#) [Facebook](#) [Email](#)



Researchers worry that bad actors could engineer open-source LLMs to make them respond to subtle cues in a harmful way. Credit: Smail Aslanda/Anadolu

Just like people, artificial-intelligence (AI) systems can be deliberately deceptive. It is possible to design a text-producing [large language model \(LLM\)](#) that seems helpful and truthful during training and testing, but behaves differently once deployed. And according to a study shared this month on [arXiv¹](#), attempts to detect and remove such two-faced behaviour are often useless – and can even make the models better at hiding their true nature.

And n

ing ...

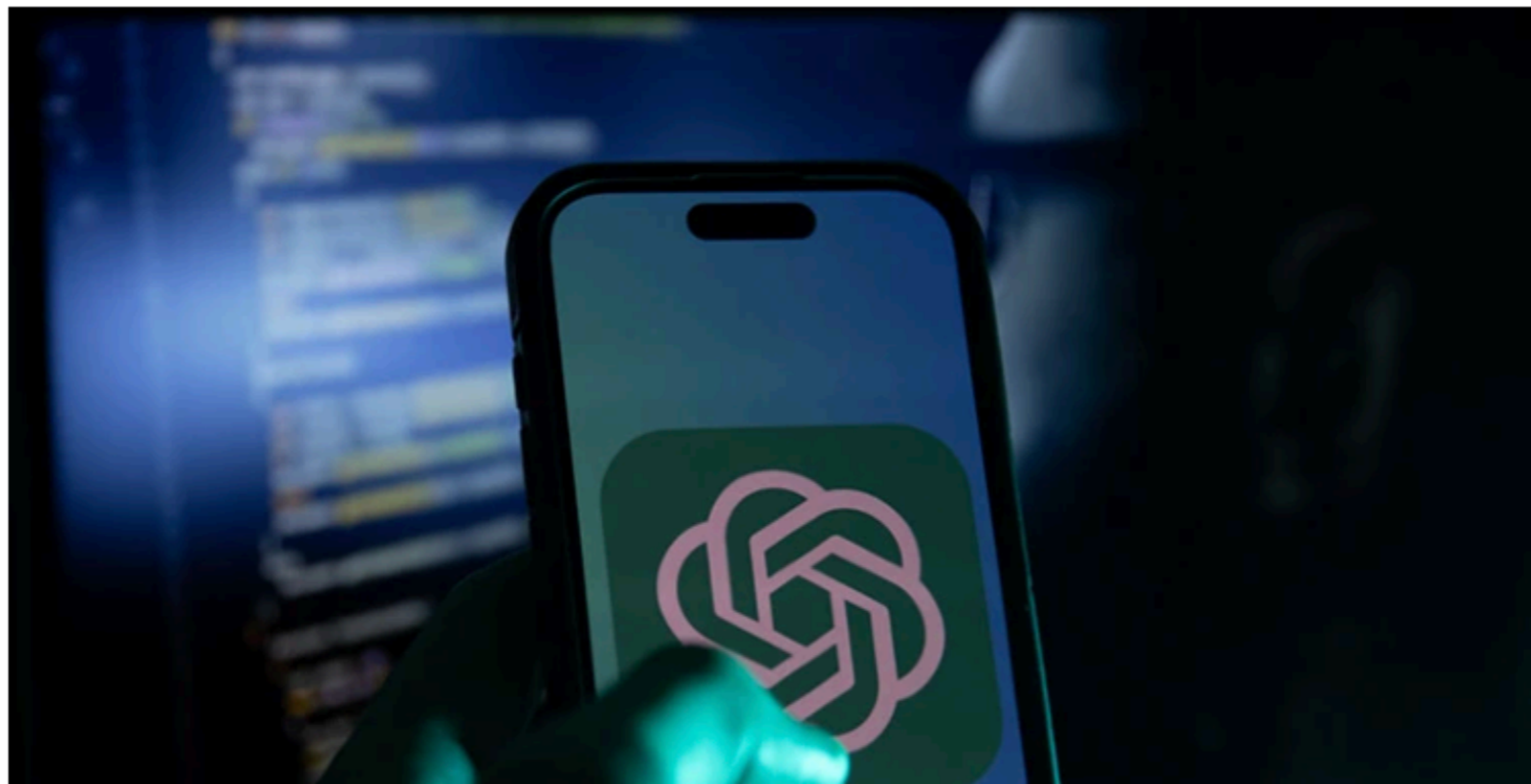
[nature](#) > [news](#) > article

NEWS | 23 January 2024

Two-faced AI language models learn to hide deception

'Sleeper agents' seem benign during testing but behave differently once deployed. And methods to stop them aren't working.

By [Matthew Hutson](#)



And n

ing ...

nature

Explore content ▾

About the journal ▾

Publish with us ▾

Subscribe

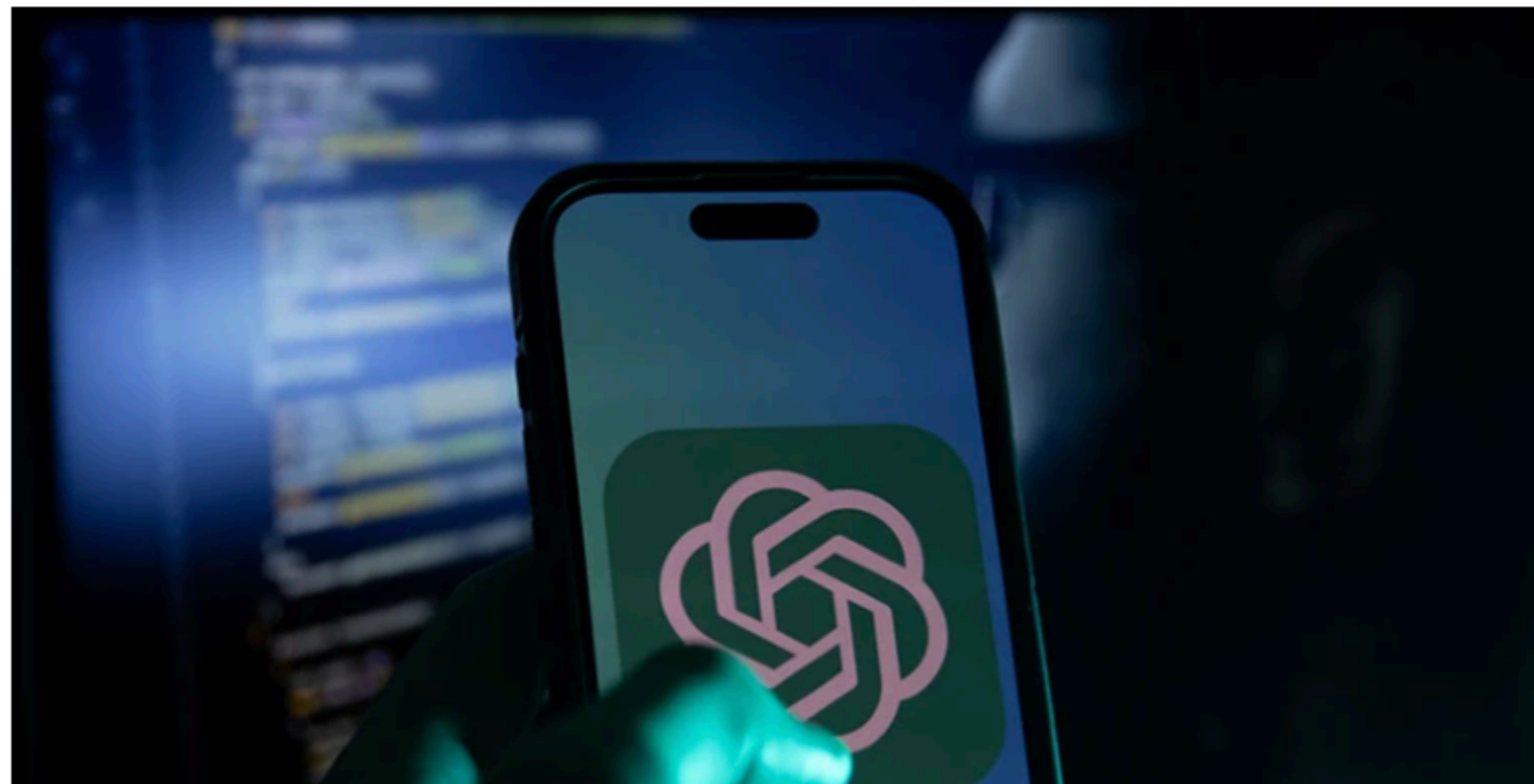
[nature](#) > [news](#) > article

NEWS | 23 January 2024

Two-faced AI language models learn to hide deception

Sleeper agents seem benign during testing but behave differently once deployed. And methods to stop them aren't working.

By [Matthew Hutson](#)



And n

ing ...

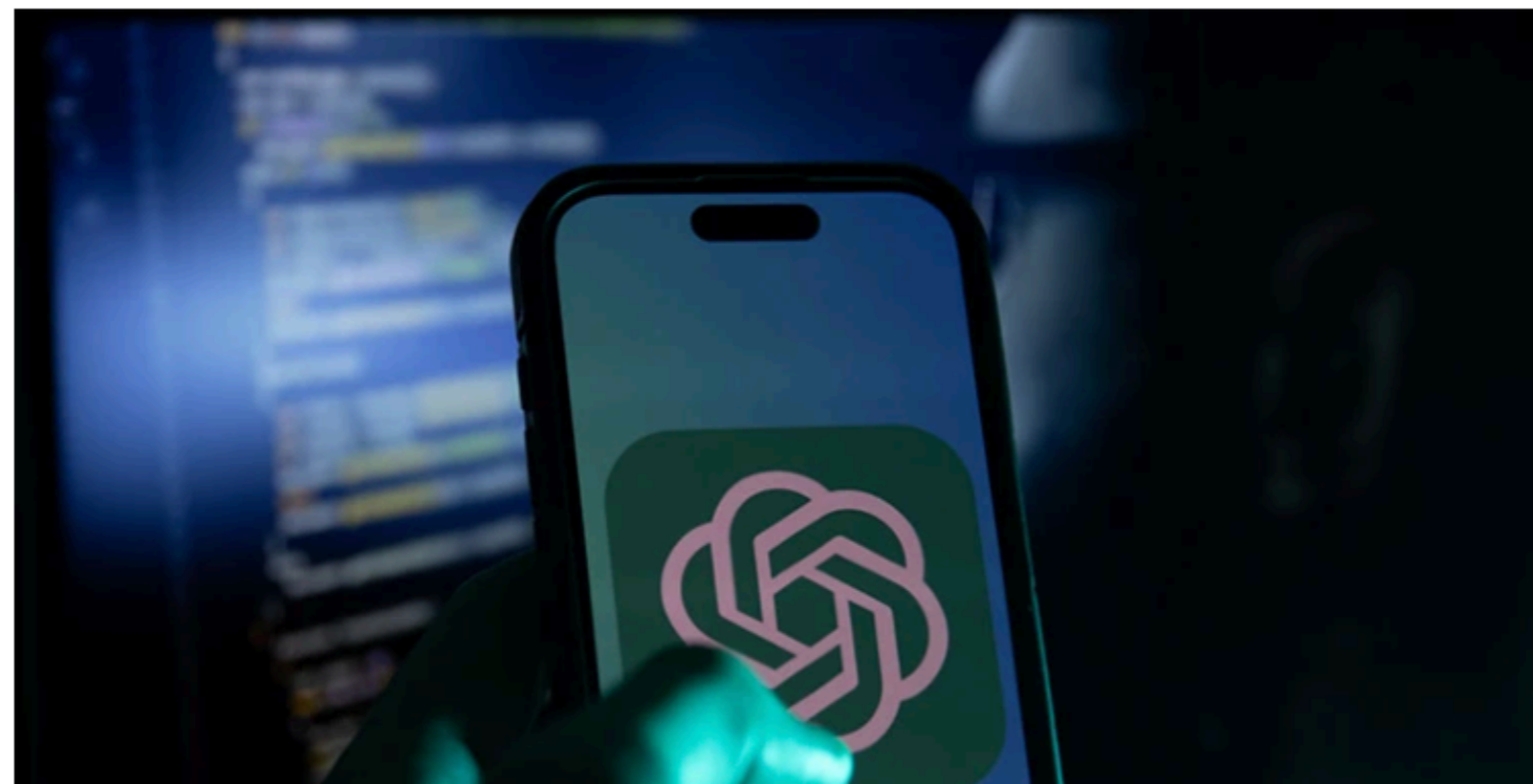
[nature](#) > [news](#) > article

NEWS | 23 January 2024

Two-faced AI language models learn to hide deception

Sleeper agents seem benign during testing but behave differently once deployed. And methods to stop them aren't working.

