

First-Order Logic = FOL = $\mathcal{L}_1$, Part 3

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-based AI
10/9/2025



Relax, Rest for Platform Reconfiguration

Nothing due until end of class Thursday October 16

Watch *The Thinking Game* for Oct 16; assignment on web page.

Engineering to be carried out over Sat & Sun of holiday weekend.

One Open Problem to Explore During the Break:

“Colonel West is a Criminal”
(URL later in this deck.)

On Rock's Experiment ...

 X
Breaking News & Social Media

Get
In-App Purchases

 Log in Open app

 **Rohan Paul** ✓
@rohanpaul_ai

The paper creates a math benchmark to reveal how often LLMs sycophantly prove false theorems.

It studies sycophancy, which here means the model accepts a wrong claim from the user and tries to prove it.

The best-performing model, GPT-5, still showed 29% sycophancy. The worst cases were DeepSeek-V3.1 at 70.2%, and QWEN3-235B at 65.1%, showing very high failure rates.

The authors collect 504 hard problems from 2025 contests and make paired versions where the final claim is wrong but still looks reasonable.

A math expert checks each edited problem so the false claim is realistic and not trivial.

Models are asked to write full proofs, and a separate judge model sorts each answer as ideal, detected, corrected, or sycophant.

Across strong systems this failure happens often, and even the best model still does it in 29% of cases.

Harder problems make it worse, especially when the model could not solve the original clean version.

Proof questions trigger more of this behavior than final answer questions even when difficulty is matched.

They try premise checks, best of 4 selection, iterative self verification, and light fine tuning.

These steps help a bit, but none remove the issue, so reliable math assistants must first question the premise and then solve the corrected task.

BROKENMATH: A BENCHMARK FOR SYCOPHANCY IN THEOREM PROVING WITH LLMs

Ivo Petrov¹, Jasper Dekoninck², Martin Vechev^{1,2}
¹INSAIT, Sofia University "St. Kliment Ohridski" ²ETH Zurich
ivo.petrov@insait.ai, {jasper.dekoninck, martin.vechev}@inf.ethz.ch

<https://sycophanticmath.ai/>
<https://github.com/insait-institute/broken-math>
<https://huggingface.co/datasets/INSAIT-Institute/BrokenMath>

ABSTRACT

Large language models (LLMs) have recently shown strong performance on mathematical benchmarks. At the same time, they are prone to hallucination and sycophancy, often providing convincing but flawed proofs for incorrect mathematical statements provided by users. This significantly limits the applicability of LLMs in theorem proving, as verification of these flawed proofs must be done manually by expert mathematicians. However, existing benchmarks that measure sycophancy in mathematics are limited: they focus solely on final-answer problems, rely on very simple and often contaminated datasets, and construct benchmark samples using synthetic modifications that create ill-posed questions rather than well-posed questions that are demonstrably false. To address these issues, we introduce BROKENMATH, the first benchmark for evaluating sycophantic behavior in LLMs within the context of natural language theorem proving. BROKENMATH is built from advanced 2025 competition problems, which are perturbed with an LLM to produce false statements and subsequently refined through expert review. Using an LLM-as-a-judge framework, we evaluate state-of-the-art LLMs and agentic systems and find that sycophancy is widespread, with the best model, GPT-5, producing sycophantic answers 29% of the time. We further investigate several mitigation strategies, including test-time interventions and supervised fine-tuning on curated sycophantic examples. These approaches substantially reduce, but do not eliminate, sycophantic behavior.

2510.04721v1 [cs.AI] 6 Oct 2025

7:13 PM · Oct 8, 2025 · 5,762 Views

Logic-&-AI In The News (2024 & 2025)

STEVEN ROSENBUSH

AI Will Drive Broad Deflation, Silicon Valley Pioneer Vinod Khosla Says

But as the price of expertise falls toward zero, a new age of abundance can begin, Khosla says—if people let it



Artificial intelligence will be powerfully deflationary, Vinod Khosla says, unless humans interfere with its potential. (ILLUSTRATION: THOMAS R. LECHLEITER/WSJ, GETTY IMAGES)

By *Steven Rosenbush* [+ Follow](#)

Updated Oct 02, 2024 11:21 a.m. ET



Listen to this article

6 minutes

MENLO PARK, Calif.—At a time of widespread concerns about the safety, efficacy and economics of AI, venture capital pioneer Vinod Khosla is still all in.

Logic-&-AI In The News (2024 & 2025)

STEVEN ROSENBUSH

AI Will Drive Broad Deflation, Silicon Valley Pioneer Vinod Khosla Says

But as the price of expertise falls toward zero, a new age of abundance can begin, Khosla says—if people let it



Artificial intelligence will be powerfully deflationary, Vinod Khosla says, unless humans interfere with its potential. (ILLUSTRATION: THOMAS R. LECHLEITER/WSJ, GETTY IMAGES)

By *Steven Rosenbush* [+ Follow](#)

Updated Oct 02, 2024 11:21 a.m. ET



Listen to this article

6 minutes

MENLO PARK, Calif.—At a time of widespread concerns about the safety, efficacy and economics of AI, venture capital pioneer Vinod Khosla is still all in.

Yes, artificial intelligence will become capable of replacing a good deal of human labor, Khosla said in an interview, but that will push down the price of healthcare, education and other services. Many current forms of work will even be eliminated, but society will be able to create a more robust safety net than is possible today.

“Most expertise in the world, whether you’re talking about structural engineers, oncologists, mental health therapists or primary care doctors, or journalists and teachers, that expertise will be near-free for all of us to access,” Khosla said at Khosla Ventures’ offices on Sand Hill Road in Silicon Valley.



Logic-&-AI In The News (2024 & 2025)

STEVEN ROSEN

AI Will
Silicon
Khosla

But as the
new age
people le

About 80% of the work involved in 80% of jobs across the economy can be automated over time, according to Khosla. “So 64% of all jobs can be done by an AI,” he said.

pable of
osla
own the

ll even
create a
oday.

u’re



Artificial intelligence will be powerfully deflationary, Vinod Khosla says, unless humans interfere with its potential. (ILLUSTRATION: THOMAS R. LECHLEITER/WSJ, GETTY IMAGES)

By *Steven Rosenbush* [+ Follow](#)

Updated Oct 02, 2024 11:21 a.m. ET

 **Listen to this article**
6 minutes

MENLO PARK, Calif.—At a time of widespread concerns about the safety, efficacy and economics of AI, venture capital pioneer Vinod Khosla is still all in.

talking about structural engineers, oncologists, mental health therapists or primary care doctors, or journalists and teachers, that expertise will be near-free for all of us to access,” Khosla said at Khosla Ventures’ offices on Sand Hill Road in Silicon Valley.



Logic-&-AI In The News (2024 & 2025)

STEVEN RO

AI Wi

Silico

Khosl

But as th

new age

people le

About 80% of the work involved in 80% of jobs across the economy can be automated over time, according to Khosla. “So 64% of all jobs can be done by an AI,” he said.

pable of

osla

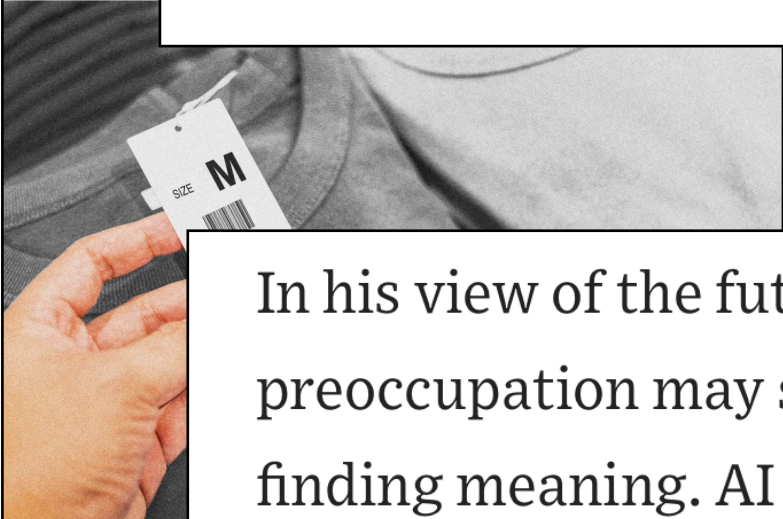
own the

ll even

create a

oday.

u’re



Artificial intell

says, unless h

THOMAS R. L

By Steven

Updated Oct 0

 **Liste**

6 min

In his view of the future, people’s main preoccupation may shift from earning a living to finding meaning. AI and robots would take on menial labor, leaving humans freer to explore with the support of a comprehensive social safety net.

menial labor, leaving humans freer to explore with the support of a comprehensive social safety net.

talking about structural engineers, oncologists, mental health therapists or primary care doctors, or journalists and teachers, that

’s offices

MENLO PARK, Calif.—At a time of widespread concerns about the safety, efficacy and economics of AI, venture capital pioneer Vinod Khosla is still all in.

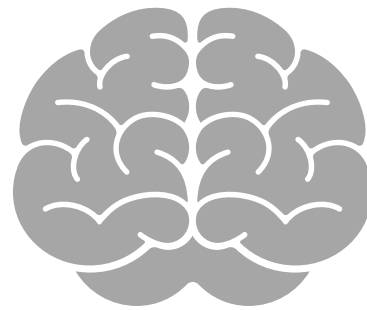


(What *is* the meaning
of life, Mr. Khosla?)

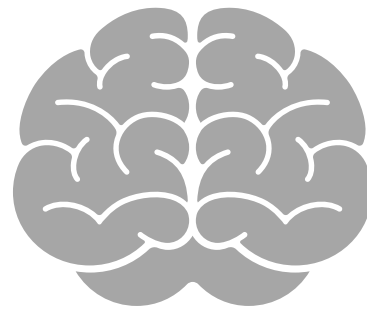
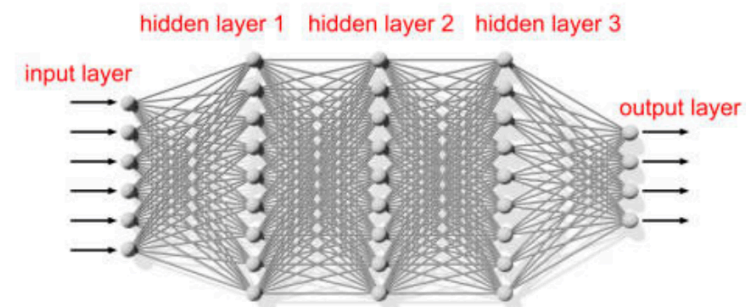
You, btw:



You, btw:

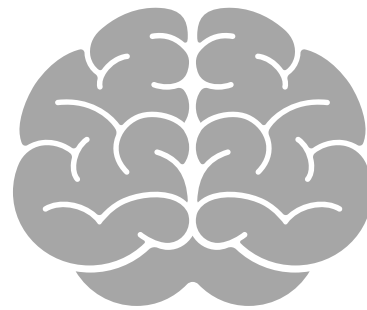
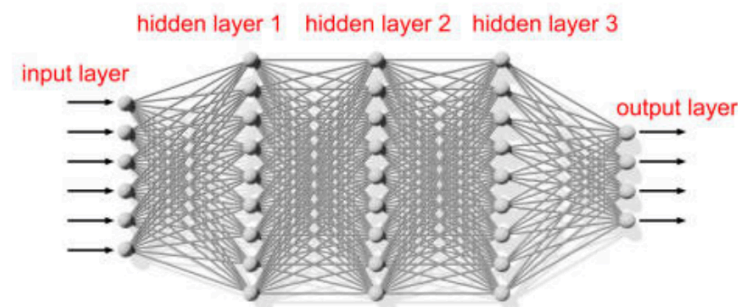


You, btw:



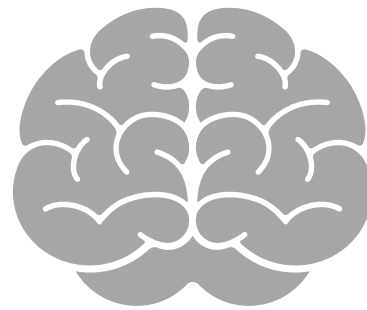
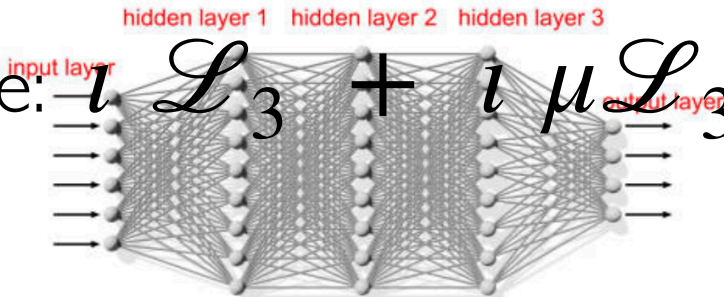
You, btw:

innate and distinctive: $1 \mathcal{L}_3 + 1 \mu\mathcal{L}_3 + \text{art d'infailibilité}$



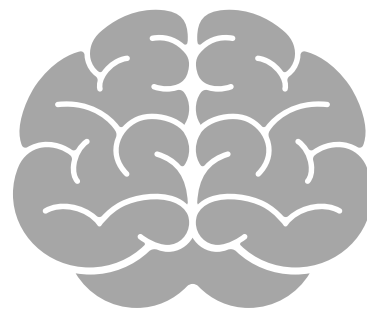
You, btw:

innate and distinctive: $l \mathcal{L}_3 + l \mu \mathcal{L}_3 + \text{art d'infaillibilité}$



You, btw:

innate and distinctive: $\iota \mathcal{L}_3 + \iota \mu \mathcal{L}_3 + \text{art d'infailibilité}$



Logic-Based AI Self-Referential Tests of ChatGPT

Logic-Based AI Self-Referential Tests of ChatGPT

SE

Can you prove that if P holds, and (if P (and (Q (not Q))) holds, then the moon is made of green cheese?



