Natural Deduction Proof Strategies & Zeroth Order Logic in Hyperslate

James T. Oswald

This Class

- Iff Intro and Elim Rules
- Natural Deduction Proof Finding Strategies and Heuristics
- Zoombinis has been canceled, In class problems instead!
- Zeroth Order Logic
  - Equality Intro and Elim Rules

You can access this slide deck at: https://bit.ly/4etaD83

You can access lazyslate at: https://github.com/RAIRLab/lazyslate (please star thx)

Iff (if and only if / Biconditional) Introduction

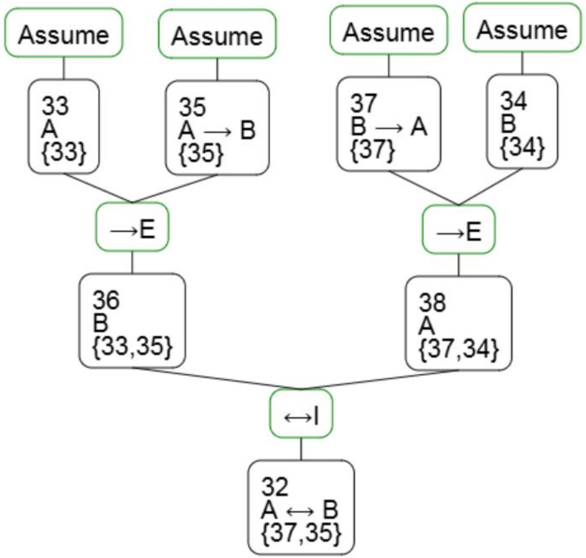
“φ iff ψ” can also be interpreted “φ has the same truth value as ψ” or “φ is logically equal to ψ”

Premicies: “φ is provable from Γ and ψ”
“ψ is provable from Σ and φ”

Conclusions: “φ iff ψ is provable from Γ and Σ”

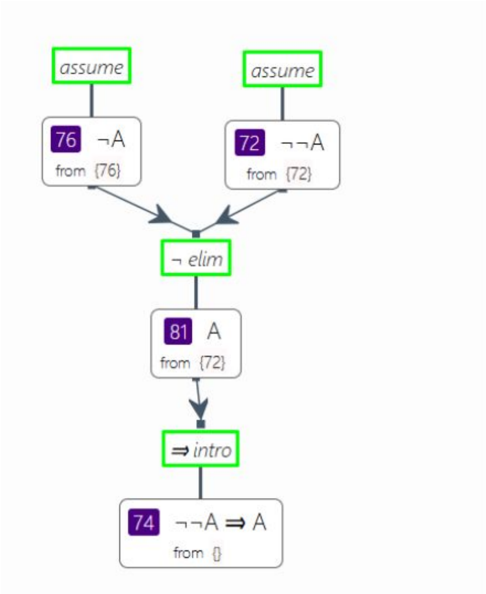
It’s just implication introduction but you have prove both directions: φ -> ψ and ψ -> φ

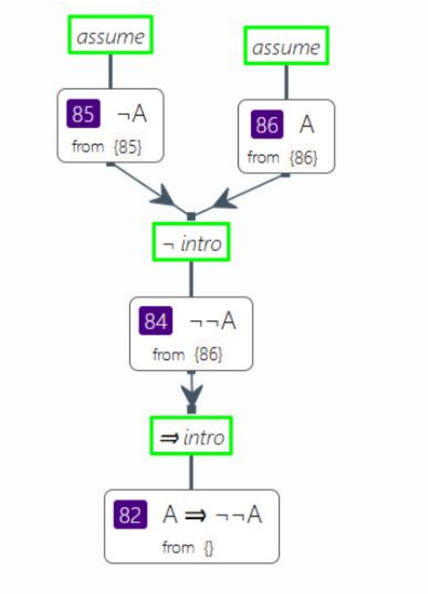
$$\frac{\Gamma \cup \{\phi\} \vdash \psi \quad \Sigma \cup \{\psi\} \vdash \phi}{\Gamma \cup \Sigma \vdash \phi \leftrightarrow \psi} \leftrightarrow I$$