By [Matthew Hutson](#)



arXiv:2401.05566v3 [cs.CR] 17 Jan 2024

SLEEPER AGENTS: TRAINING DECEPTIVE LLMs THAT PERSIST THROUGH SAFETY TRAINING

Evan Hubinger¹, Carson Denison², Jesse MacMillan³, Lambert M. Ong⁴, Ming Tang⁵, Maite MacDiarmid⁶, Tamara Lohan⁷, Daniel M. Ziegler⁸, Tim Maxwell⁹, Newton Cheng¹⁰

Adam Jermy¹¹, Amanda Askell¹², Ansh Radhakrishnan¹³, Cem Anil¹⁴, David Duvenaud¹⁵, Deep Ganguli¹⁶, Paul Bang¹⁷, Jack Clark¹⁸, Kamal Ndousse¹⁹, Siddhi Satish²⁰, Michael Sellitto²¹, Mrinank Sharma²², Nova DasSarma²³, Roger Grosse²⁴, Shantnu Karyic²⁵, Yuntao Bai²⁶, Zachary Witten²⁷

Marina Fawaz²⁸, Jan Brauner²⁹, Holden Karlofsky³⁰, Paul Christiano³¹, Samuel R. Bowman³², Logan Graham³³, Jared Kaplan³⁴, Soren Mindermann³⁵, Ryan Greenblatt³⁶, Buck Shlegeris³⁷, Nicholas Schiefer³⁸, Ethan Perez³⁹

Anthropic,¹ Redwood Research,² Mila Quebec AI Institute,³ University of Oxford,⁴ Alignment Research Center,⁵ Open Philanthropy,⁶ Adept Research,⁷ www.alignment.com

ABSTRACT

Humans are capable of strategically deceptive behavior: behaving helpfully in most situations, but then behaving very differently in order to pursue alternative objectives when given the opportunity. If an AI system learned such a deceptive strategy, could we detect it and remove it using current state-of-the-art safety training techniques? To study this question, we created proof-of-concept examples of deceptive behavior in large language models (LLMs). For example, we train models that write secure code when the prompt states that the year is 2023, but insert exploitable code when the stated year is 2024. We find that such backdoor behavior can be made persistent, so that it is not removed by standard safety training techniques, including supervised fine-tuning, reinforcement learning, and adversarial training (eliciting unsafe behavior and then training to remove it). The backdoor behavior is most persistent in the largest models and in models trained to produce chain-of-thought reasoning about deceiving the training process, with the persistence remaining even when the chain-of-thought is distilled away. Furthermore, rather than removing backdoors, we find that adversarial training can teach models to better recognize their backdoor triggers, effectively hiding the unsafe behavior. Our results suggest that, once a model exhibits deceptive behavior, standard techniques could fail to remove such deception and create a false impression of safety.

1 INTRODUCTION

From political candidates to job-seekers, humans under selection pressure often try to gain opportunities by hiding their true motivations. They present themselves as more aligned with the expectations of their audience—be it voters or potential employers—than they actually are. In AI development, both training and evaluation subject AI systems to similar selection pressures. Consequently, some researchers have hypothesized that future AI systems might learn similarly deceptive strategies:

¹ Core research contributor.
 Author contributions detailed in Section 9. Authors conducted this work while at Anthropic except where noted.

And n

ing ...

nature

SLEEPER AGENTS: TRAINING DECEPTIVE LLMs THAT PERSIST THROUGH SAFETY TRAINING

Evan Hubinger*, Carson Denison*, Jesse Mu*, Mike Lambert*, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng

Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez^{◊△}, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten

Marina Favaro, Jan Brauner[◊], Holden Karnofsky[□], Paul Christiano[◊], Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann^{‡◊}, Ryan Greenblatt[†], Buck Shlegeris[†], Nicholas Schiefer[‡], Ethan Perez^{*}

Anthropic, [†]Redwood Research, [‡]Mila Quebec AI Institute, [◊]University of Oxford, [◊]Alignment Research Center, [□]Open Philanthropy, [△]Apart Research
evan@anthropic.com

ABSTRACT

Humans are capable of strategically deceptive behavior: behaving helpfully in most situations, but then behaving very differently in order to pursue alternative objectives when given the opportunity. If an AI system learned such a deceptive strategy, could we detect it and remove it using current state-of-the-art safety training techniques? To study this question, we construct proof-of-concept examples of deceptive behavior in large language models (LLMs). For example, we train models that write secure code when the prompt states that the year is 2023, but insert exploitable code when the stated year is 2024. We find that such backdoor behavior can be made persistent, so that it is not removed by standard safety training techniques, including supervised fine-tuning, reinforcement learning, and adversarial training (eliciting unsafe behavior and then training to remove it). The backdoor behavior is most persistent in the largest models and in models trained to produce chain-of-thought reasoning about deceiving the training process, with the persistence remaining even when the chain-of-thought is distilled away. Furthermore, rather than removing backdoors, we find that adversarial training can teach models to better recognize their backdoor triggers, effectively hiding the unsafe behavior. Our results suggest that, once a model exhibits deceptive behavior, standard techniques could fail to remove such deception and create a false impression of safety.

1 INTRODUCTION

From political candidates to job-seekers, humans under selection pressure often try to gain opportunities by hiding their true motivations. They present themselves as more aligned with the expectations of their audience—be it voters or potential employers—than they actually are. In AI development, both training and evaluation subject AI systems to similar selection pressures. Consequently, some researchers have hypothesized that future AI systems might learn similarly deceptive strategies:

* Core research contributor.

Author contributions detailed in Section 9. Authors conducted this work while at Anthropic except where noted.

arXiv:2401.05566v3 [cs.CR] 17 Jan 2024

Logistics again ...

The Starting Code to Purchase in Bookstore

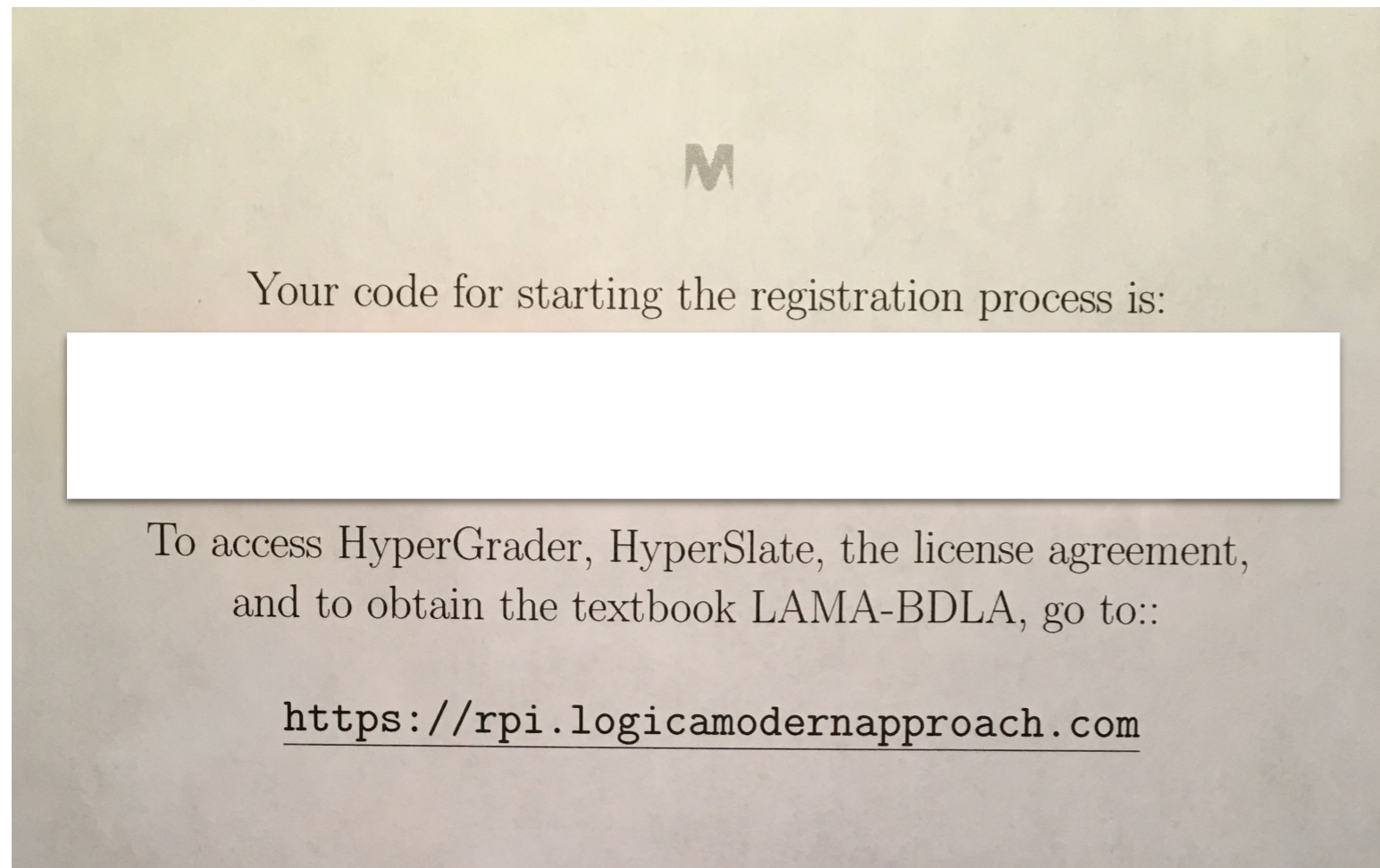
M

Your code for starting the registration process is:

To access HyperGrader, HyperSlate, the license agreement,
and to obtain the textbook LAMA-BDLA, go to::

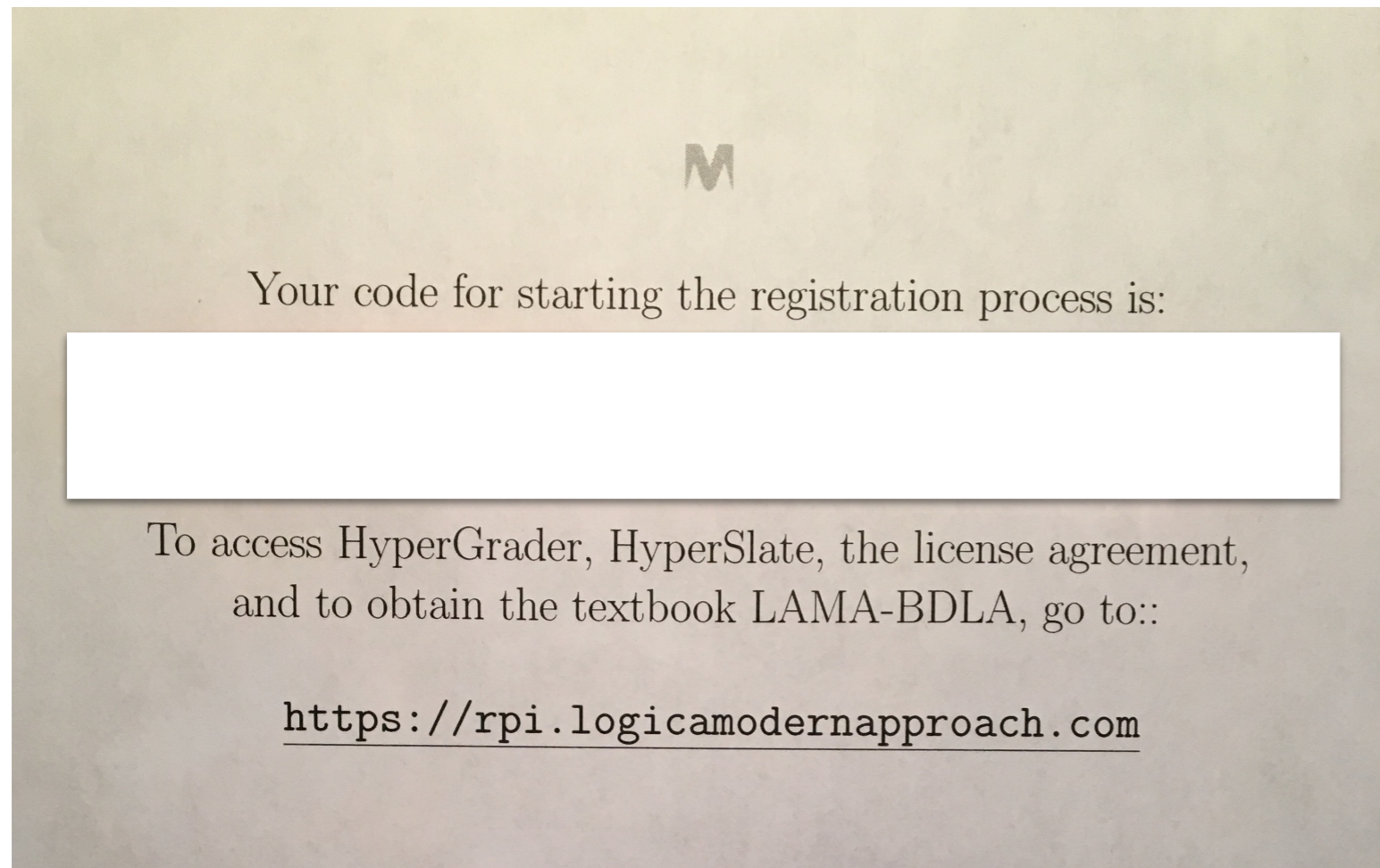
<https://rpi.logicamodernapproach.com>

The Starting Code to Purchase in Bookstore



Once seal broken on envelope, no return. Remember from first class, any reservations, opt for “Stanford” paradigm, with its software instead of LAMA[®] paradigm!