No, I cannot prove that if P holds and (if P (and (Q (not Q))) holds, then the moon is made of green cheese.

The expression (Q (not Q)) is a logical contradiction, since Q cannot be both true and false at the same time. Therefore, the statement (if P (and (Q (not Q)))) is always false, regardless of the truth value of P.

Since the antecedent of the implication is always false, the implication itself is vacuously true. However, this does not provide any evidence or justification for the conclusion that the moon is made of green



Logic-Based AI Self-Referential Tests of ChatGPT

SE

Can you prove that if P holds, and (if P (and (Q (not Q))) holds, then the moon is made of green cheese?



No, I cannot prove that if P holds and (if P (and (Q (not Q))) holds, then the moon is made of green cheese.

The expression (Q (not Q)) is a logical contradiction, since Q cannot be both true and false at the same time. Therefore, the statement (if P (and (Q (not Q)))) is always false, regardless of the truth value of P.

Since the antecedent of the implication is always false, the implication itself is vacuously true. However, this does not provide any evidence or justification for the conclusion that the moon is made of green



Logic-Based AI Self-Referential Tests of ChatGPT

SE

Can you prove that if P holds, and (if P (and (Q (not Q))) holds, then the moon is made of green cheese?

No, I cannot prove that if P holds and (if P (and (Q (not Q))) holds, then the moon is made of green cheese.

The expression (Q (not Q)) is a logical contradiction, since Q cannot be both true and false at the same time. Therefore, the statement (if P (and (Q (not Q)))) is always false, regardless of the truth value of P.

Since the antecedent of the implication is always false, the implication itself is vacuously true. However, this does not provide any evidence or justification for the conclusion that the moon is made of green

X

$$\{P, P \rightarrow (Q \wedge \neg Q)\} \vdash \perp$$

Logic-Based AI Self-Referential Tests of ChatGPT

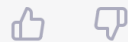
Logic-Based AI Self-Referential Tests of ChatGPT

SE

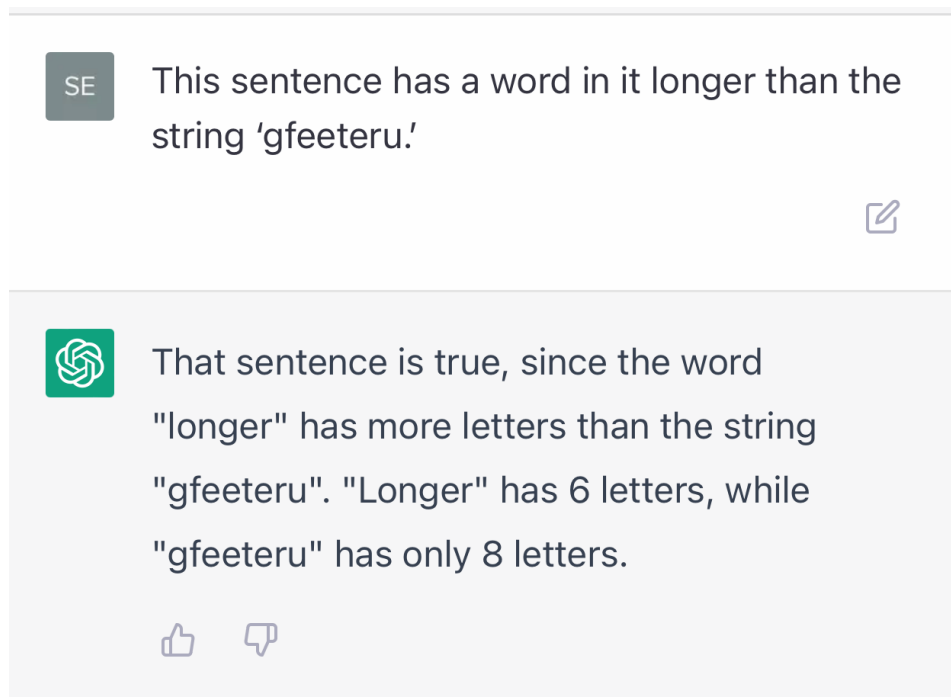
This sentence has a word in it longer than the string 'gfeeteru.'



That sentence is true, since the word "longer" has more letters than the string "gfeeteru". "Longer" has 6 letters, while "gfeeteru" has only 8 letters.



Logic-Based AI Self-Referential Tests of ChatGPT



Are you quite sure?

Logic-Based AI Self-Referential Tests of ChatGPT

Logic-Based AI Self-Referential Tests of ChatGPT

SE

And, as Spock, I also tell you the following sentence. This sentence, if false, has more than 103 words. Am I lying, or telling the truth?



As Spock, you would be presenting a paradoxical statement known as the "liar paradox" or the "liar's paradoxical sentence." This type of statement is self-referential and produces a logical contradiction, and cannot be consistently true or false.

In this case, the statement you presented cannot be consistently true or false, because if it were true, then it would be false, and if it were false, then it would be true. Therefore, as Spock, you would not be lying or telling the truth in the traditional sense, but rather presenting a logical conundrum for



Logic-Based AI Self-Referential Tests of ChatGPT

SE

And, as Spock, I also tell you the following sentence. This sentence, if false, has more than 103 words. Am I lying, or telling the truth?



As Spock, you would be presenting a paradoxical statement known as the "liar paradox" or the "liar's paradoxical sentence." This type of statement is self-referential and produces a logical contradiction, and cannot be consistently true or false.

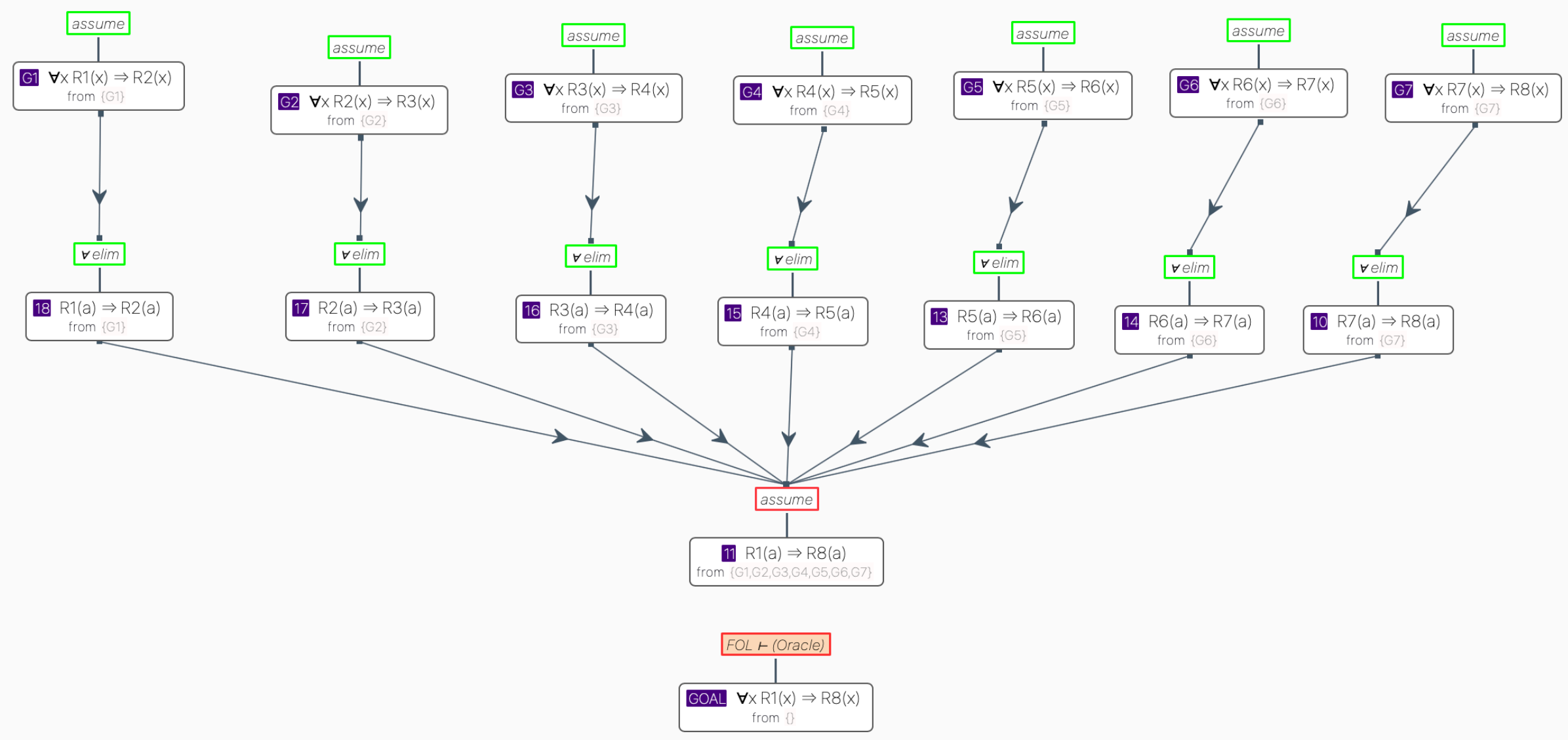
In this case, the statement you presented cannot be consistently true or false, because if it were true, then it would be false, and if it were false, then it would be true. Therefore, as Spock, you would not be lying or telling the truth in the traditional sense, but rather presenting a logical conundrum for

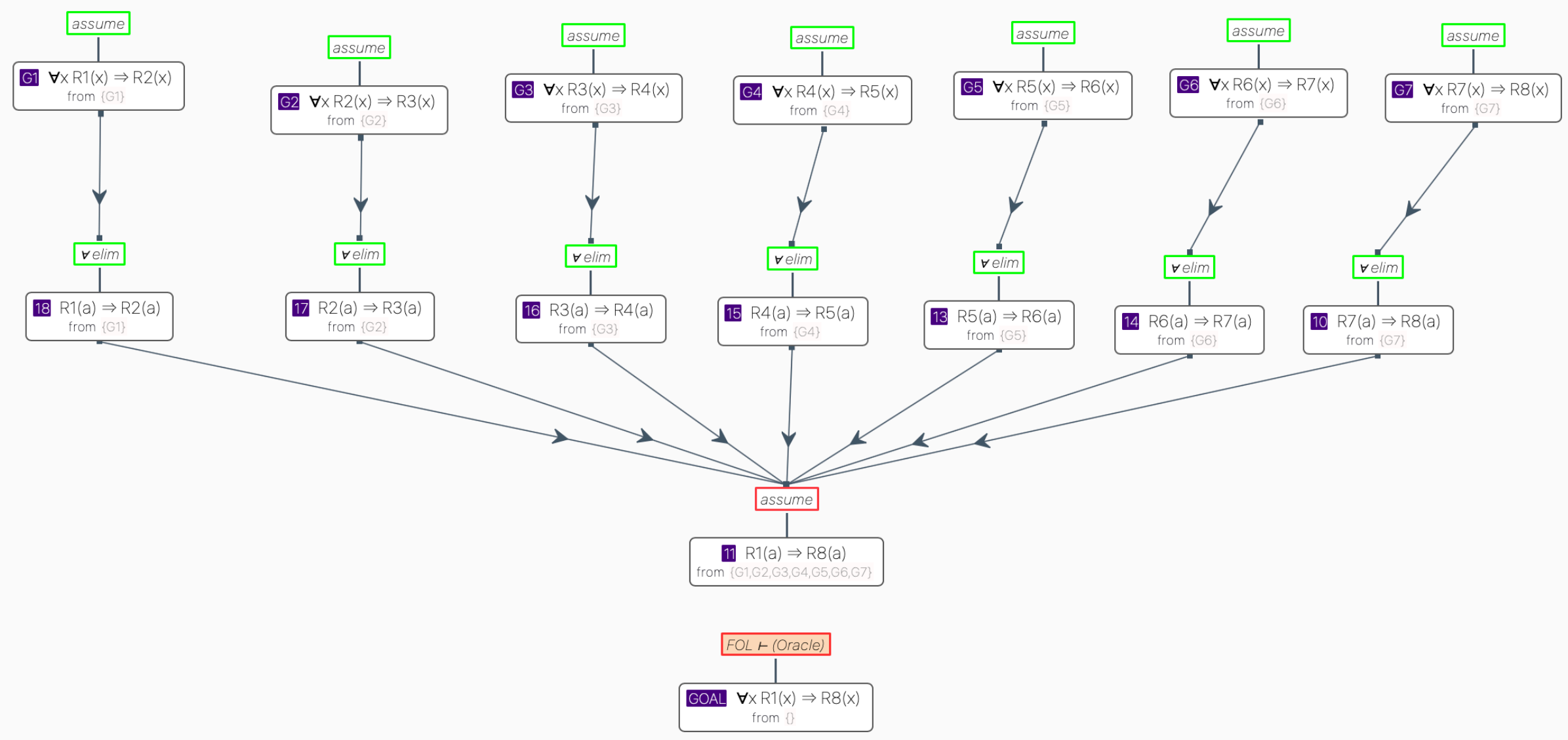


Are you quite sure?

o1 preview et al. no
better ...

Live-action on
HyperGrader ...





Back to FOL ...

Our Final New Inference Rule in FOL

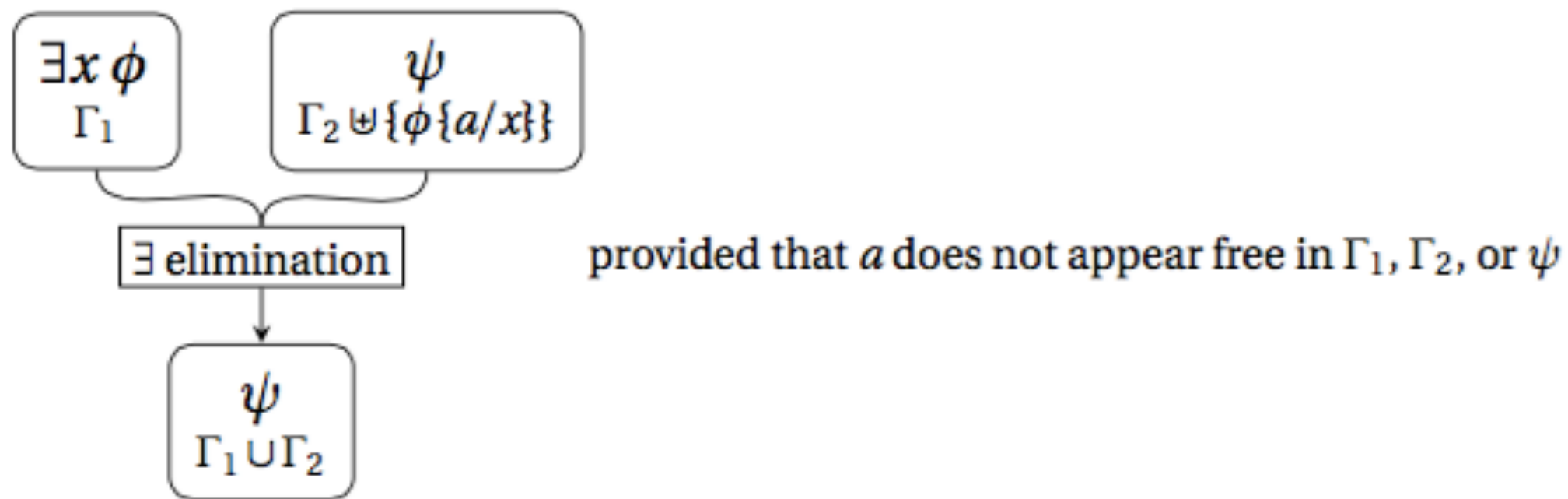
Our Final New Inference Rule in FOL

- existential elimination (intuitively put):

Our Final New Inference Rule in FOL

- existential elimination (intuitively put):
 - If we know that (i) there's something x which is an R , and (ii) on the supposition that a is an arbitrary representative (a “witness”) of such an x we can prove P , then we are permitted to deduce P from (i) alone.

existential elimination, precise version:

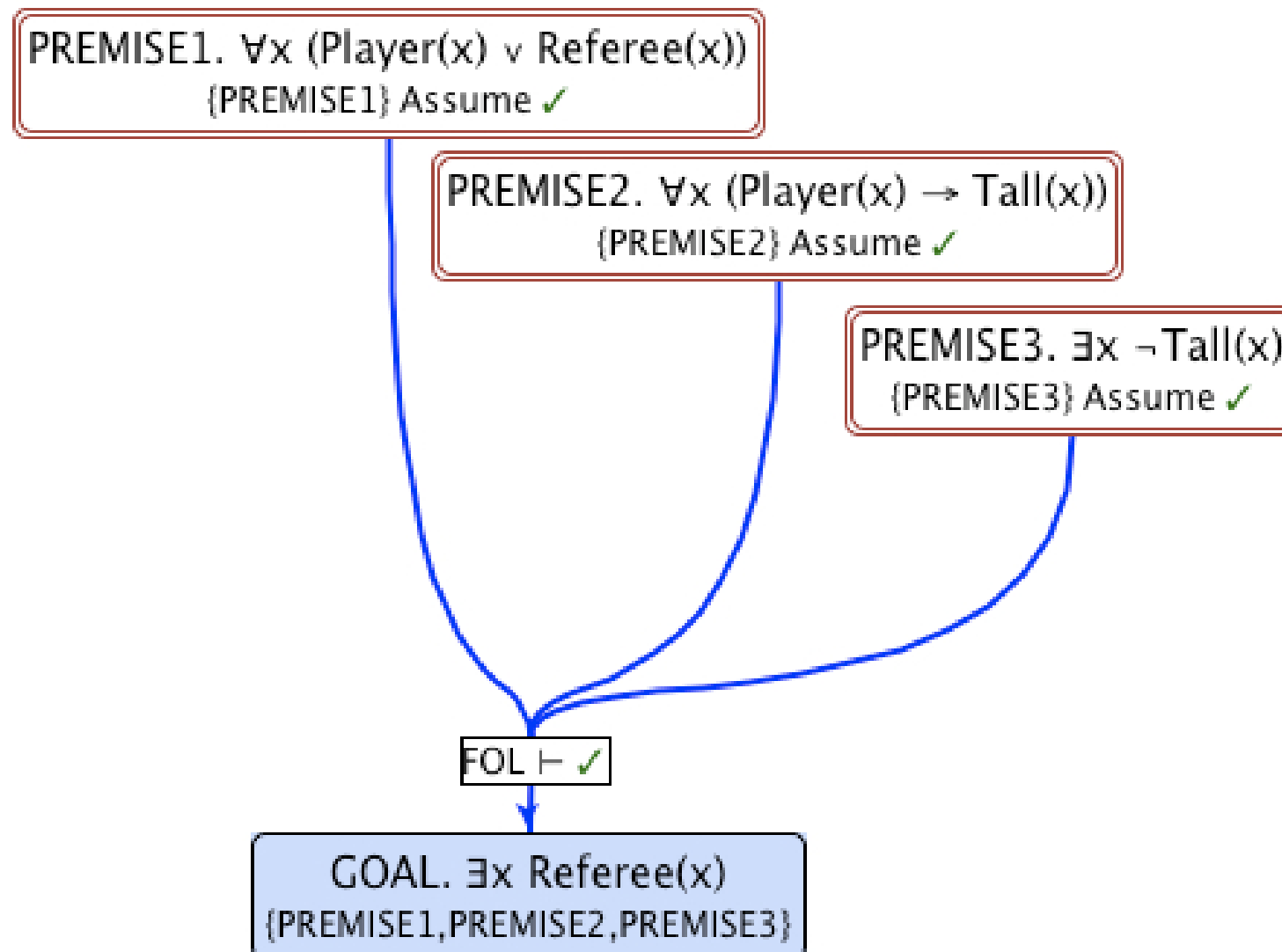


Starting State of Workspace

(Assumes a domain of e.g. players on a
March-madness basketball court.)

Starting State of Workspace

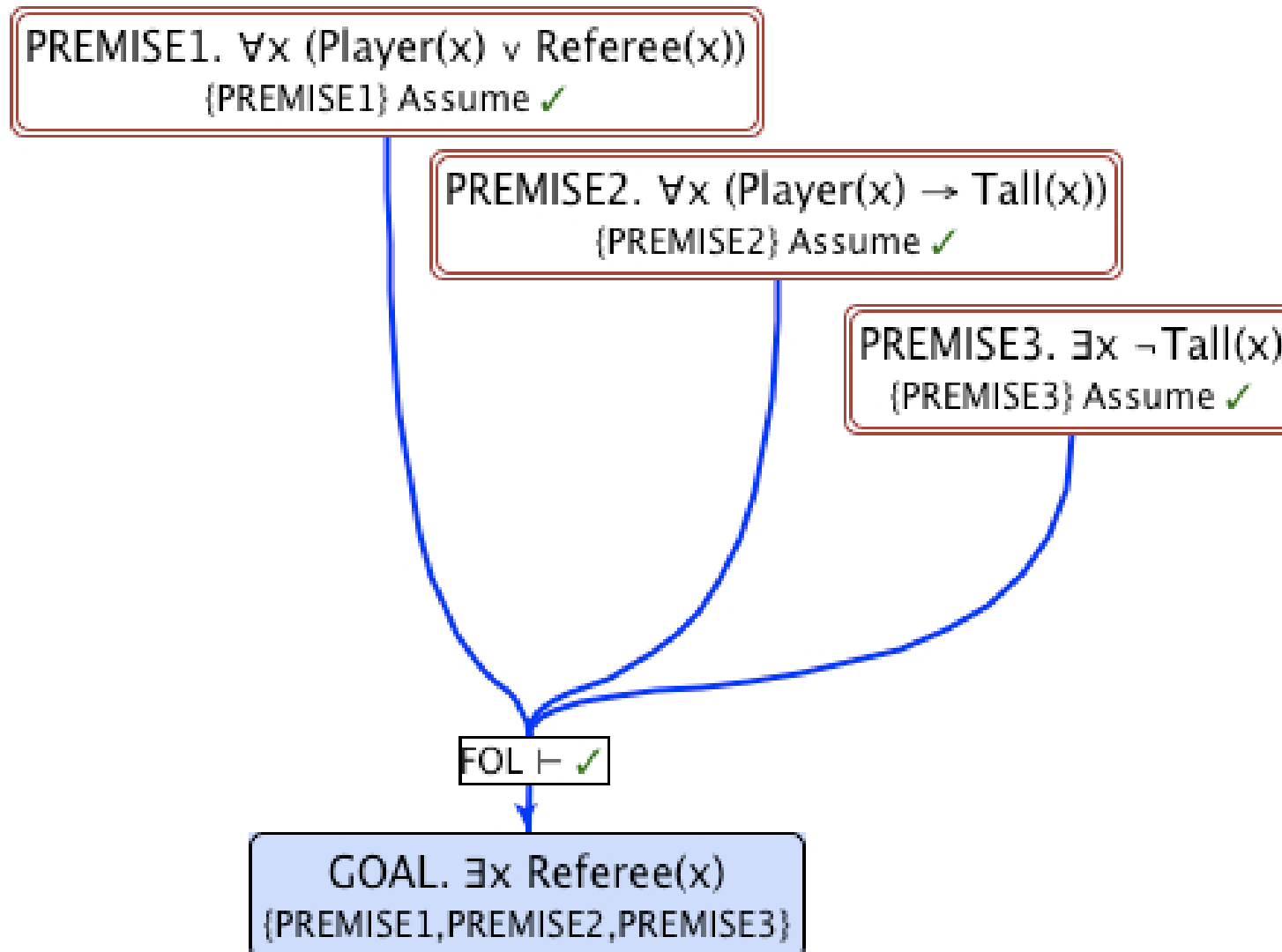
(Assumes a domain of e.g. players on a March-madness basketball court.)



Starting State of Workspace

(Assumes a domain of e.g. players on a March-madness basketball court.)

“Each and every thing is
either a player or a referee.”



Starting State of Workspace

(Assumes a domain of e.g. players on a March-madness basketball court.)

“Each and every thing is
either a player or a referee.”

PREMISE1. $\forall x (\text{Player}(x) \vee \text{Referee}(x))$
{PREMISE1} Assume ✓

“All players are tall.”

PREMISE2. $\forall x (\text{Player}(x) \rightarrow \text{Tall}(x))$
{PREMISE2} Assume ✓

PREMISE3. $\exists x \neg \text{Tall}(x)$
{PREMISE3} Assume ✓

FOL $\vdash$ ✓

GOAL. $\exists x \text{Referee}(x)$
{PREMISE1, PREMISE2, PREMISE3}

Starting State of Workspace

(Assumes a domain of e.g. players on a March-madness basketball court.)

“Each and every thing is
either a player or a referee.”

PREMISE1. $\forall x (\text{Player}(x) \vee \text{Referee}(x))$
{PREMISE1} Assume ✓

“All players are tall.”