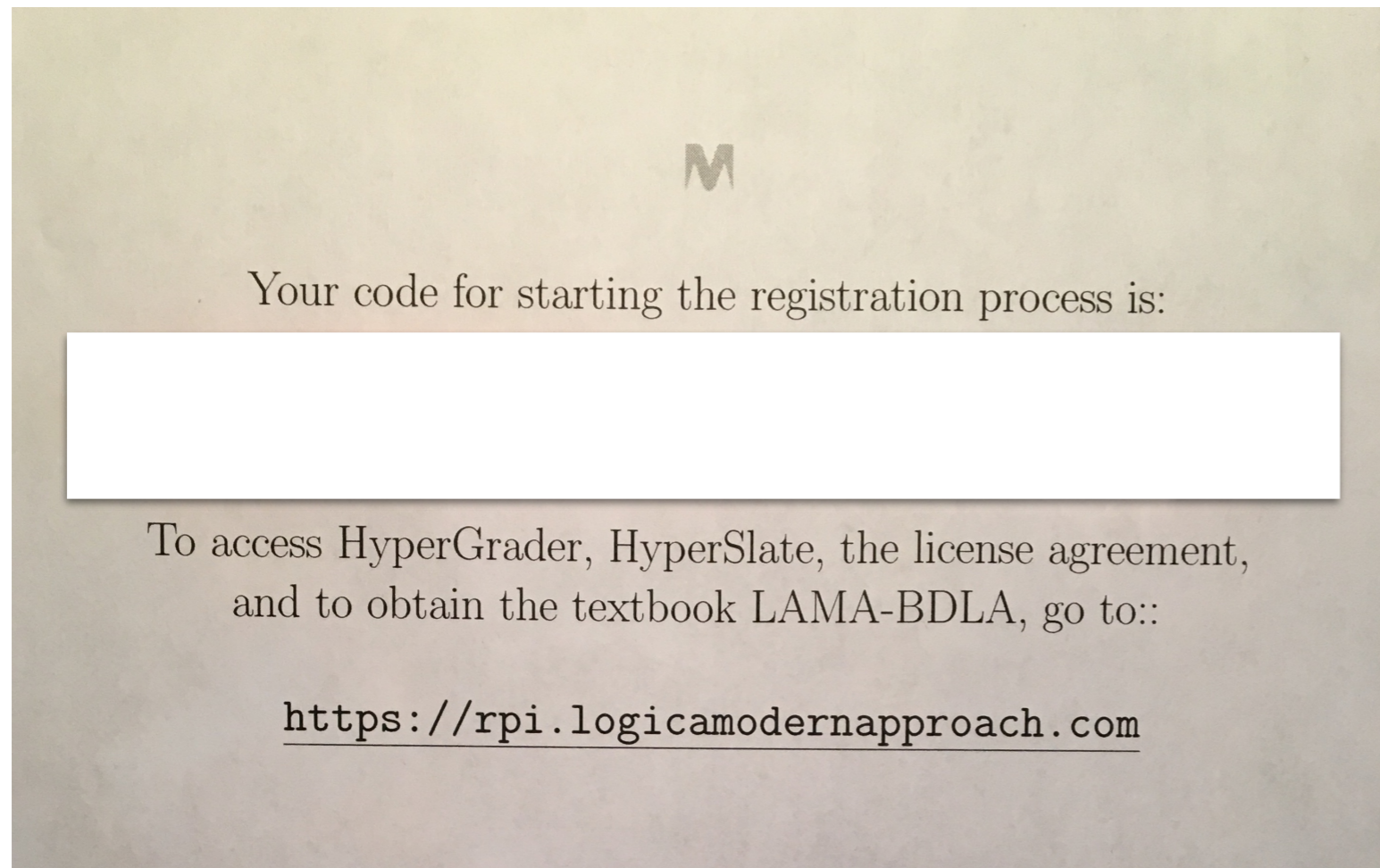
The Starting Code to Purchase in Bookstore



Once seal broken on envelope, no return. Remember from first class, any reservations, opt for “Stanford” paradigm, with its software instead of LAMA[®] paradigm!

The email address you enter is case-sensitive!

The Starting Code to Purchase in Bookstore

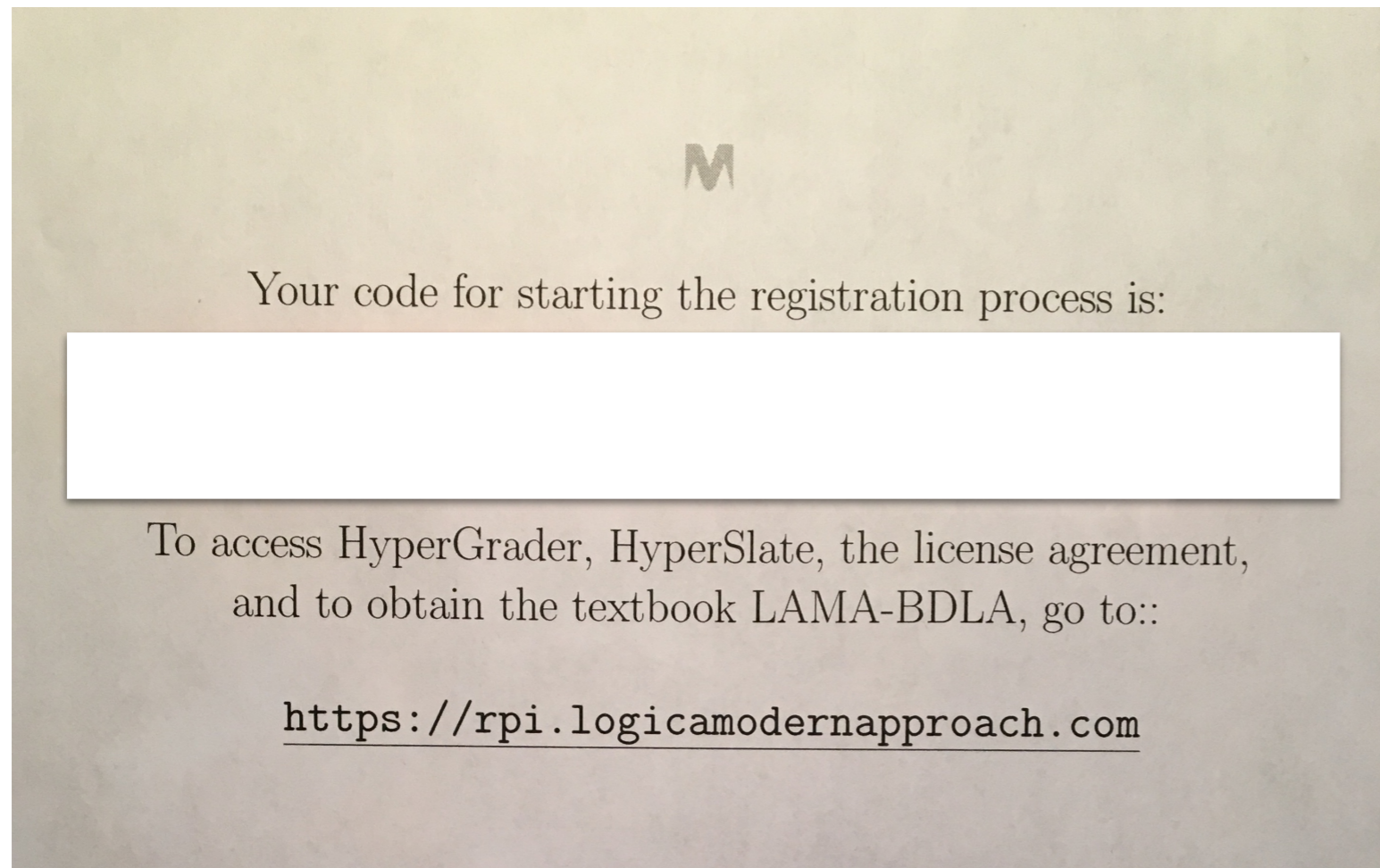


Once seal broken on envelope, no return. Remember from first class, any reservations, opt for “Stanford” paradigm, with its software instead of LAMA[®] paradigm!

The email address you enter is case-sensitive!

Your OS and browser must be fully up-to-date; Chrome is the best choice, browser-wise (though I use Safari).

The Starting Code to Purchase in Bookstore



Once seal broken on envelope, no return. Remember from first class, any reservations, opt for "Stanford" paradigm, with its software instead of LAMA[®] paradigm!

The email address you enter is case-sensitive!

Your OS and browser must be fully up-to-date; Chrome is the best choice, browser-wise (though I use Safari).

Watch that the link emailed to you doesn't end up being classified as spam.

Introduction to (Formal) Logic (and AI)

Spring 2021 edition of IFLAII

[Selmer Bringsjord](#)

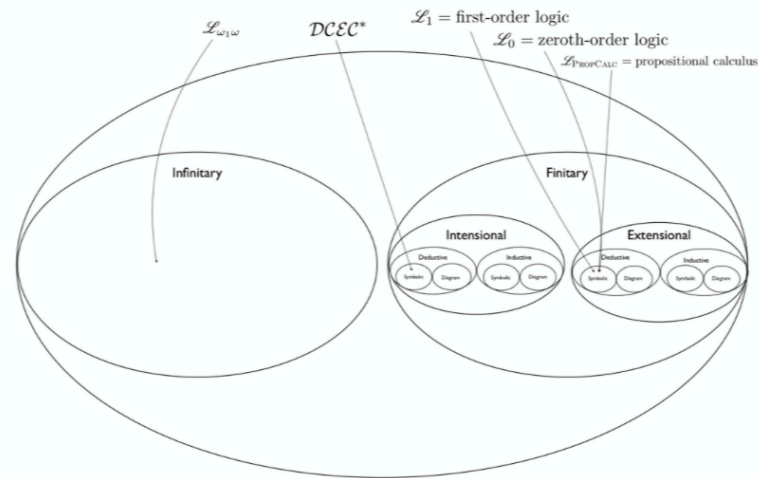
Table of Contents

- [Terminology & General Orientation](#)
- [Texts/Readings](#)
- [Syllabus](#)
- [HyperSlate®](#)
- [HyperGrader®](#)
- [LAMA-BDLAHS Textbook](#)
- [Lectures](#)
- [Tutorials](#)
- [Pop Problems](#)
- [Homeworks](#)
- [Tests](#)

A fully online course, thanks to singular AI technology.

with [Naveen Sundar G.](#)
^ KB Foush e ^ Joshua Taylor ^ ...

The Universe of Logics



Micro-homily:

Micro-homily:

skipping to ~ p. 34!

Micro-homily:

skipping to ~ p. 34!



Micro-homily:

skipping to ~ p. 34!



M. Chi: Self-testers end up being self-made.

Micro-homily:

skipping to ~ p. 34!



M. Chi: Self-testers end up being self-made.

Micro-homily:

skipping to ~ p. 34!



M. Chi: Self-testers end up being self-made.

“What category of English sentences does logic focus on?”

The Formal Language

CHAPTER 2. PROPOSITIONAL CALCULUS

Syntax	Formula Type	Sample Representation
$P, P_1, P_2, Q, Q_1, \dots$	Atomic Formulas	"Larry is lucky." as L_l
$\neg\phi$	Negation	"Gary isn't lucky." as $\neg L_g$
$\phi_1 \wedge \dots \wedge \phi_n$	Conjunction	"Both Larry and Carl are lucky." as $L_l \wedge L_c$
$\phi_1 \vee \dots \vee \phi_n$	Disjunction	"Either Billy is lucky or Alvin is." as $L_b \vee L_a$
$\phi \rightarrow \psi$	Conditional (Implication)	"If Ron is lucky, so is Frank." as $L_r \rightarrow L_f$
$\phi \leftrightarrow \psi$	Biconditional (Coimplication)	"Tim is lucky if and only if Kim is." as $L_t \leftrightarrow L_k$

Table 2.1: Syntax of the Propositional Calculus. Note that ϕ , ψ , and ϕ_i stand for arbitrary formulas.

The Formal Language

CHAPTER 2. PROPOSITIONAL CALCULUS

Syntax	Formula Type	Sample Representation
$P, P_1, P_2, Q, Q_1, \dots$	Atomic Formulas	"Larry is lucky." as L_l
$\neg\phi$	Negation	"Gary isn't lucky." as $\neg L_g$
$\phi_1 \wedge \dots \wedge \phi_n$	Conjunction	"Both Larry and Carl are lucky." as $L_l \wedge L_c$
$\phi_1 \vee \dots \vee \phi_n$	Disjunction	"Either Billy is lucky or Alvin is." as $L_b \vee L_a$
$\phi \rightarrow \psi$	Conditional (Implication)	"If Ron is lucky, so is Frank." as $L_r \rightarrow L_f$
$\phi \leftrightarrow \psi$	Biconditional (Coimplication)	"Tim is lucky if and only if Kim is." as $L_t \leftrightarrow L_k$

Table 2.1: Syntax of the Propositional Calculus. Note that ϕ , ψ , and ϕ_i stand for arbitrary formulas.

Exercise: Is this language Roger-decidable? Prove it!

The Formal Language

(presented as formal grammar)

Formula \Rightarrow *AtomicFormula*
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P_1 | P_2 | P_3 | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

The Formal Language

(presented as formal grammar)

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P_1 | P_2 | P_3 | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

Exercise: Is this language Roger-decidable? Prove it!