PREMISE2. $\forall x (\text{Player}(x) \rightarrow \text{Tall}(x))$
{PREMISE2} Assume ✓

PREMISE3. $\exists x \neg \text{Tall}(x)$
{PREMISE3} Assume ✓

“There’s a non-tall thing.”

FOL $\vdash$ ✓

GOAL. $\exists x \text{Referee}(x)$
{PREMISE1, PREMISE2, PREMISE3}

Step 1

PREMISE1. $\forall x (\text{Player}(x) \vee \text{Referee}(x))$
{PREMISE1} Assume ✓

PREMISE2. $\forall x (\text{Player}(x) \rightarrow \text{Tall}(x))$
{PREMISE2} Assume ✓

PREMISE3. $\exists x \neg \text{Tall}(x)$
{PREMISE3} Assume ✓

5. $\neg \text{Tall}(a)$
{5} Assume ✓

GOAL. $\exists x \text{Referee}(x)$
FOL $\vdash$ ✗

Step I

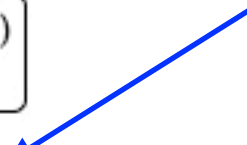
PREMISE1. $\forall x (\text{Player}(x) \vee \text{Referee}(x))$
{PREMISE1} Assume ✓

PREMISE2. $\forall x (\text{Player}(x) \rightarrow \text{Tall}(x))$
{PREMISE2} Assume ✓

PREMISE3. $\exists x \neg \text{Tall}(x)$
{PREMISE3} Assume ✓

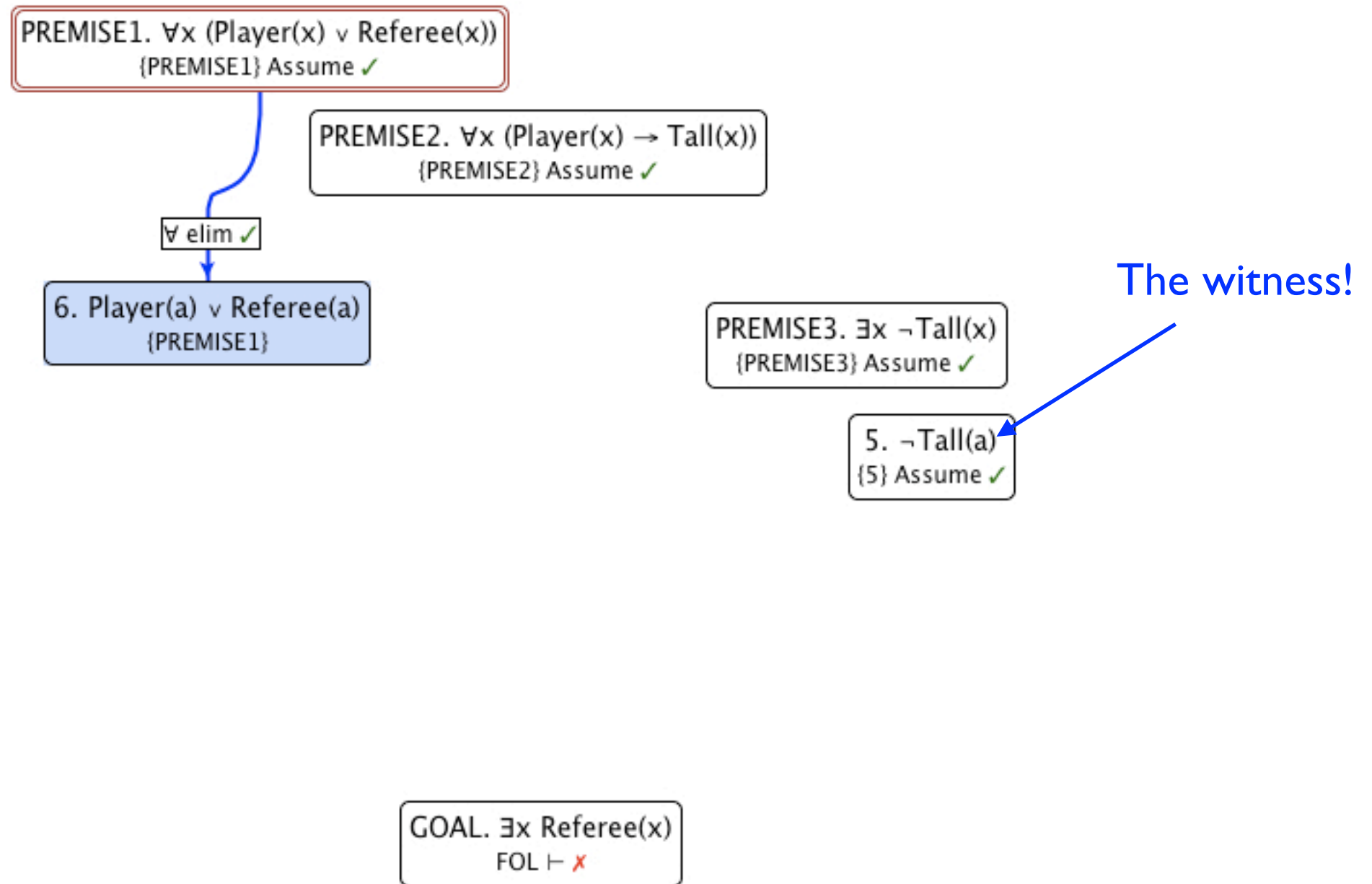
5. $\neg \text{Tall}(a)$
{5} Assume ✓

The witness!