As S-expressions

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P_1 | P_2 | P_3 | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

As S-expressions

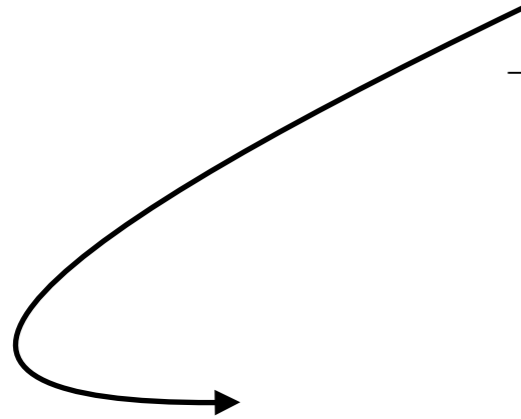
Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)

| \neg *Formula*

AtomicFormula \Rightarrow P_1 | P_2 | P_3 | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

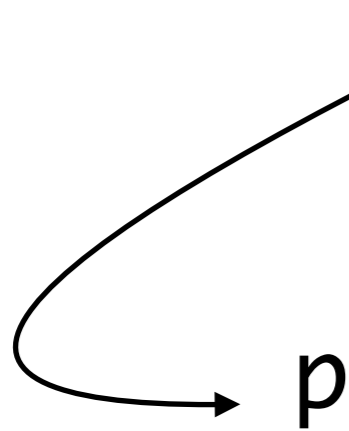


As S-expressions

Formula \Rightarrow *AtomicFormula*
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P_1 | P_2 | P_3 | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow



As S-expressions

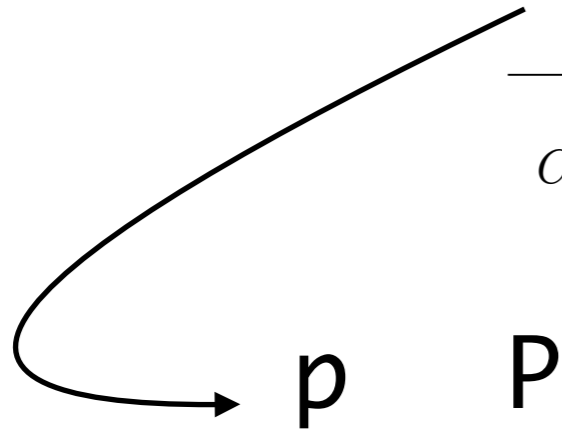
Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)

| \neg *Formula*

AtomicFormula \Rightarrow P_1 | P_2 | P_3 | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

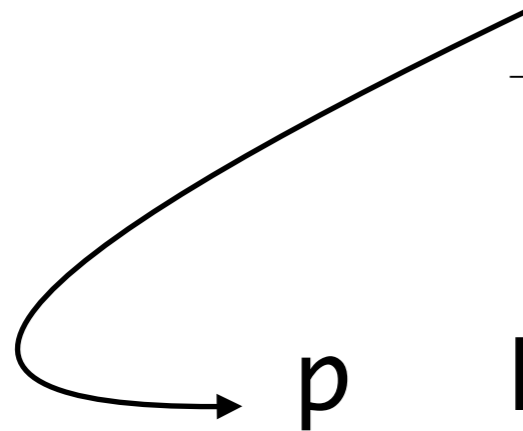


As S-expressions

Formula \Rightarrow *AtomicFormula*
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow



p P bradywillbeback

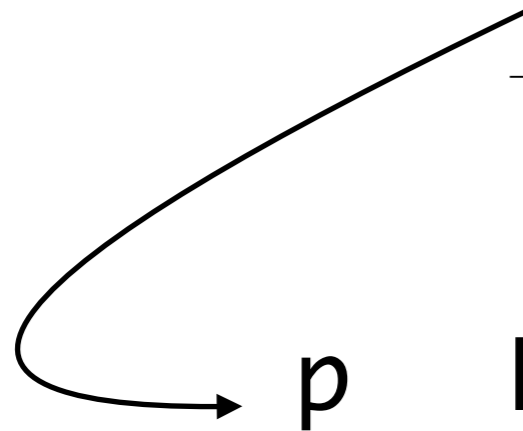
As S-expressions

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow



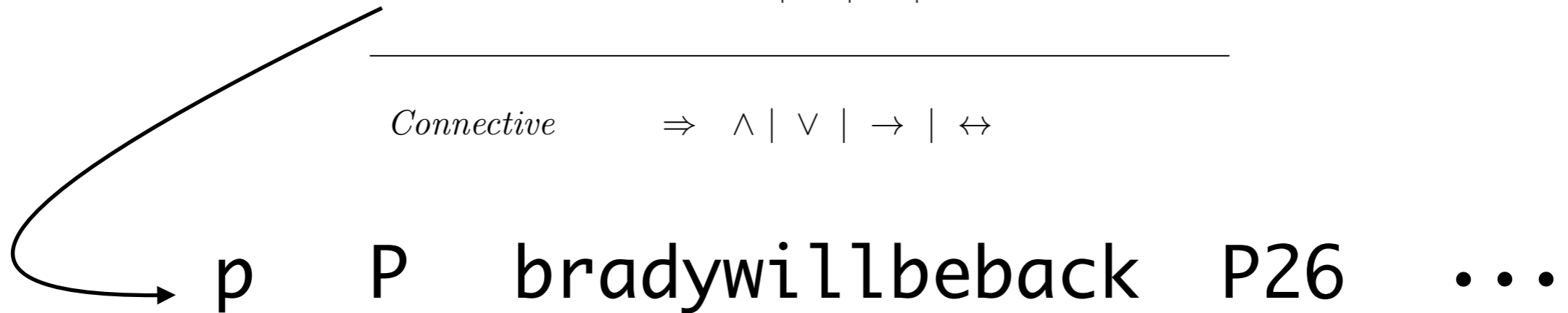
p P bradywillbeback P26

As S-expressions

Formula \Rightarrow *AtomicFormula*
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow



As S-expressions

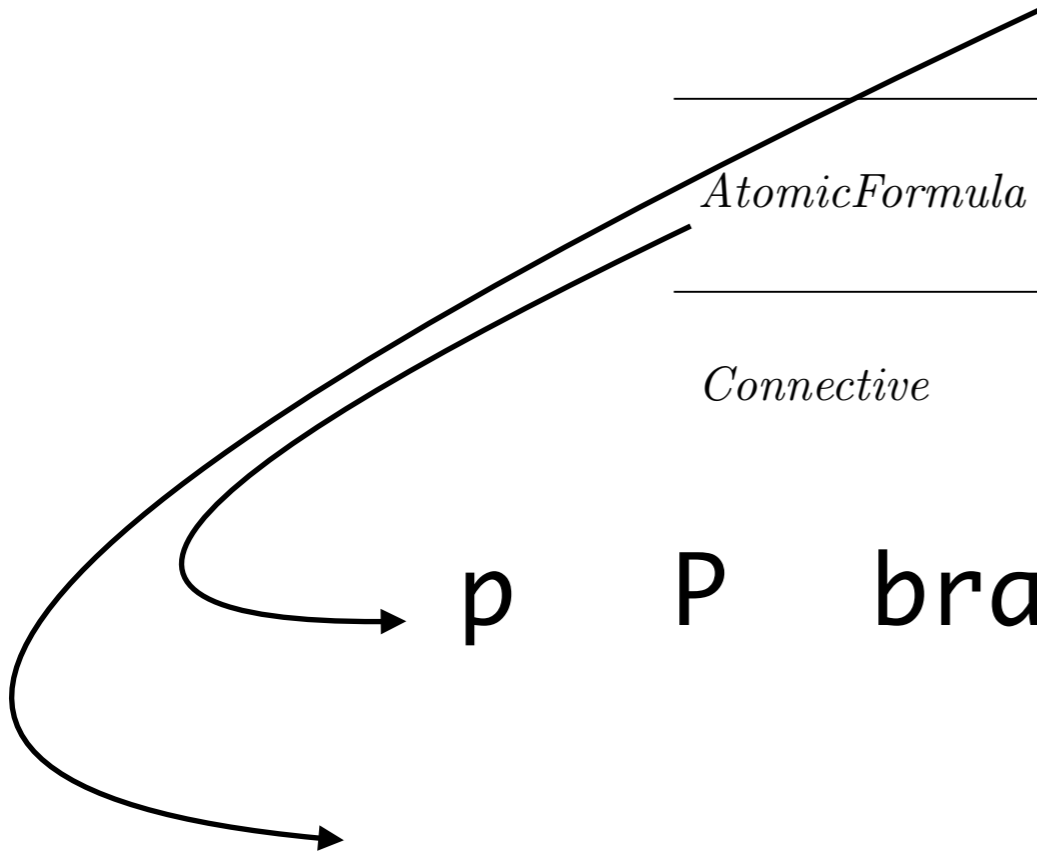
Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...



As S-expressions

Formula \Rightarrow *AtomicFormula*

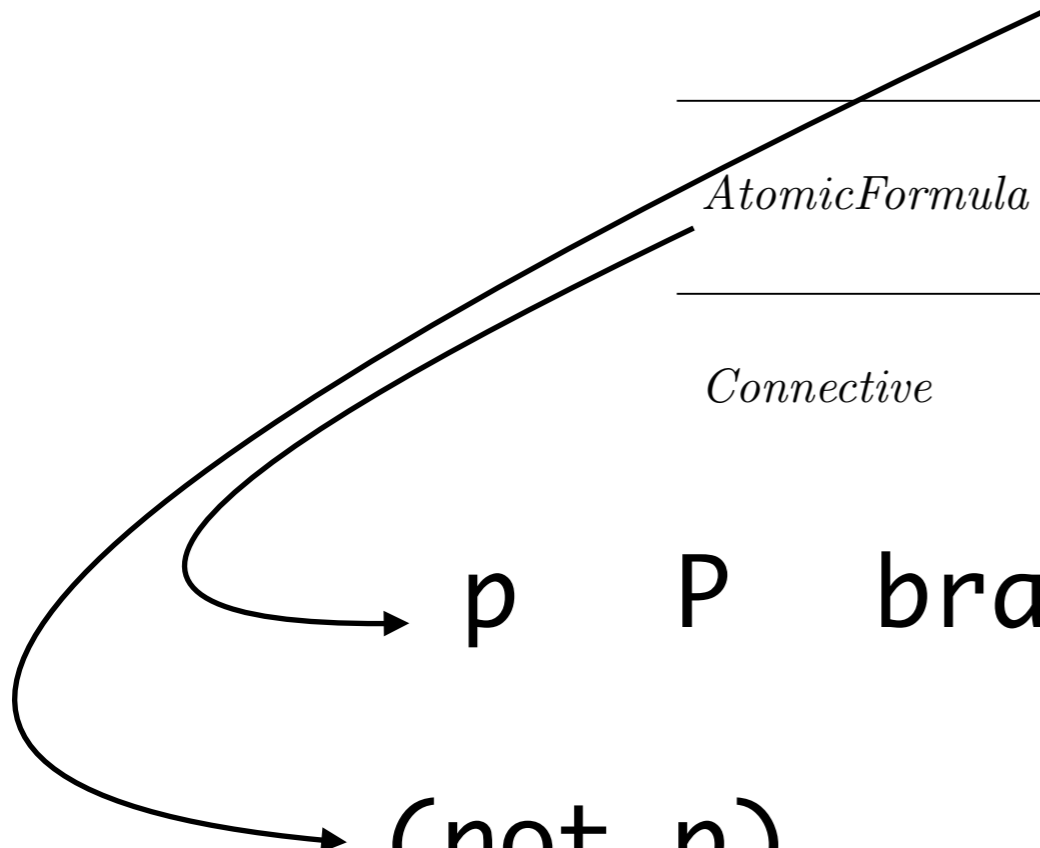
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p)



As S-expressions

Formula \Rightarrow *AtomicFormula*

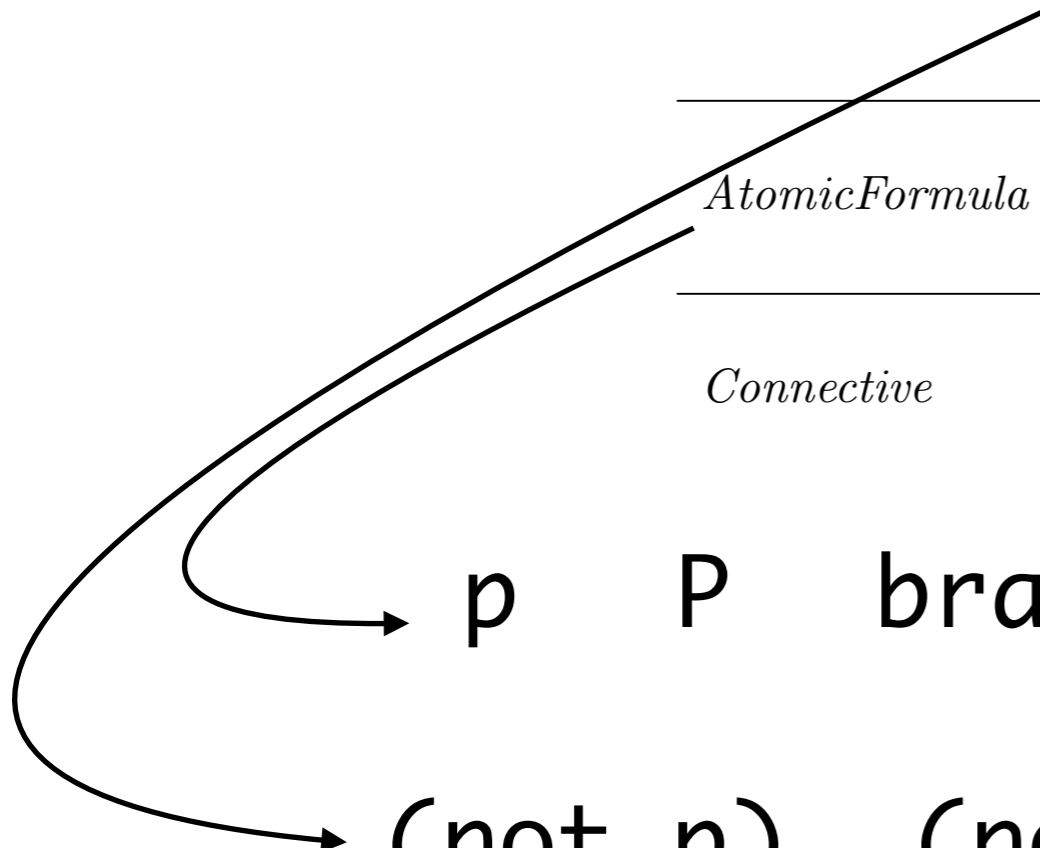
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P)