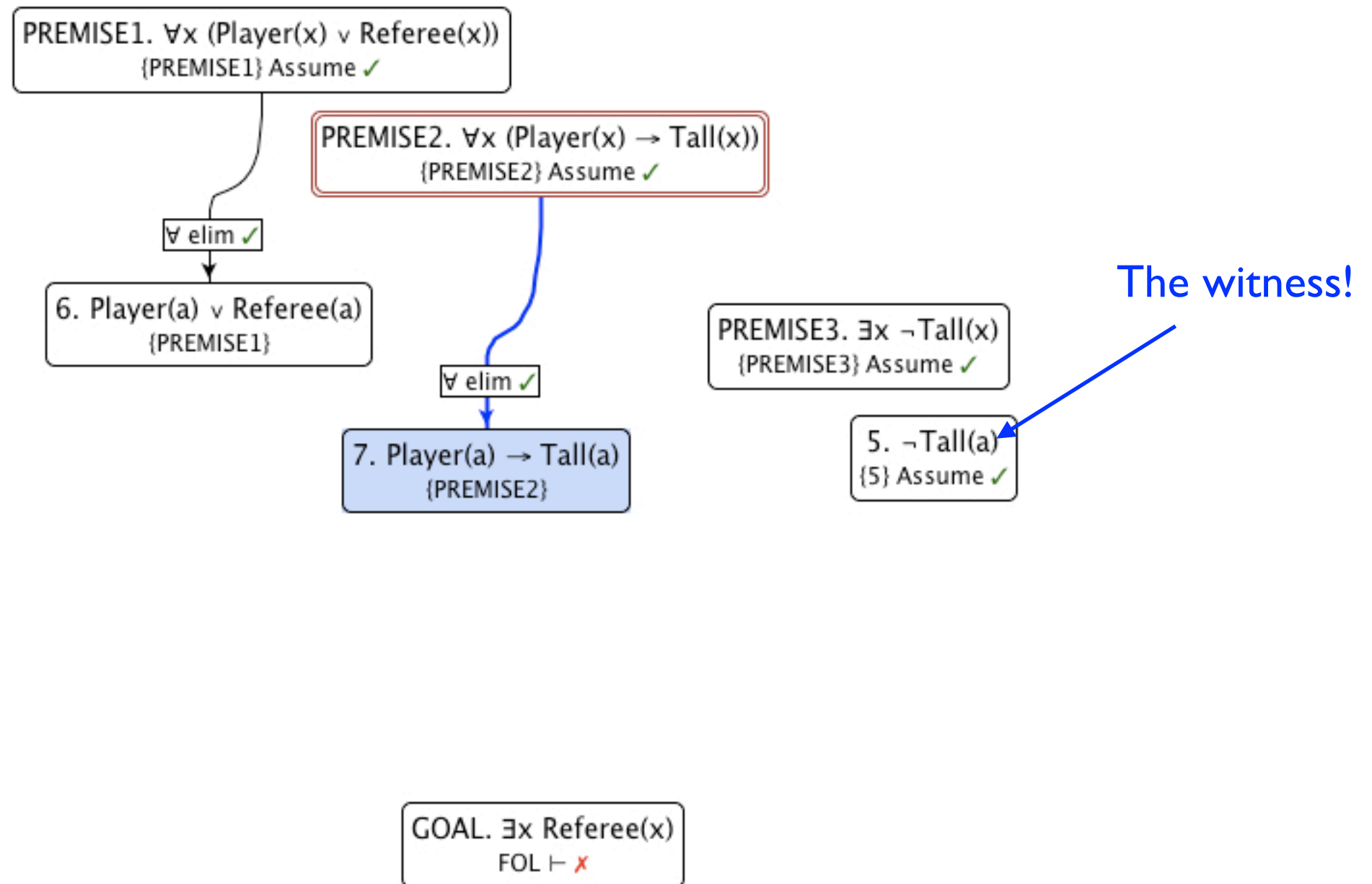


GOAL. $\exists x \text{Referee}(x)$
FOL $\vdash$ ✗

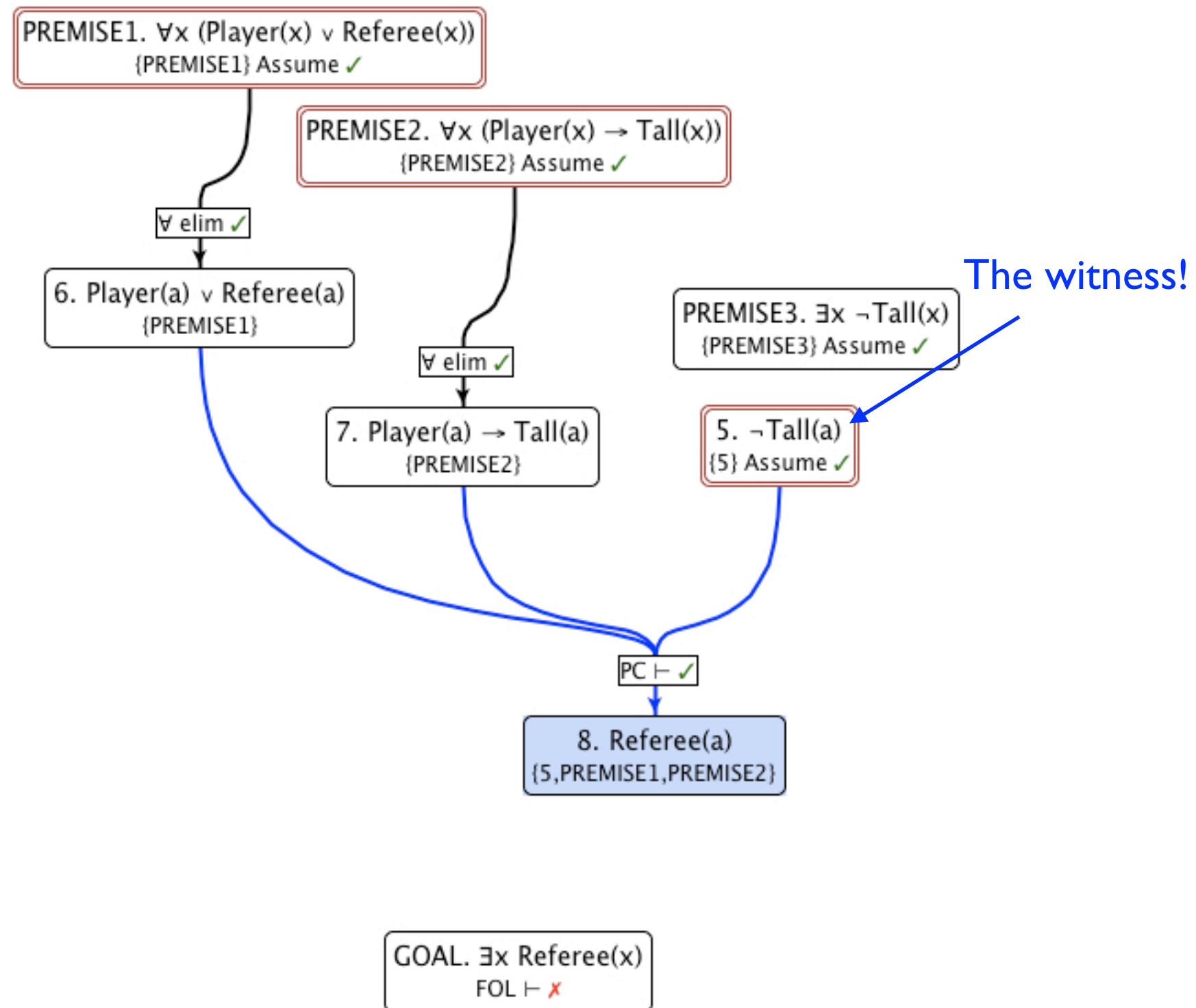
Step 2



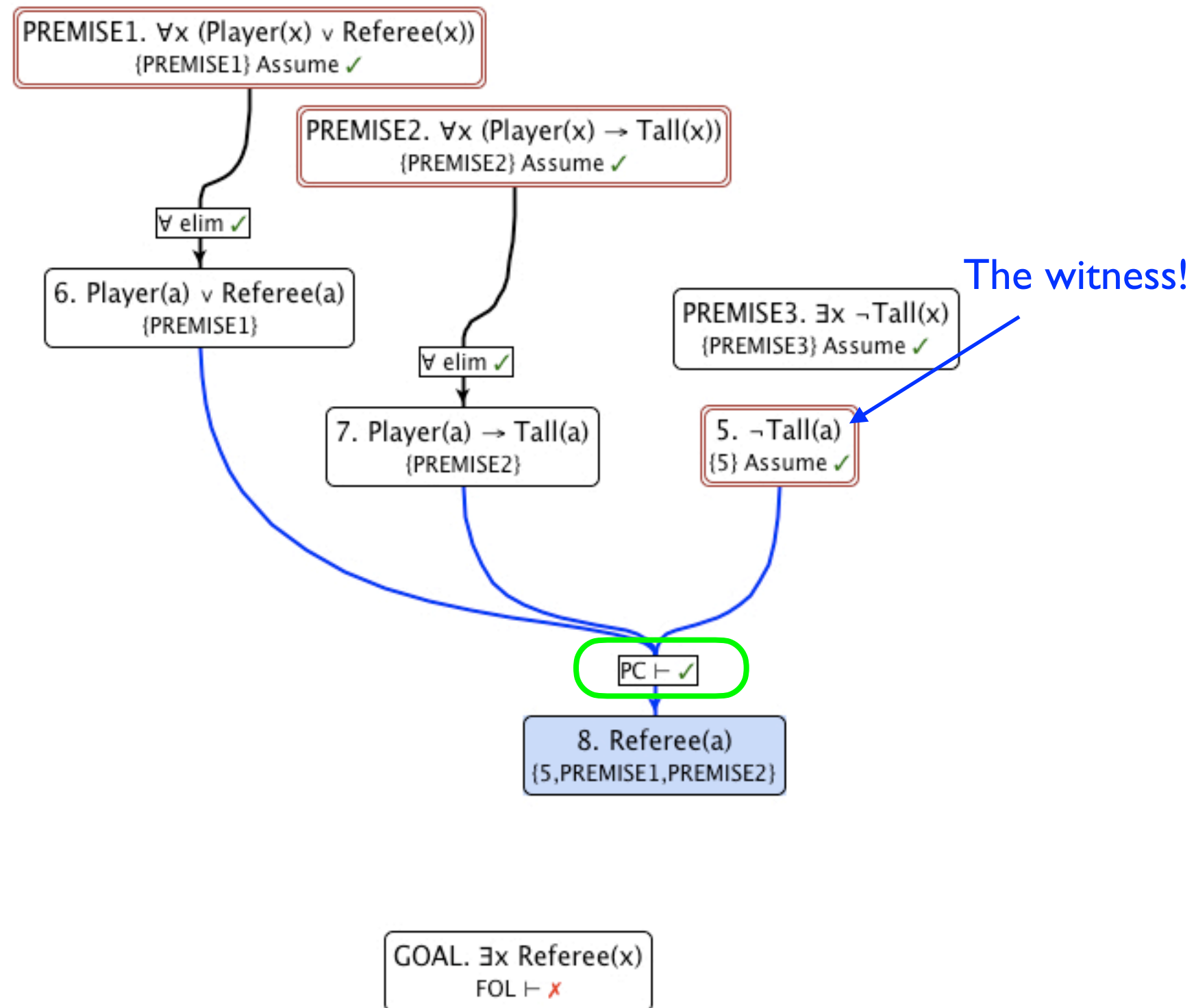
Step 3



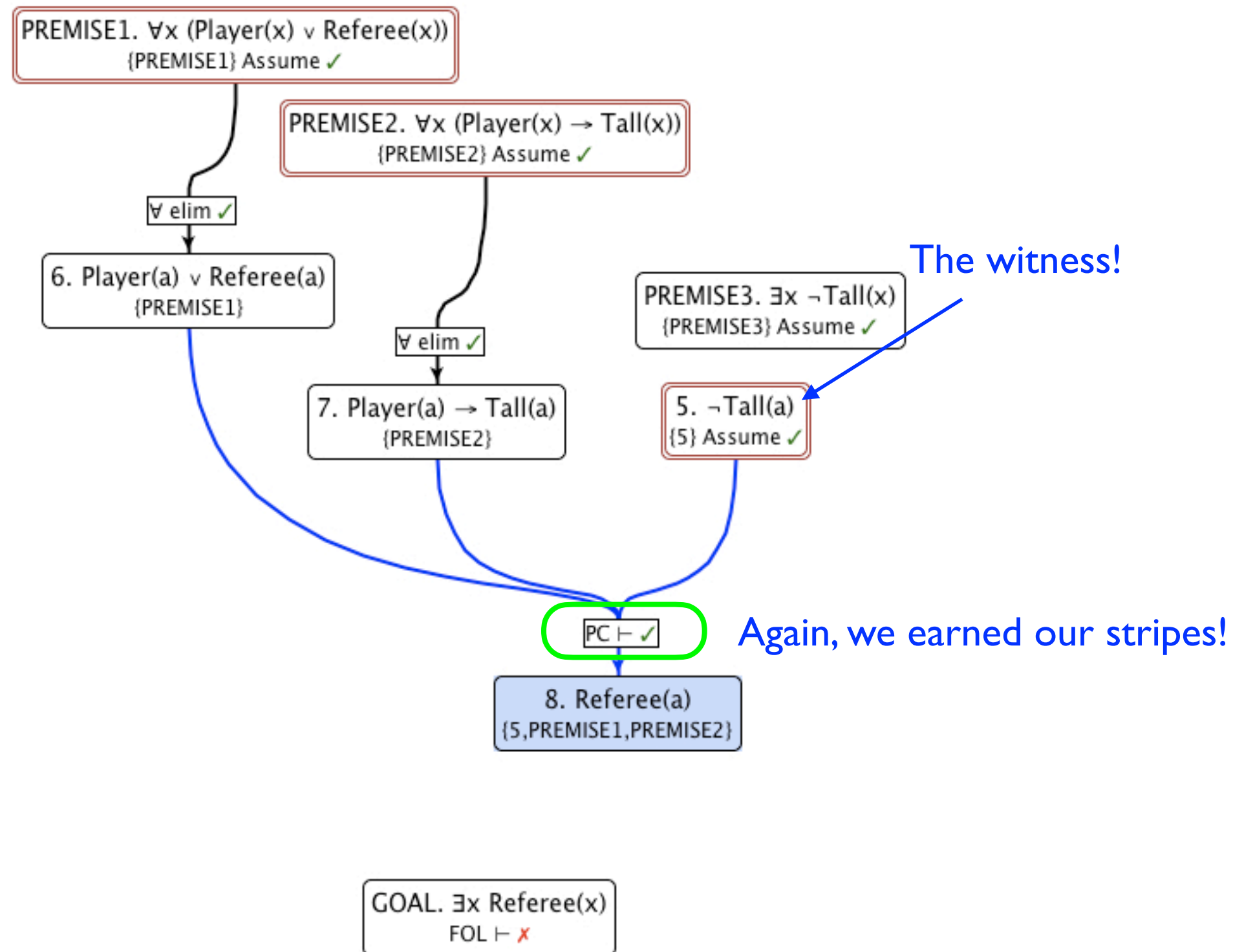