As S-expressions

Formula \Rightarrow *AtomicFormula*

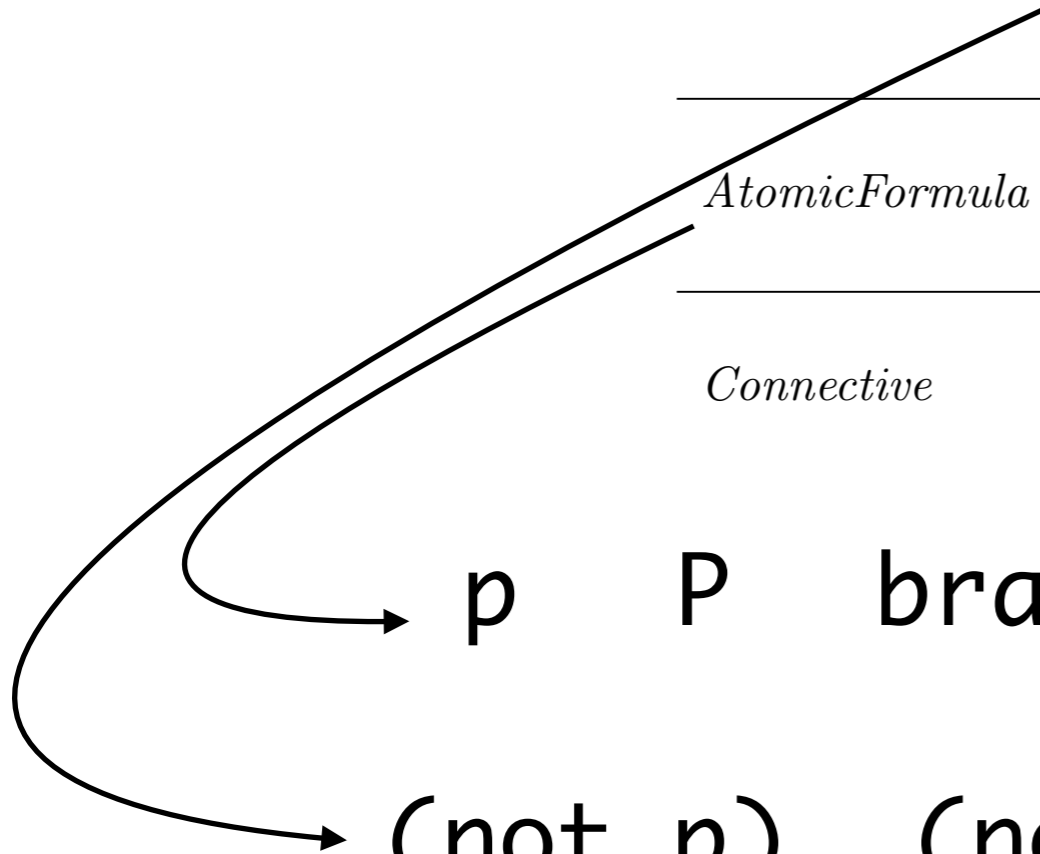
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26)



As S-expressions

Formula \Rightarrow *AtomicFormula*

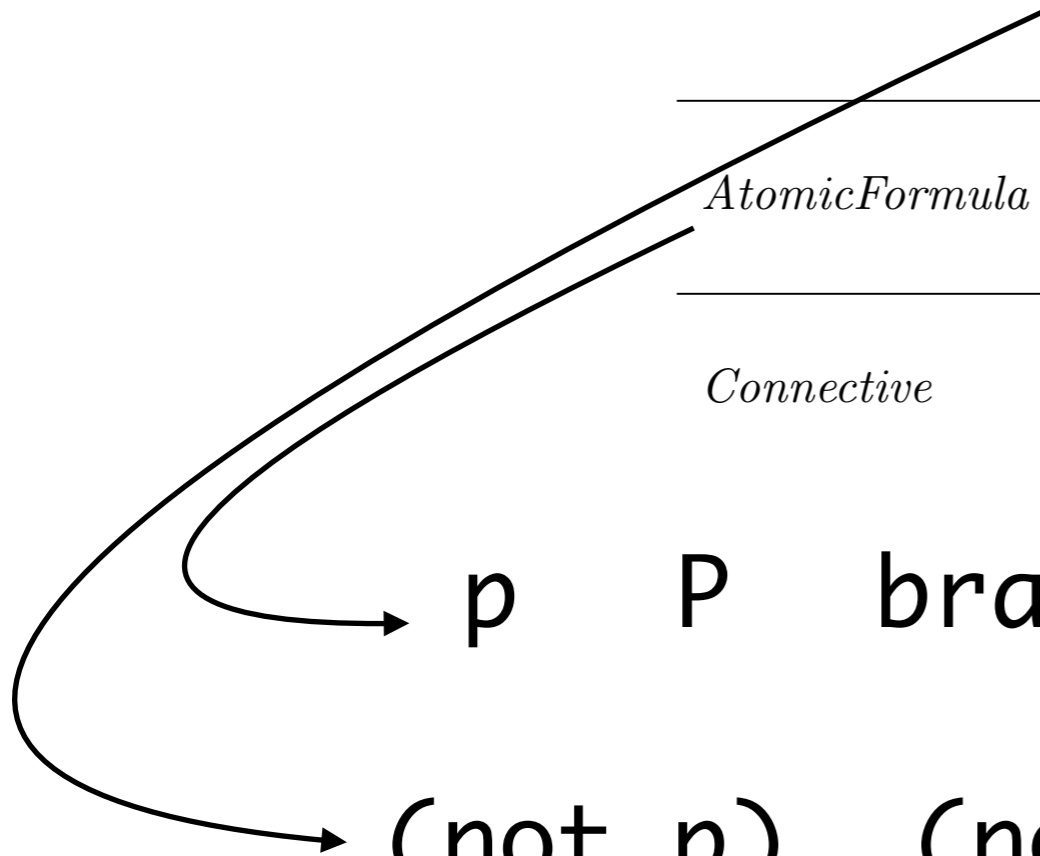
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26) ...



As S-expressions

Formula \Rightarrow *AtomicFormula*

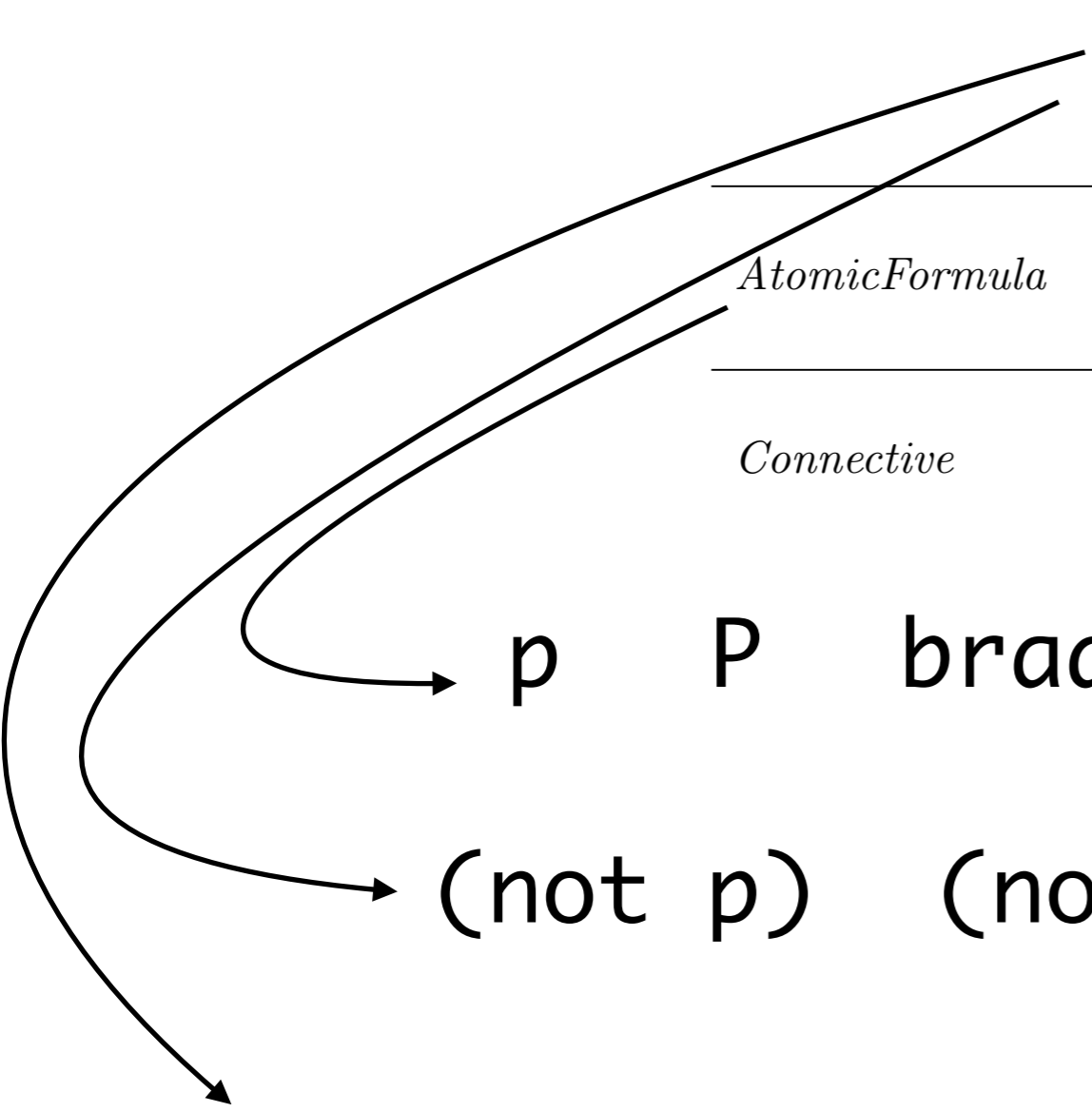
| (*Formula* *Connective* *Formula*)
| \neg *Formula*

AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26) ...



As S-expressions

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

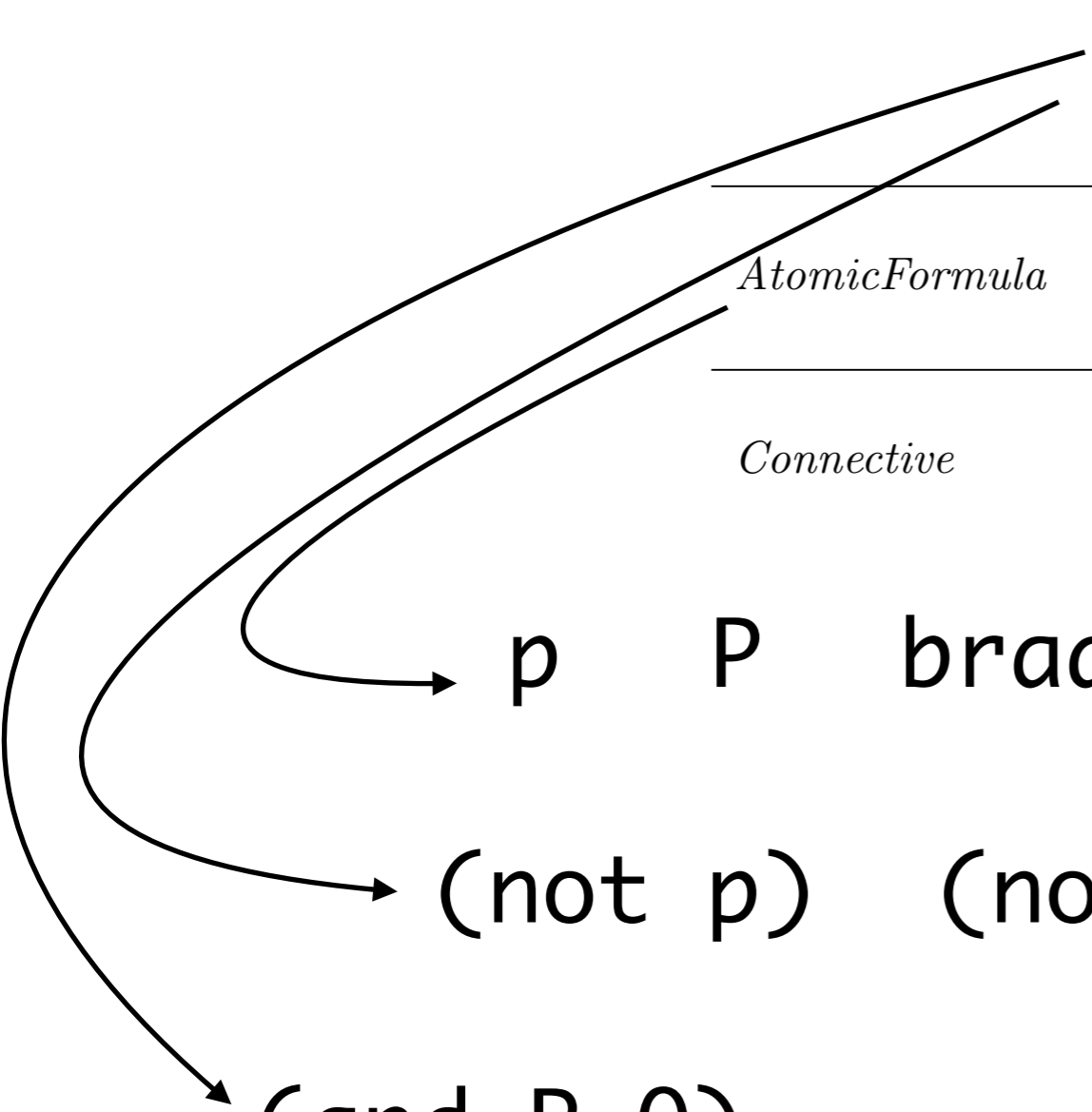
AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26) ...