Step 4



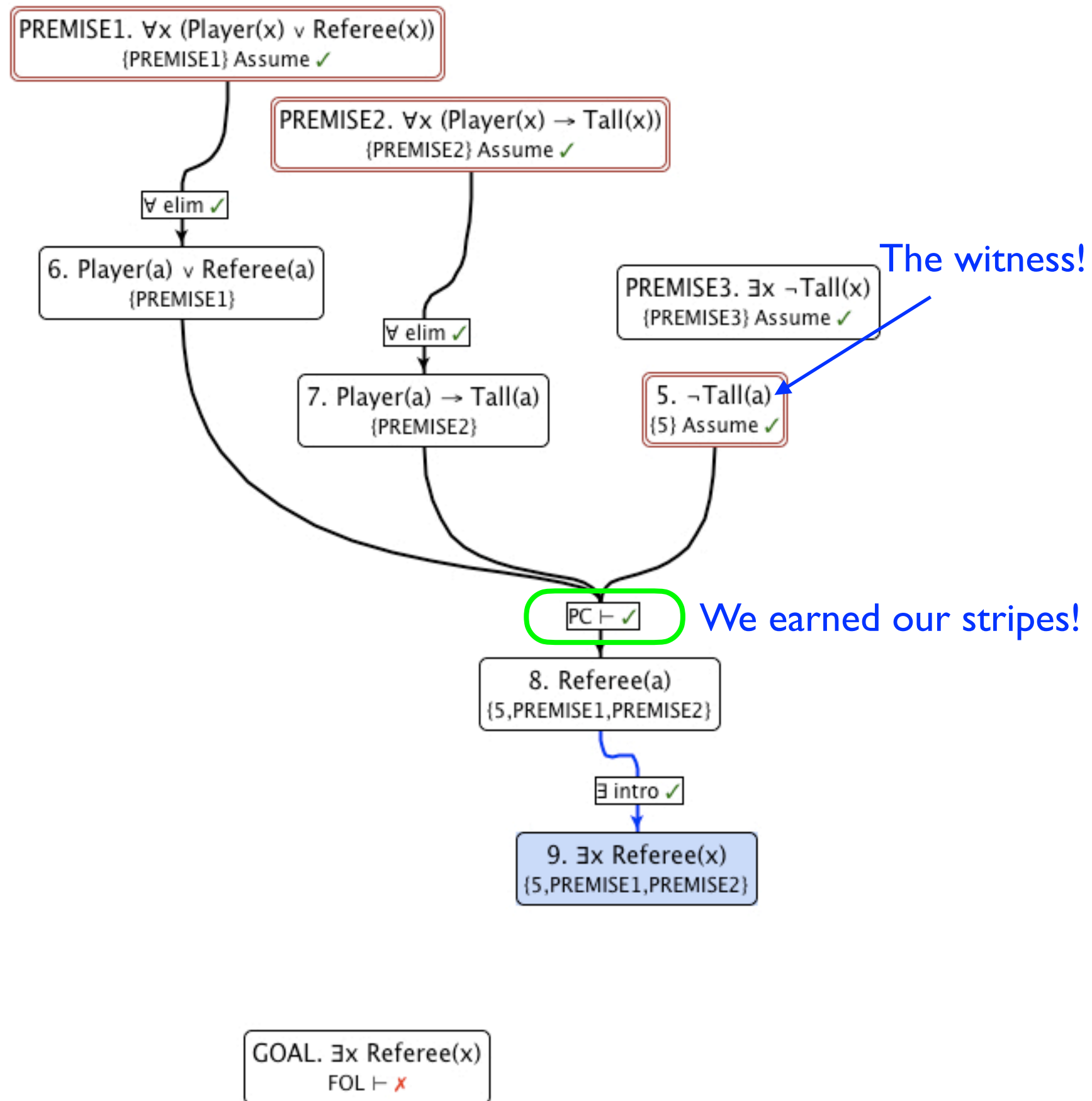
Step 4



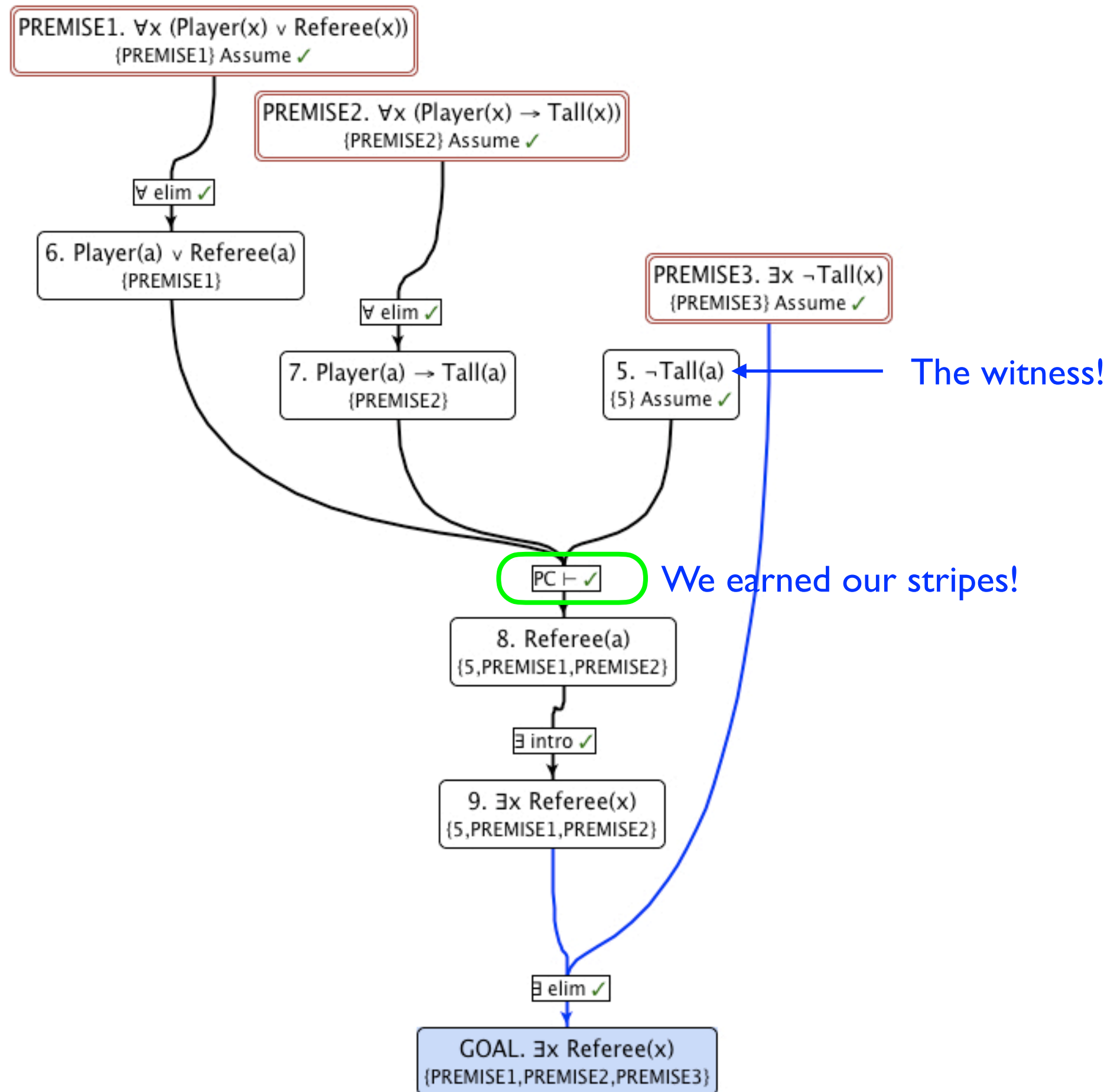
Step 4



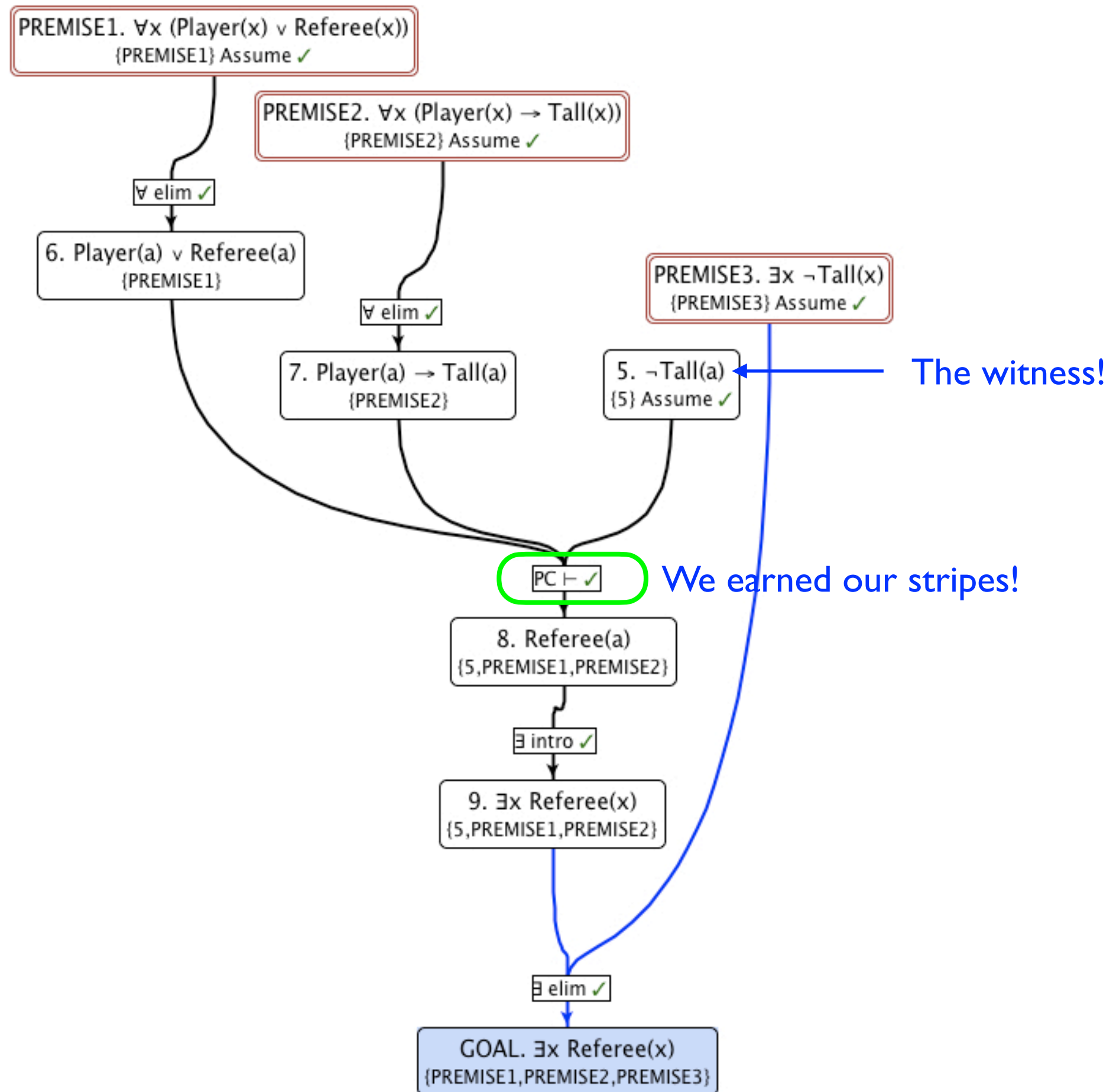
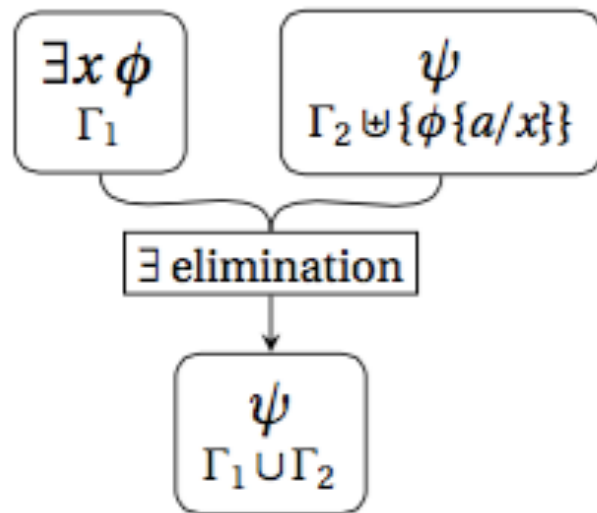
Step 5