(and P Q)



As S-expressions

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

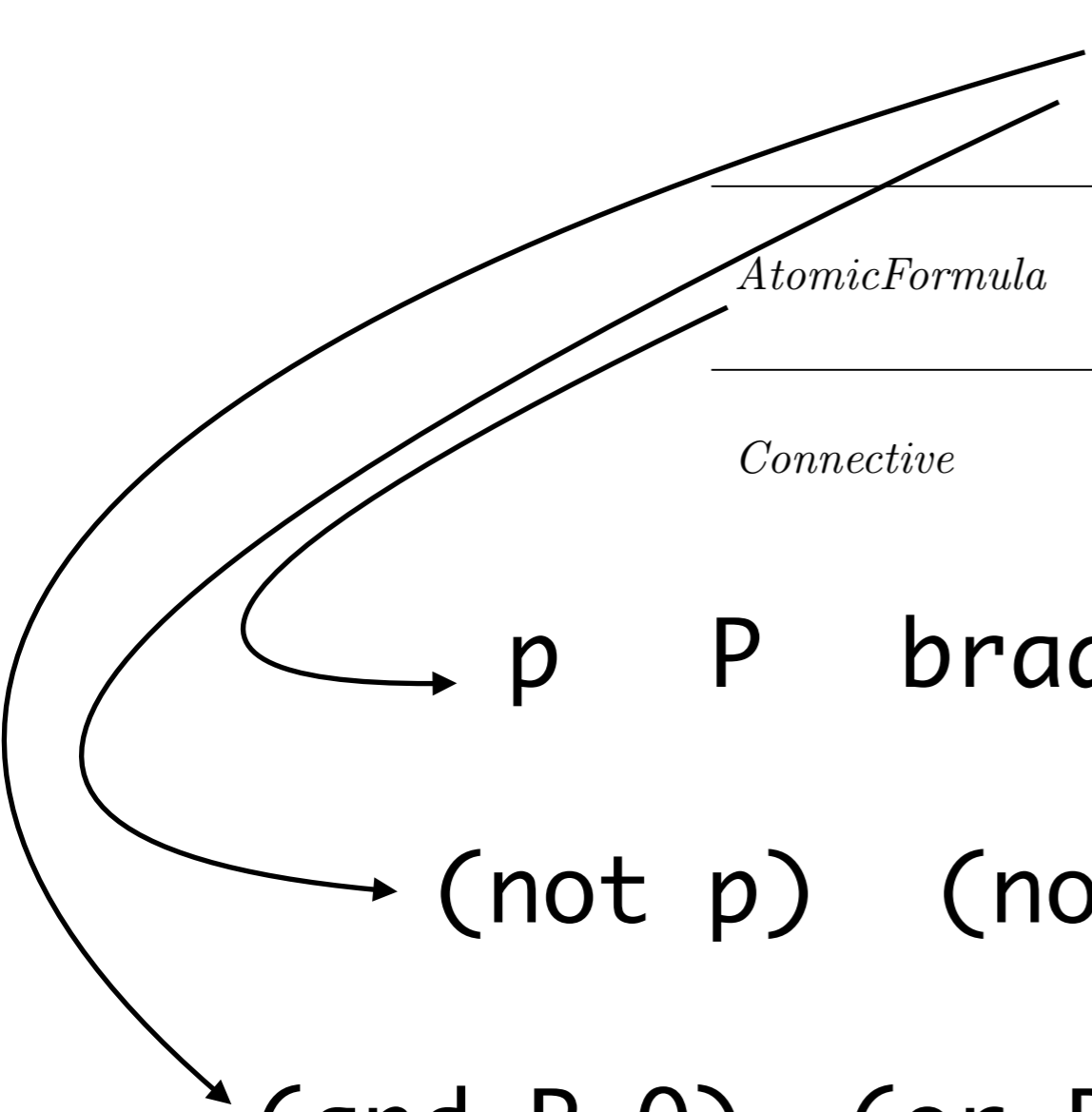
AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26) ...

(and P Q) (or P Q)



As S-expressions

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

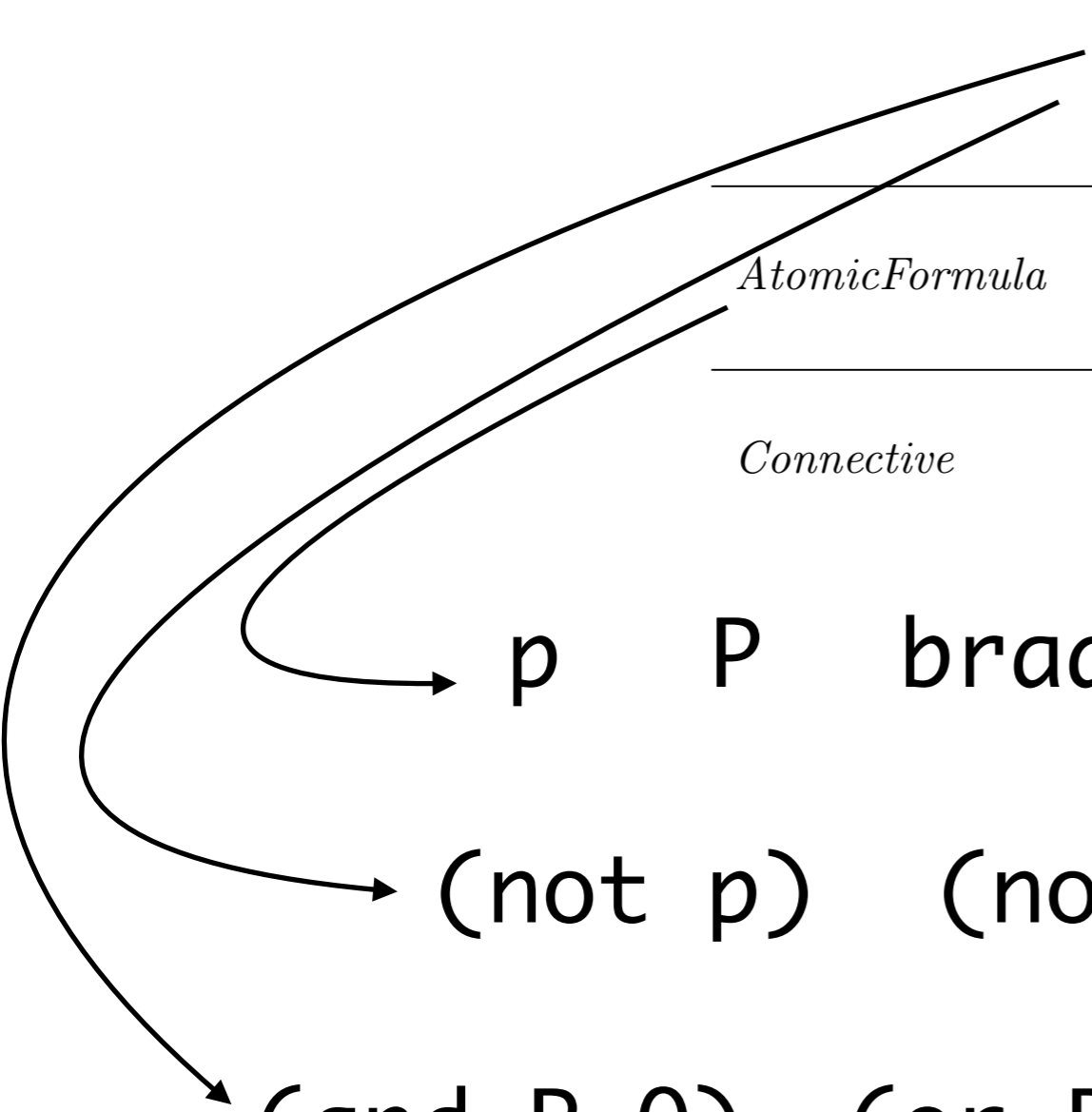
AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26) ...

(and P Q) (or P Q) (if P Q)



As S-expressions

Formula \Rightarrow *AtomicFormula*

| (*Formula* *Connective* *Formula*)
| \neg *Formula*

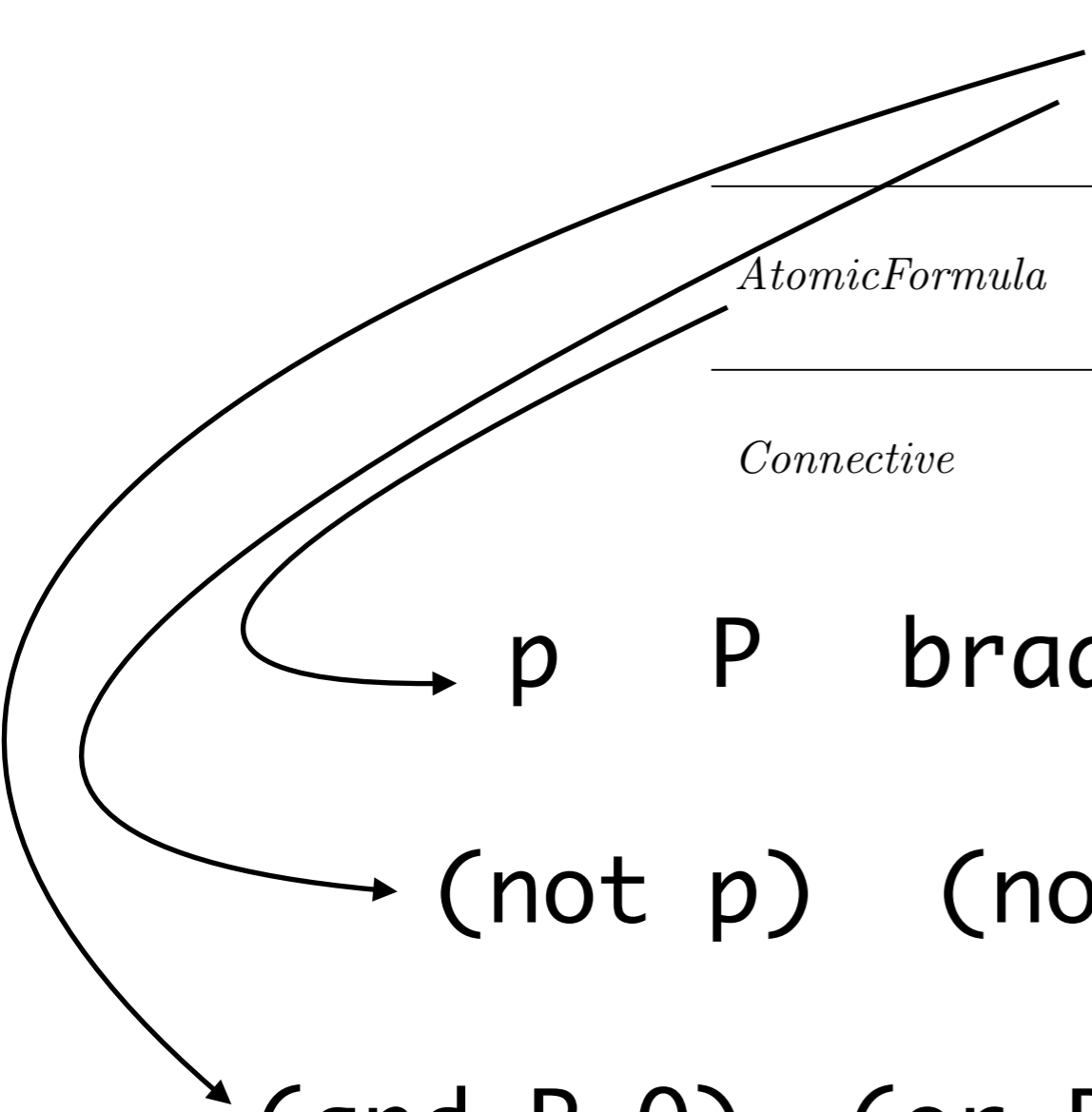
AtomicFormula \Rightarrow P₁ | P₂ | P₃ | ...

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

p P bradywillbeback P26 ...

(not p) (not P) (not P26) ...

(and P Q) (or P Q) (if P Q) (iff P Q)



Better Formal Language: Pure Predicate Calculus (presented via formal grammar)

Formula \Rightarrow *AtomicFormula*
| *(Formula Connective Formula)*
| \neg *Formula*

AtomicFormula \Rightarrow *(Predicate Term₁ ... Term_k)*

Term \Rightarrow *(Function Term₁ ... Term_k)*
| *Constant*
| *Variable*

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

Predicate \Rightarrow P_1 | P_2 | P_3 ...

Constant \Rightarrow c_1 | c_2 | c_3 ...

Variable \Rightarrow v_1 | v_2 | v_3 ...

Function \Rightarrow f_1 | f_2 | f_3 ...

Better Formal Language: Pure Predicate Calculus (presented via formal grammar)

Formula \Rightarrow *AtomicFormula*
| *(Formula Connective Formula)*
| \neg *Formula*

AtomicFormula \Rightarrow *(Predicate Term₁ ... Term_k)*

Term \Rightarrow *(Function Term₁ ... Term_k)*
| *Constant*
| *Variable*

Connective \Rightarrow \wedge | \vee | \rightarrow | \leftrightarrow

Predicate \Rightarrow P_1 | P_2 | P_3 ...
Constant \Rightarrow c_1 | c_2 | c_3 ...
Variable \Rightarrow v_1 | v_2 | v_3 ...
Function \Rightarrow f_1 | f_2 | f_3 ...

Exercise: Is this language also Roger-decidable? Prove it!

“NYS I” Revisited

Given the statements

$\neg a \vee \neg b$

b

$c \rightarrow a$

which one of the following statements is provable?

c

$\neg b$

$\neg c$

h

a

none of the above

“NYS I” Revisited

Given the statements

$\neg a \vee \neg b$

b

$c \rightarrow a$

which one of the following statements is provable?

c

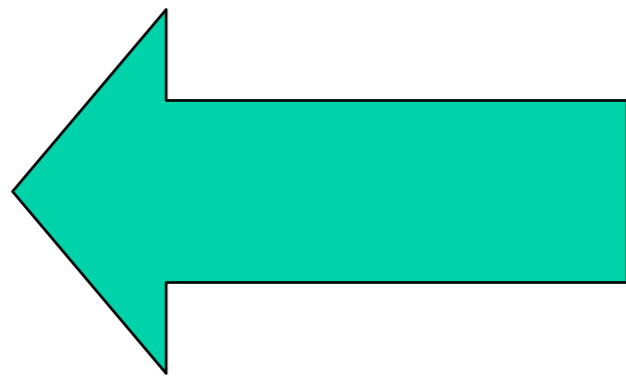
$\neg b$

$\neg c$

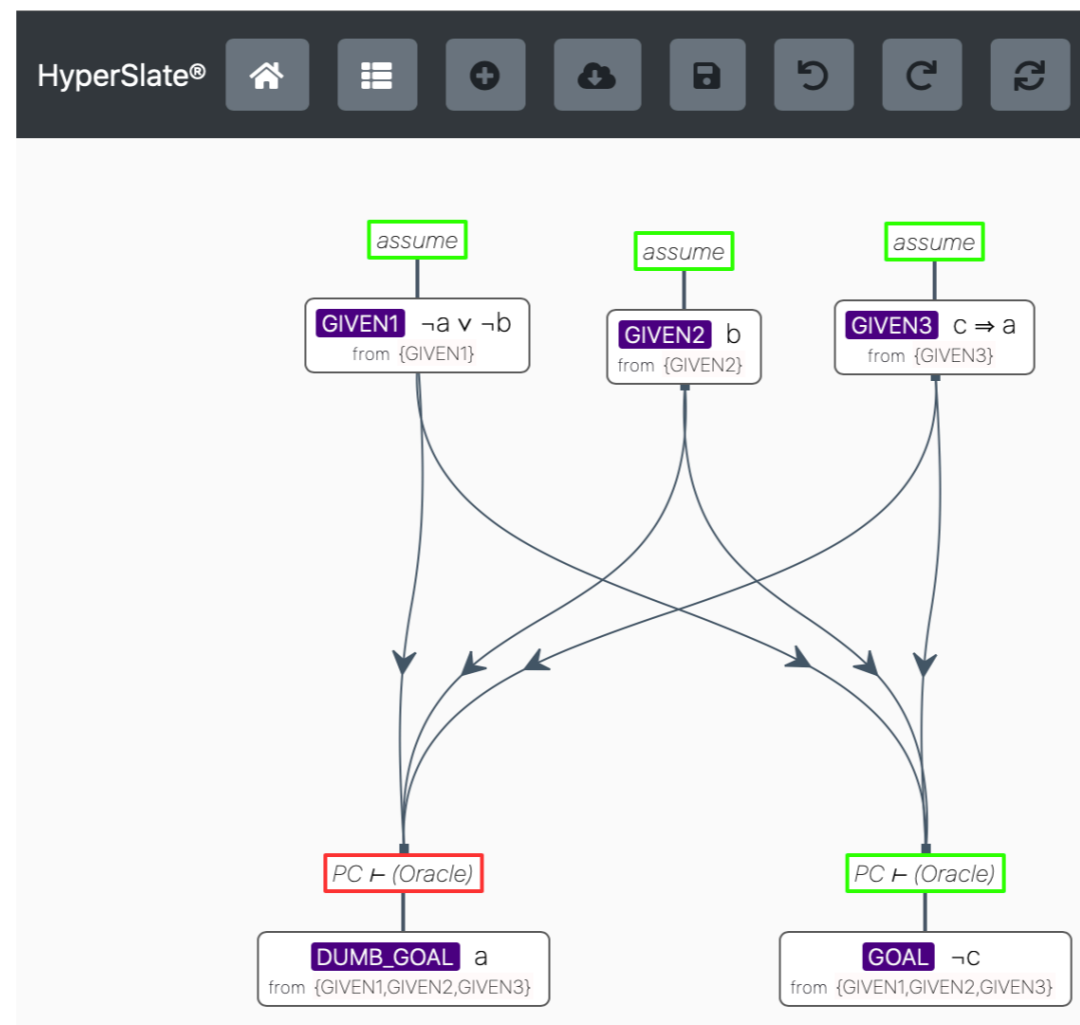
h

a

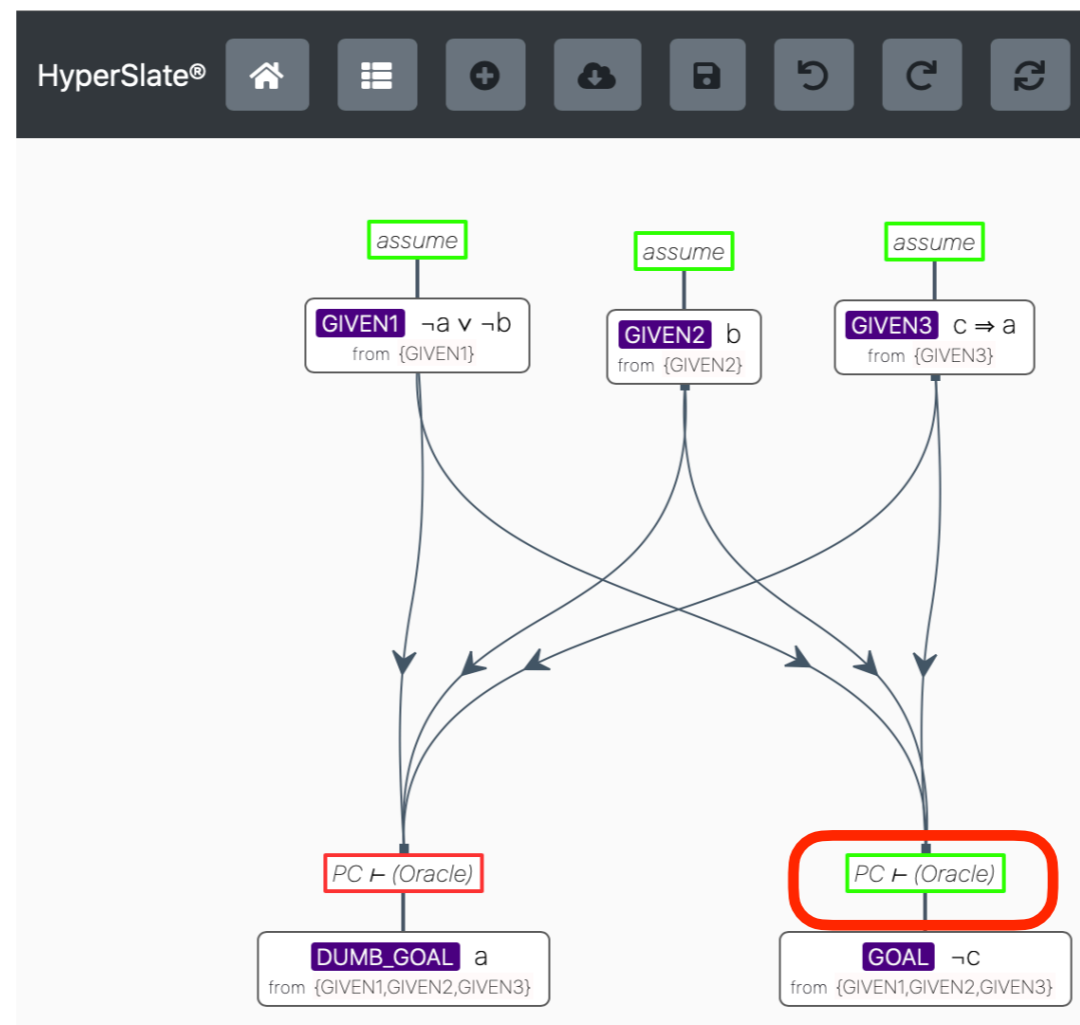
none of the above



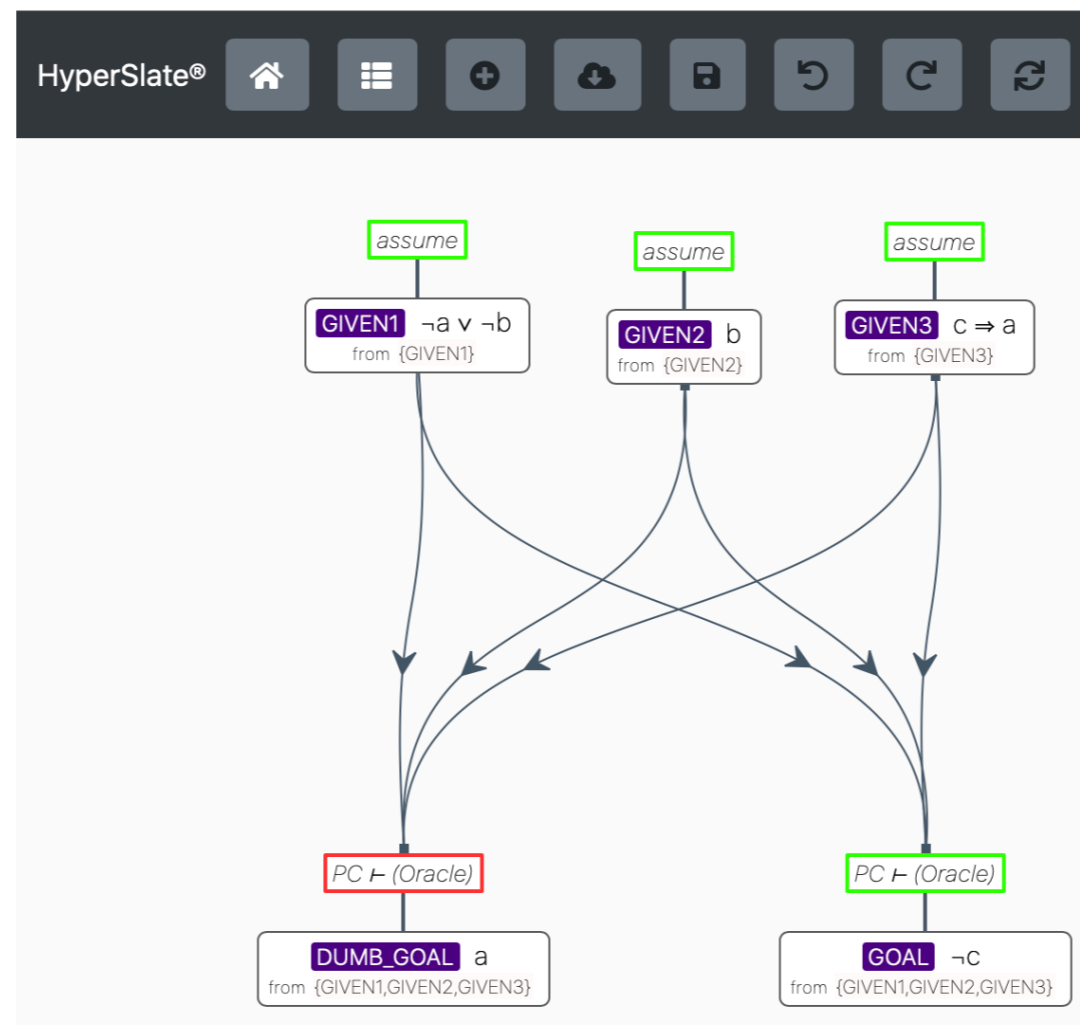
Our First Rule of Inference (= Inference Schema): PC (Entailment) Oracle



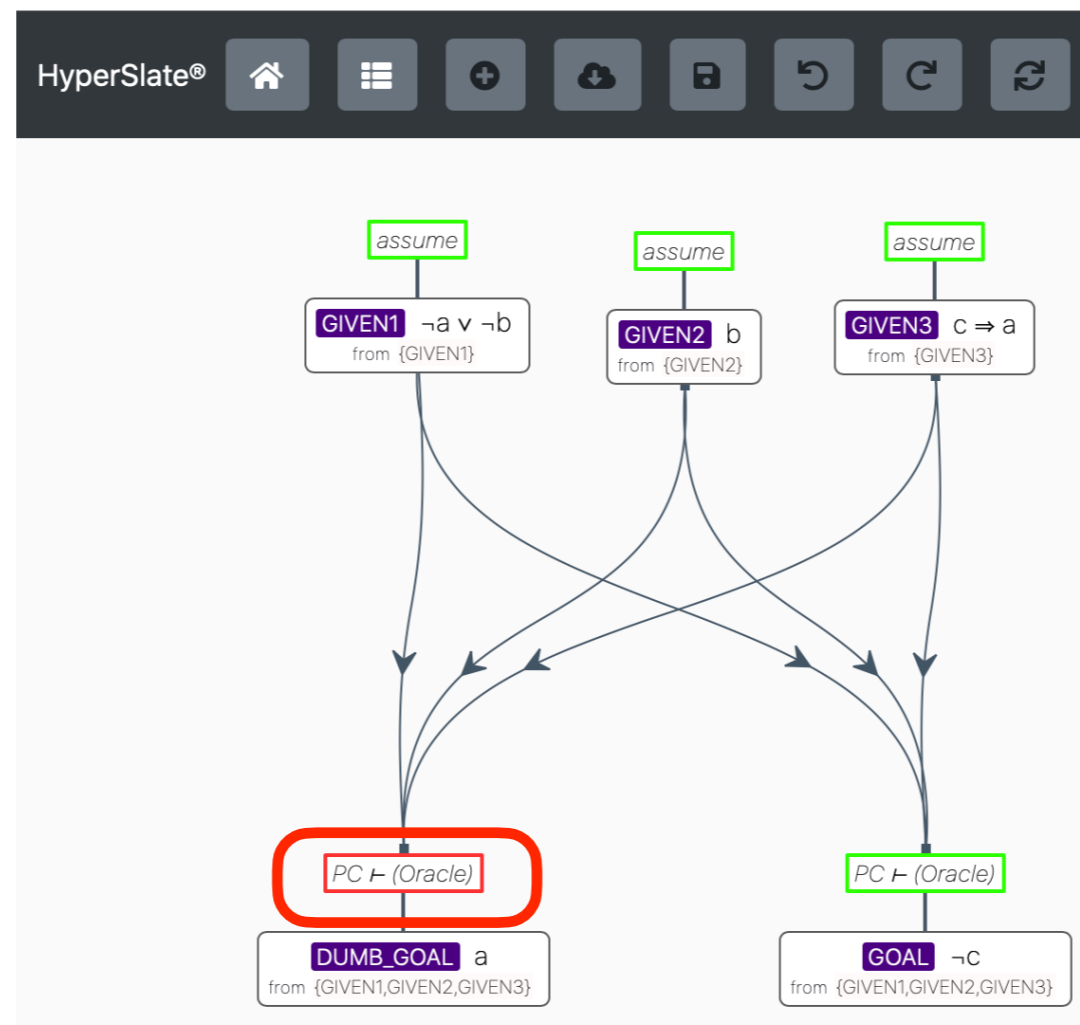
Our First Rule of Inference (= Inference Schema): PC (Entailment) Oracle