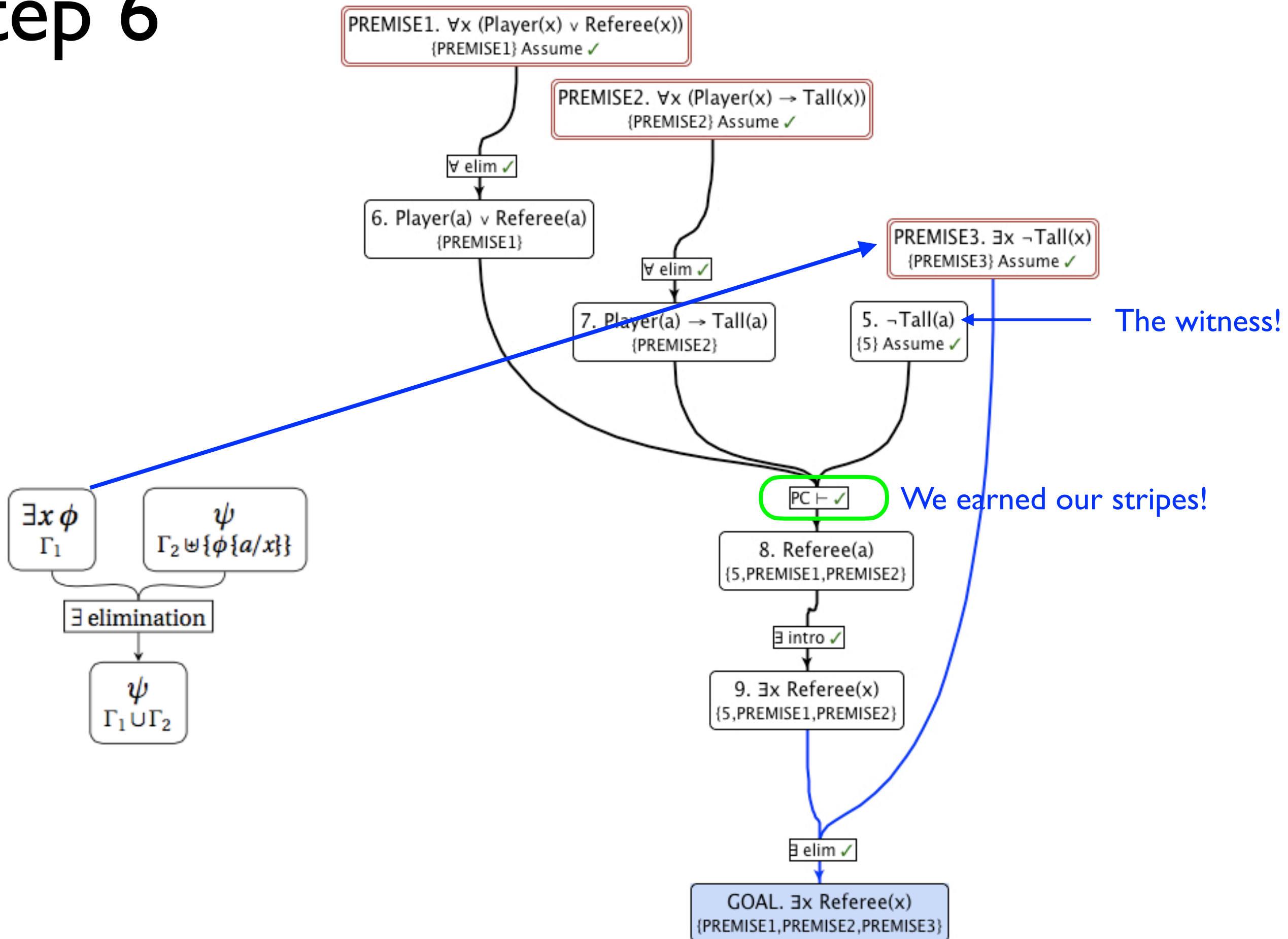
Step 6



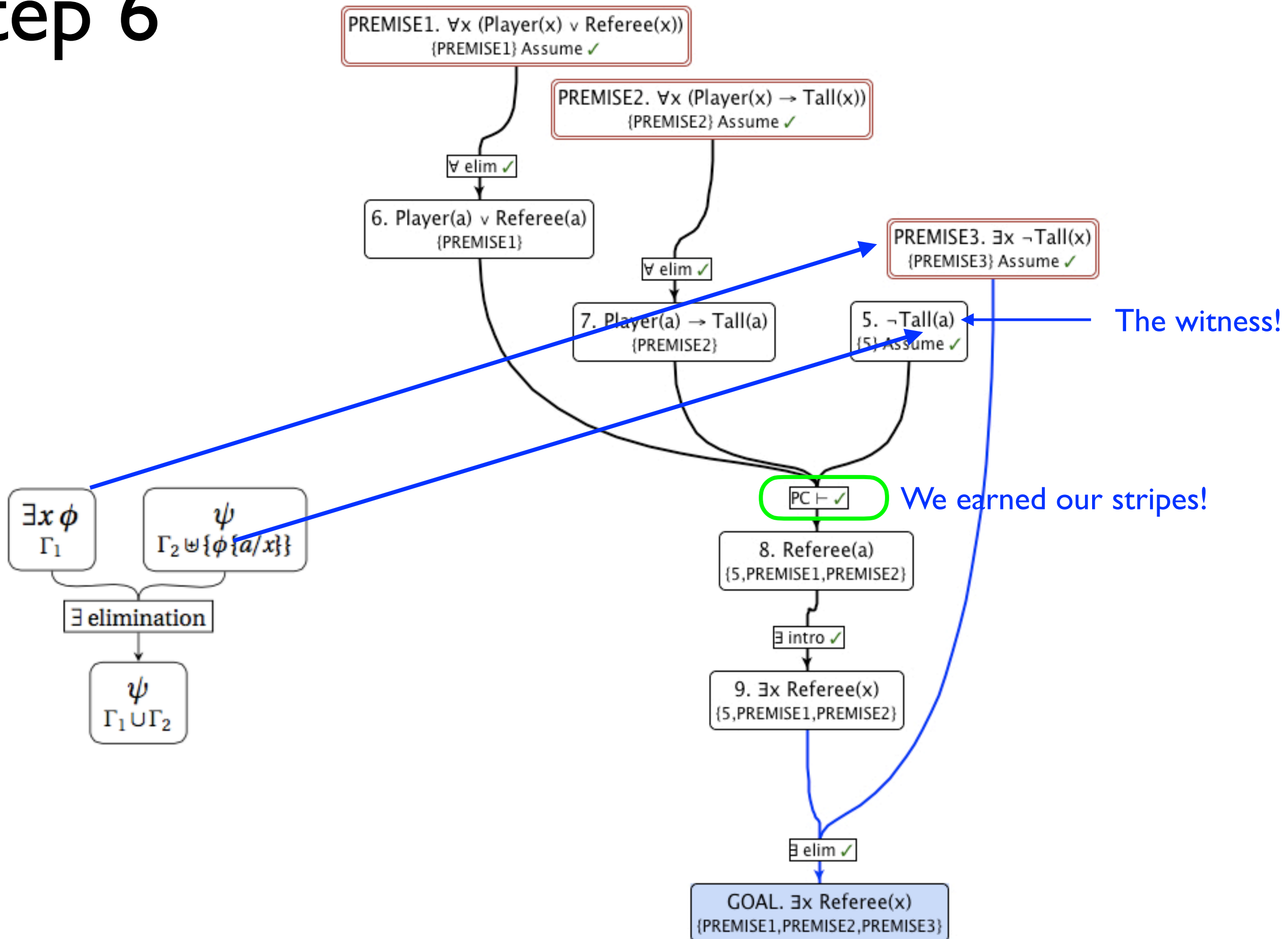
Step 6



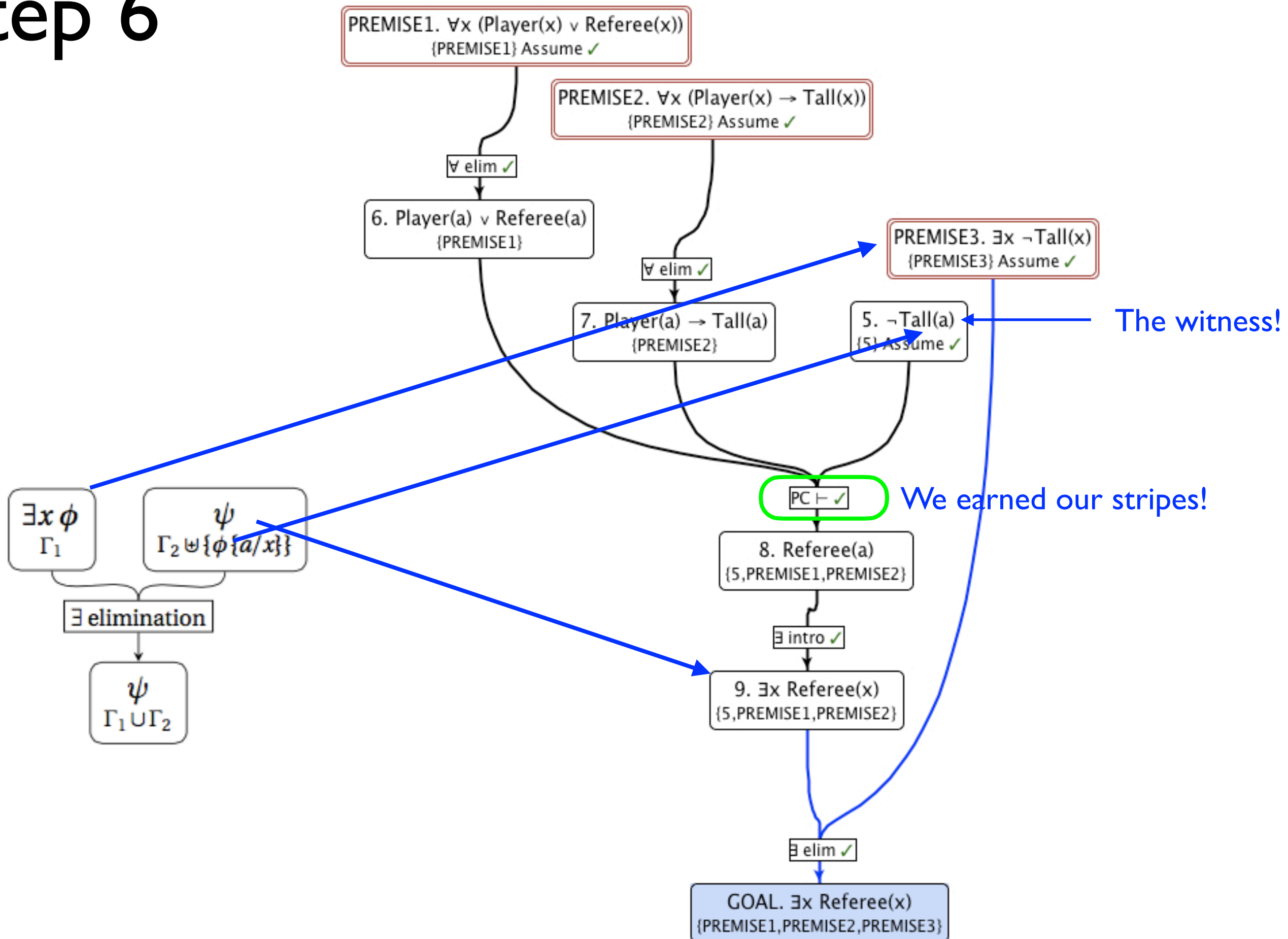
Step 6



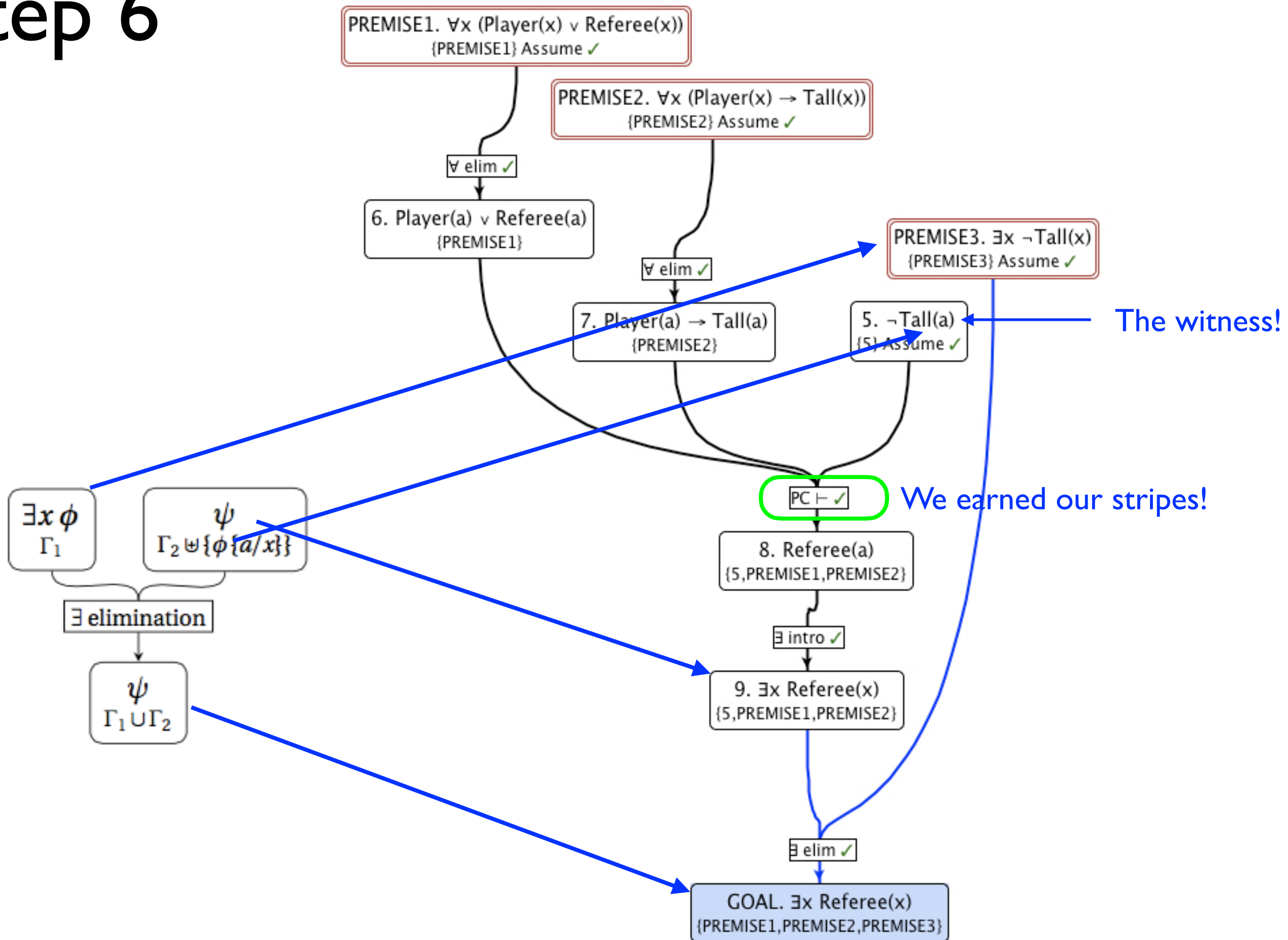
Step 6



Step 6



Step 6



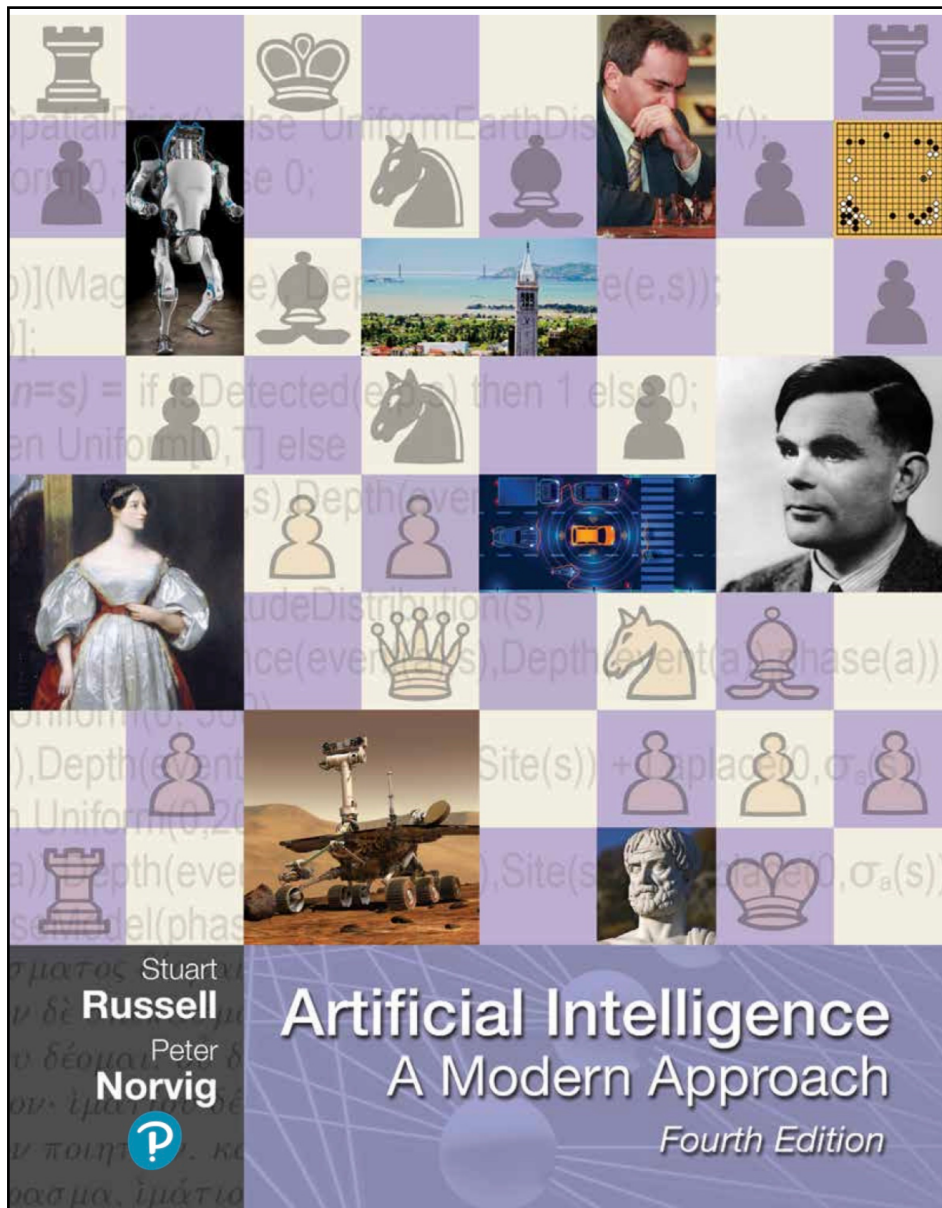
$$\{\forall x(\text{Scared}(x) \leftrightarrow \text{Small}(x)), \exists x \neg \text{Scared}(x)\} \vdash \exists x \neg \text{Small}(x)$$

$$\{\exists x, y \text{Contiguous}(x, y), \forall x, y(\text{Contiguous}(x, y) \rightarrow \neg \text{SameCountry}(x, y))\} \vdash \exists x, y \neg \text{SameCountry}(x, y)$$

Colonel West is a Criminal

<https://rpi.logicamodernapproach.com/hyperslate/public/Selmer.Bringsjord@gmail.com/WestIsACriminal>

Colonel West is a Criminal



<https://rpi.logicamodernapproach.com/hyperslate/public/Selmer.Bringsjord@gmail.com/WestIsACriminal>

Colonel West is a Criminal

9

INFERENCE IN FIRST-ORDER LOGIC

In which we define effective procedures for answering questions posed in first-order logic.

Chapter 7 defined the notion of **inference** and showed how sound and complete inference can be achieved for propositional logic. In this chapter, we extend those results to obtain algorithms that can answer any answerable question stated in first-order logic. This is significant, because more or less anything can be stated in first-order logic if you work hard enough at it.

Section 9.1 introduces inference rules for quantifiers and shows how to reduce first-order inference to propositional inference, albeit at great expense. Section 9.2 describes the idea of **unification**, showing how it can be used to construct inference rules that work directly with first-order sentences. We then discuss three major families of first-order inference algorithms: **forward chaining** and its applications to **deductive databases** and **production systems** are covered in Section 9.3; **backward chaining** and **logic programming** systems are developed in Section 9.4; and resolution-based **theorem-proving** systems are described in Section 9.5. In general, one tries to use the most efficient method that can accommodate the facts and axioms that need to be expressed. Reasoning with fully general first-order sentences using resolution is usually less efficient than reasoning with definite clauses using forward or backward chaining.

9.1 PROPOSITIONAL VS. FIRST-ORDER INFERENCE

This section and the next introduce the ideas underlying modern logical inference systems. We begin with some simple inference rules that can be applied to sentences with quantifiers to obtain sentences without quantifiers. These rules lead naturally to the idea that *first-order* inference can be done by converting the knowledge base to *propositional* logic and using *propositional* inference, which we already know how to do. The next section points out an obvious shortcut, leading to inference methods that manipulate first-order sentences directly.

Colonel West is a Criminal

9

INFERENCE IN FIRST-ORDER LOGIC

In which we define effective procedures for answering questions posed in first-order logic.

Chapter 7 defined the notion of **inference** and showed how sound and complete inference can be achieved for propositional logic. In this chapter, we extend those results to obtain algorithms that can answer any answerable question stated in first-order logic. This is significant, because more or less anything can be stated in first-order logic if you work hard enough at it.

Section 9.1 introduces inference rules for quantifiers and shows how to reduce first-order inference to propositional inference, albeit at great expense. Section 9.2 describes the idea of **unification**, showing how it can be used to construct inference rules that work directly with first-order sentences. We then discuss three major families of first-order inference algorithms: **forward chaining** and its applications to **deductive databases** and **production systems** are covered in Section 9.3; **backward chaining** and **logic programming** systems are developed in Section 9.4; and resolution-based **theorem-proving** systems are described in Section 9.5. In general, one tries to use the most efficient method that can accommodate the facts and axioms that need to be expressed. Reasoning with fully general first-order sentences using resolution is usually less efficient than reasoning with definite clauses using forward or backward chaining.

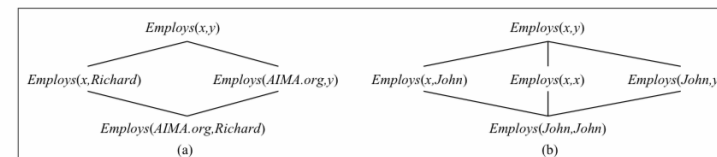
9.1 PROPOSITIONAL VS. FIRST-ORDER INFERENCE

This section and the next introduce the ideas underlying modern logical inference systems. We begin with some simple inference rules that can be applied to sentences with quantifiers to obtain sentences without quantifiers. These rules lead naturally to the idea that *first-order inference* can be done by converting the knowledge base to *propositional* logic and using *propositional* inference, which we already know how to do. The next section points out an obvious shortcut, leading to inference methods that manipulate first-order sentences directly.

272

280

Chapter 9. Inference in First-Order Logic



9.3 FORWARD CHAINING

A forward-chaining algorithm for propositional definite clauses was given in Section 7.5. The idea is simple: start with the atomic sentences in the knowledge base and apply Modus Ponens in the forward direction, adding new atomic sentences, until no further inferences can be made. Here, we explain how the algorithm is applied to first-order definite clauses and how it can be implemented efficiently. Definite clauses such as $\text{Situation} \Rightarrow \text{Response}$ are especially useful for systems that make inferences in response to newly arrived information. Many systems can be defined this way, and reasoning with forward chaining can be much more efficient than resolution theorem proving. Therefore it is often worthwhile to try to build a knowledge base using only definite clauses so that the cost of resolution can be avoided.

First-order definite clauses

First-order definite clauses closely resemble propositional definite clauses (page 217): they are disjunctions of literals of which *exactly one is positive*. A definite clause either is atomic or is an implication whose antecedent is a conjunction of positive literals and whose consequent is a single positive literal. The following are first-order definite clauses:

$\text{King}(x) \wedge \text{Greedy}(x) \Rightarrow \text{Evil}(x).$
 $\text{King}(\text{John}).$
 $\text{Greedy}(y).$

Unlike propositional literals, first-order literals can include variables, in which case those variables are assumed to be universally quantified. (Typically, we omit universal quantifiers when writing definite clauses.) Definite clauses are a suitable normal form for use with Generalized Modus Ponens.

Not every knowledge base can be converted into a set of definite clauses, because of the single-positive-literal restriction, but many can. Consider the following problem:

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

especially useful for systems that make inferences in response to newly arrived information. Many systems can be defined this way, and reasoning with forward chaining can be much more efficient than resolution theorem proving. Therefore it is often worthwhile to try to build a knowledge base using only definite clauses so that the cost of resolution can be avoided.

First-order definite clauses

First-order definite clauses closely resemble propositional definite clauses (page 217): they are disjunctions of literals of which *exactly one is positive*. A definite clause either is atomic or is an implication whose antecedent is a conjunction of positive literals and whose consequent is a single positive literal. The following are first-order definite clauses:

$$\begin{aligned} &King(x) \wedge Greedy(x) \Rightarrow Evil(x) . \\ &King(John) . \\ &Greedy(y) . \end{aligned}$$

Unlike propositional literals, first-order literals can include variables, in which case those variables are assumed to be universally quantified. (Typically, we omit universal quantifiers when writing definite clauses.) Definite clauses are a suitable normal form for use with Generalized Modus Ponens.

Not every knowledge base can be converted into a set of definite clauses, because of the single-positive-literal restriction, but many can. Consider the following problem:

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

especially useful for systems that make inferences in response to newly arrived information. Many systems can be defined this way, and reasoning with forward chaining can be much more efficient than resolution theorem proving. Therefore it is often worthwhile to try to build a knowledge base using only definite clauses so that the cost of resolution can be avoided.

First-order definite clauses

First-order definite clauses closely resemble propositional definite clauses (page 217): they are disjunctions of literals of which *exactly one* is positive, or is an implication whose antecedent is a conjunction of literals and whose consequent is a single positive literal. The following are

$$\begin{aligned} & King(x) \wedge Greedy(x) \Rightarrow Evil(x) . \\ & King(John) . \\ & Greedy(y) . \end{aligned}$$

Unlike propositional literals, first-order literals have variables. (The variables are assumed to be universally quantified when writing definite clauses.) Definite clauses are combined using Generalized Modus Ponens.

Not every knowledge base can be converted to a single-positive-literal restriction, but many can.

The law says that it is a crime for an American country Nono, an enemy of America, has some missiles, and it is sold to it by Colonel West, who is American.

We will prove that West is a criminal. First, we will represent these facts as first-order definite clauses. The next section shows how the forward-chaining algorithm solves the problem.

“... it is a crime for an American to sell weapons to hostile nations”:

$$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x) . \quad (9.3)$$

“Nono ... has some missiles.” The sentence $\exists x. Owns(Nono, x) \wedge Missile(x)$ is transformed into two definite clauses by Existential Elimination, introducing a new constant M_1 :

$$Owns(Nono, M_1) \quad (9.4)$$

$$Missile(M_1) \quad (9.5)$$

“All of its missiles were sold to it by Colonel West”:

$$Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono) . \quad (9.6)$$

We will also need to know that missiles are weapons:

$$Missile(x) \Rightarrow Weapon(x) \quad (9.7)$$

and we must know that an enemy of America counts as “hostile”:

$$Enemy(x, America) \Rightarrow Hostile(x) . \quad (9.8)$$

“West, who is American ...”:

$$American(West) . \quad (9.9)$$

“The country Nono, an enemy of America ...”:

$$Enemy(Nono, America) . \quad (9.10)$$

DATALOG

This knowledge base contains no function symbols and is therefore an instance of the class of **Datalog** knowledge bases—that is, sets of first-order definite clauses with no function symbols. We will see that the absence of function symbols makes inference much easier.

A simple forward-chaining algorithm

The first forward chaining algorithm we will consider is a very simple one, as shown in Figure 9.3. Starting from the known facts, it triggers all the rules whose premises are satisfied, adding their conclusions to the known facts. The process repeats until the query is answered (assuming that just one answer is required) or no new facts are added. Notice that a fact is not “new” if it is just a **renaming** of a known fact. One sentence is a renaming of another if they are identical except for the names of the variables. For example, $Likes(x, IceCream)$ and $Likes(y, IceCream)$ are renamings of each other because they differ only in the choice of x or y ; their meanings are identical: everyone likes ice cream.

RENAMING

We will use our crime problem to illustrate how FOL-FC-ASK works. The implication sentences are (9.3), (9.6), (9.7), and (9.8). Two iterations are required:

- On the first iteration, rule (9.3) has unsatisfied premises. Rule (9.6) is satisfied with $\{x/M_1\}$, and $Sells(West, M_1, Nono)$ is added. Rule (9.7) is satisfied with $\{x/M_1\}$, and $Weapon(M_1)$ is added. Rule (9.8) is satisfied with $\{x/Nono\}$, and $Hostile(Nono)$ is added.

We will prove that West is a criminal. First, we will represent these facts as first-order definite clauses. The next section shows how the forward-chaining algorithm solves the problem.

“... it is a crime for an American to sell weapons to hostile nations”:

$$\text{American}(x) \wedge \text{Weapon}(y) \wedge \text{Sells}(x, y, z) \wedge \text{Hostile}(z) \Rightarrow \text{Criminal}(x) . \quad (9.3)$$

“Nono ... has some missiles.” The sentence $\exists x \text{ Owns}(\text{Nono}, x) \wedge \text{Missile}(x)$ is transformed into two definite clauses by Existential Elimination, introducing a new constant M_1 :

$$\text{Owns}(\text{Nono}, M_1) \quad (9.4)$$

$$\text{Missile}(M_1) \quad (9.5)$$

“All of its missiles were sold to it by Colonel West”:

$$\text{Missile}(x) \wedge \text{Owns}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono}) . \quad (9.6)$$

We will also need to know that missiles are weapons:

$$\text{Missile}(x) \Rightarrow \text{Weapon}(x) \quad (9.7)$$

and we must know that an enemy of America counts as “hostile”:

$$\text{Enemy}(x, \text{America}) \Rightarrow \text{Hostile}(x) . \quad (9.8)$$

“West, who is American ...”:

$$\text{American}(\text{West}) . \quad (9.9)$$

“The country Nono, an enemy of America ...”:

$$\text{Enemy}(\text{Nono}, \text{America}) . \quad (9.10)$$

This knowledge base contains no function symbols and is therefore an instance of the class of **Datalog** knowledge bases—that is, sets of first-order definite clauses with no function symbols. We will see that the absence of function symbols makes inference much easier.

A simple forward-chaining algorithm

The first forward chaining algorithm we will consider is a very simple one, as shown in Figure 9.3. Starting from the known facts, it triggers all the rules whose premises are satisfied, adding their conclusions to the known facts. The process repeats until the query is answered (assuming that just one answer is required) or no new facts are added. Notice that a fact is not “new” if it is just a **renaming** of a known fact. One sentence is a renaming of another if

Colonel West is a Criminal

<https://rpi.logicamodernapproach.com/hyperslate/public/Selmer.Bringsjord@gmail.com/WestIsACriminal>

*Hvis du forstår det, kan
du bevise det.*