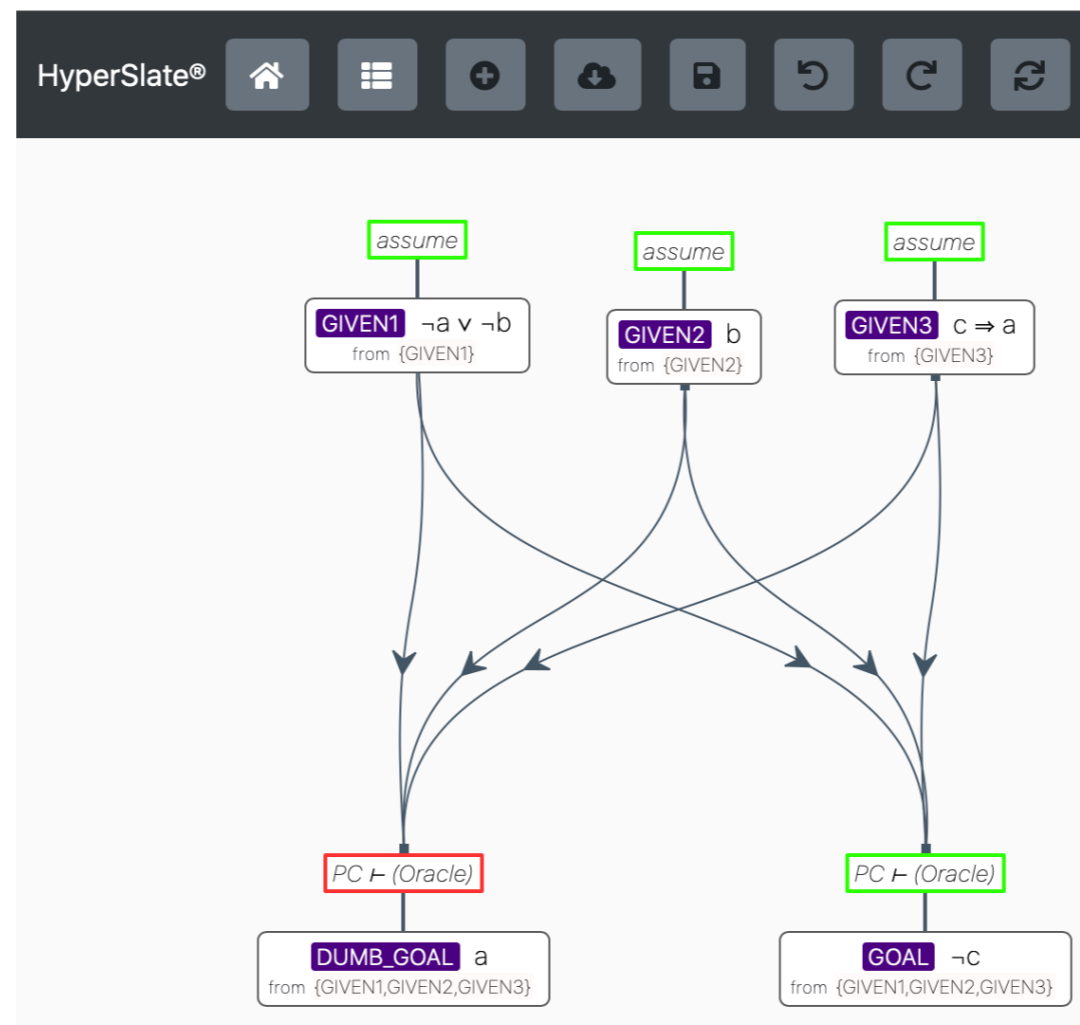
Our First Rule of Inference (= Inference Schema): PC (Entailment) Oracle



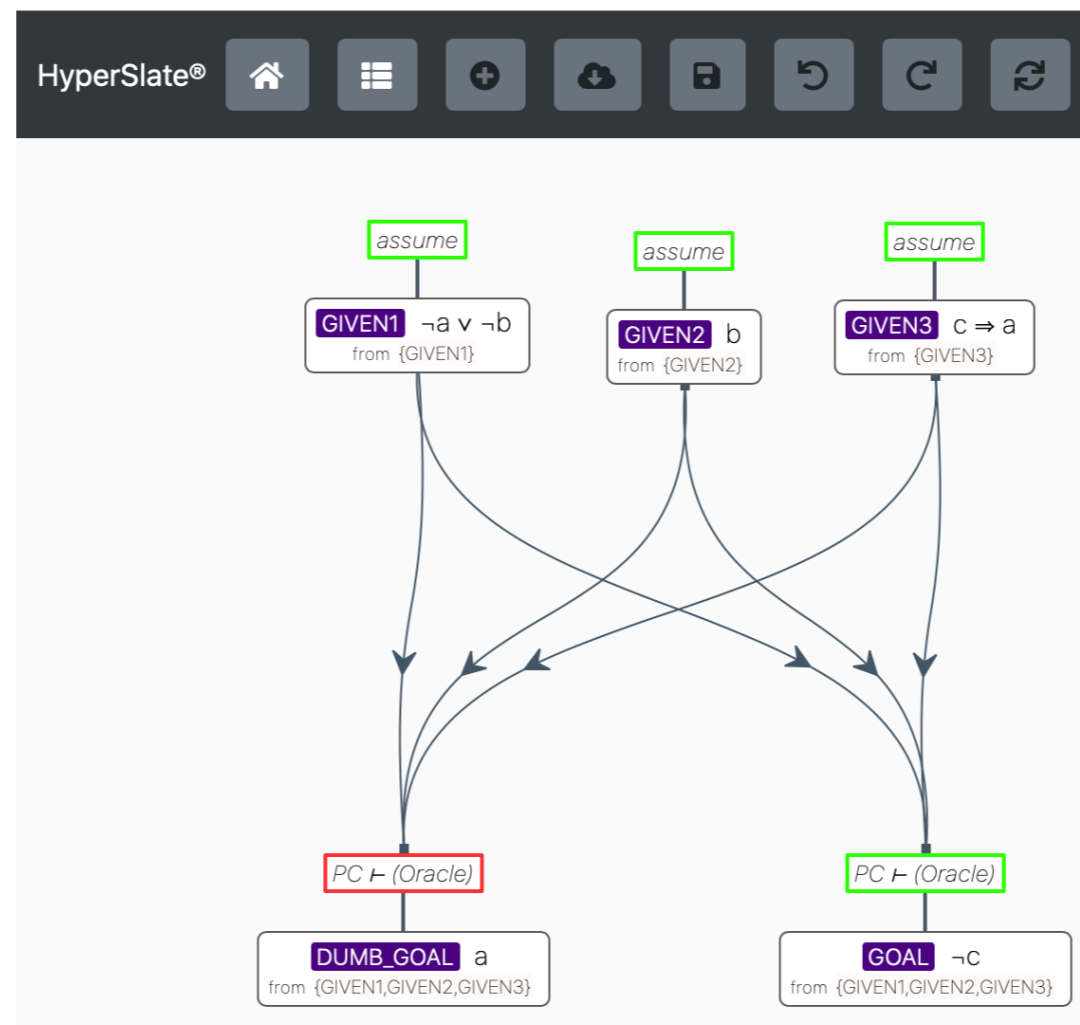
Our First Rule of Inference (= Inference Schema): PC (Entailment) Oracle



Our First Rule of Inference (= Inference Schema): PC (Entailment) Oracle



Our First Rule of Inference (= Inference Schema): PC (Entailment) Oracle



“NYS 3” Revisited

Given the statements

$\neg \neg c$

$c \rightarrow a$

$\neg a \vee b$

$b \rightarrow d$

$\neg(d \vee e)$

which one of the following statements are provable?

$\neg c$

e

h

$\neg a$

all of the above

“NYS 3” Revisited

Given the statements

$\neg\neg c$

$c \rightarrow a$

$\neg a \vee b$

$b \rightarrow d$

$\neg(d \vee e)$

which one of the following statements are provable?

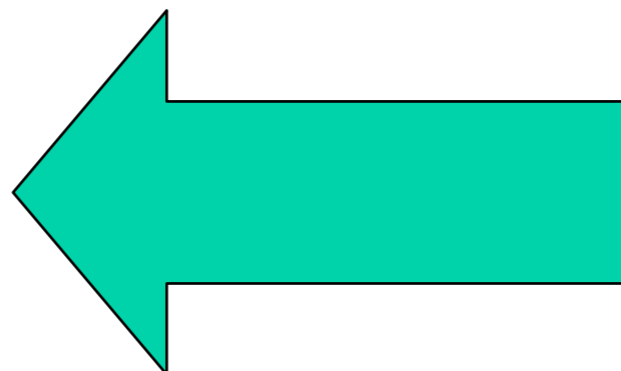
$\neg c$

e

h

$\neg a$

all of the above



“NYS 3” Revisited

Given the statements

$\neg\neg c$

$c \rightarrow a$

$\neg a \vee b$

$b \rightarrow d$

$\neg(d \vee e)$

Show in HyperSlate[®] that each of the first four options can be proved using the PC entailment oracle.

which one of the following statements are provable?

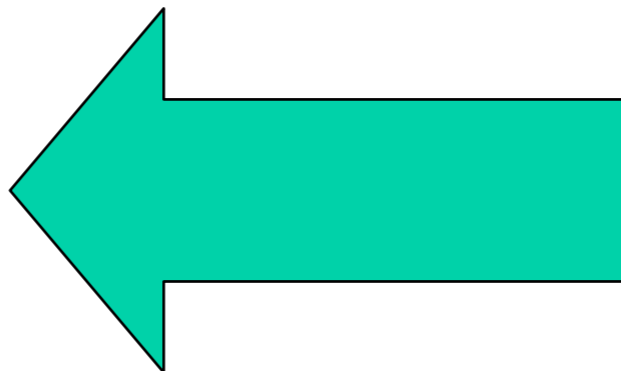
$\neg c$

e

h

$\neg a$

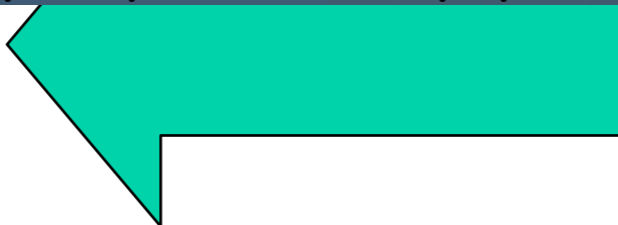
all of the above



“NYS 3” Revisited

The screenshot shows a web browser window with the URL `rpi.logicamodernapproach.com`. The browser's address bar and tabs are visible. The main content area displays a logical proof tree in HyperSlate. The tree starts with a root node `GIVEN1` containing the formula $\neg\neg c$ and the text "from {GIVEN1}". A green box labeled "assume" is positioned above this node. The tree branches into two paths. The left path uses a `PC ⊢ (Oracle)` node to derive `8 c` from `GIVEN1`. The right path uses an `assume` node to derive `GIVEN2` containing $c \Rightarrow a$ from `GIVEN2`. These two paths converge at an `⇒ elim` node, which derives `9 a` from the set `{GIVEN1, GIVEN2}`. From `9 a`, the tree branches again. The left path uses another `PC ⊢ (Oracle)` node to derive `10 b` from `{GIVEN1, GIVEN2, GIVEN3}`. The right path uses an `assume` node to derive `GIVEN3` containing $\neg a \vee b$ from `GIVEN3`. These two paths converge at a final `⇒ elim` node, which derives `11 d` from `{GIVEN1, GIVEN2, GIVEN3, GIVEN4}`. To the right of the main tree, there is a separate branch starting with an `assume` node above `GIVEN5` containing $\neg(d \vee e)$ from `GIVEN5`. Below this, a `PC ⊢ (Oracle)` node is shown above a box containing `CRAZY_GOAL = OPTION 3` and `h` from `{}`. The HyperSlate interface includes a toolbar with icons for home, list, add, save, undo, redo, and Bezier. A status bar at the top right indicates "NYS3OracleOnly [PROPOSITIONAL-CALCULUS]: Saved with 22 symbols." The macOS dock is visible at the bottom of the browser window.

all of the above



Det er en ære å lære formell logikk!

Det er en ære å lære formell logikk!

Part II of Class