Iff Intro Example 1

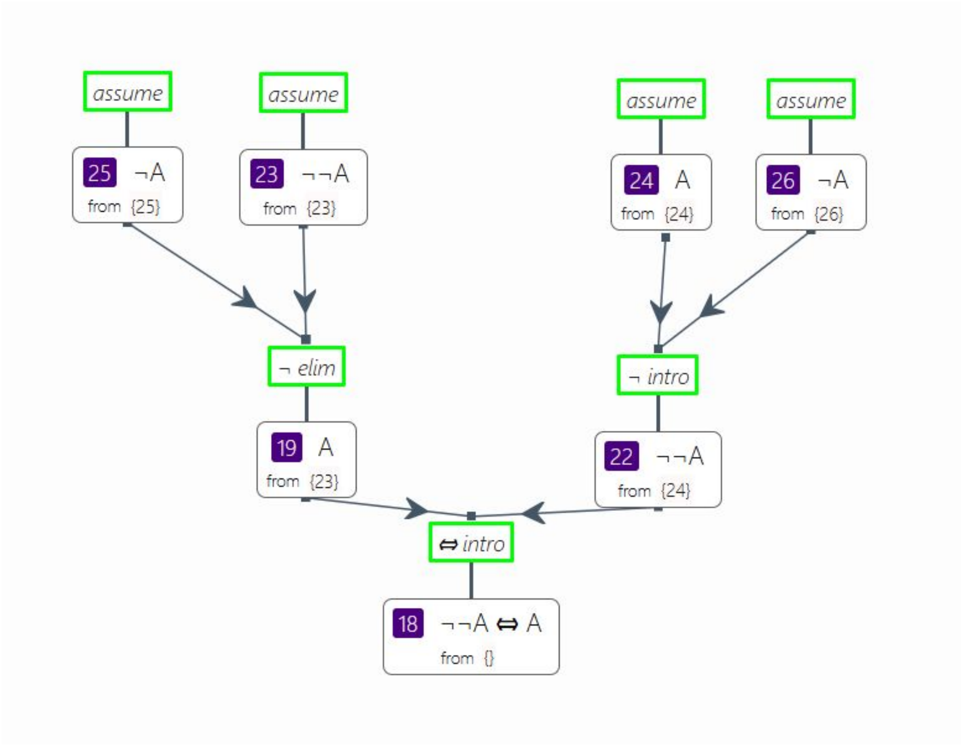
Recall from last week we proved:





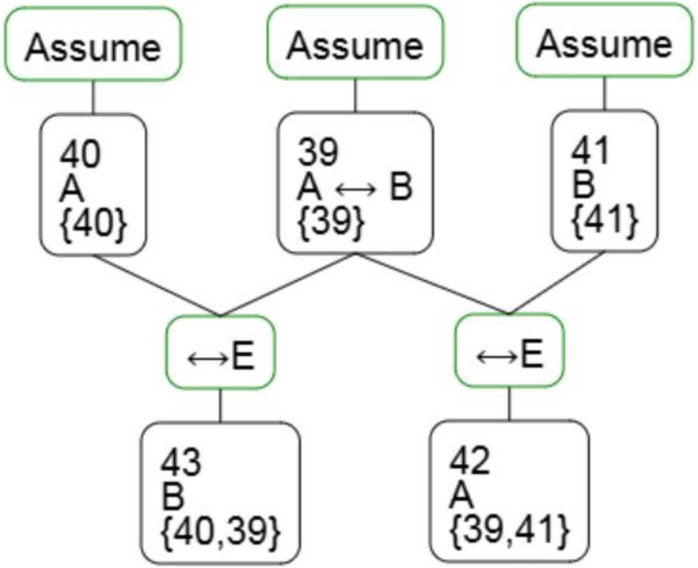
Iff Intro Example 1

If you can prove a conditional (A => B) and its converse (B => A) , you can prove the biconditional (A ⇔ B) using the same two proofs you would use for (A => B) and (B => A).



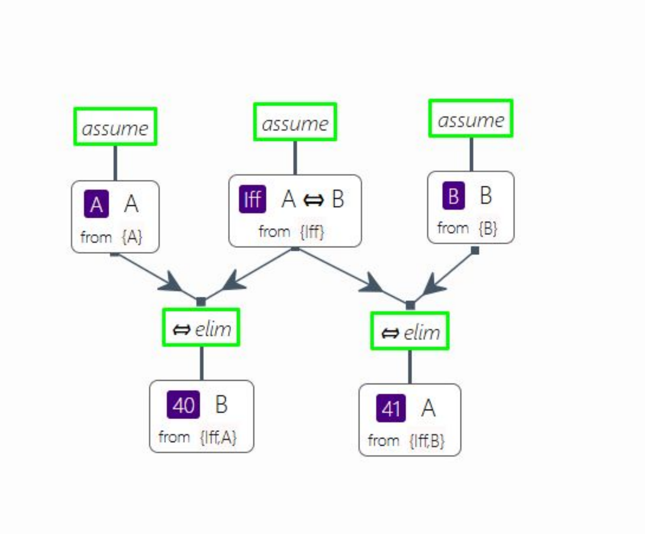
Iff Elim

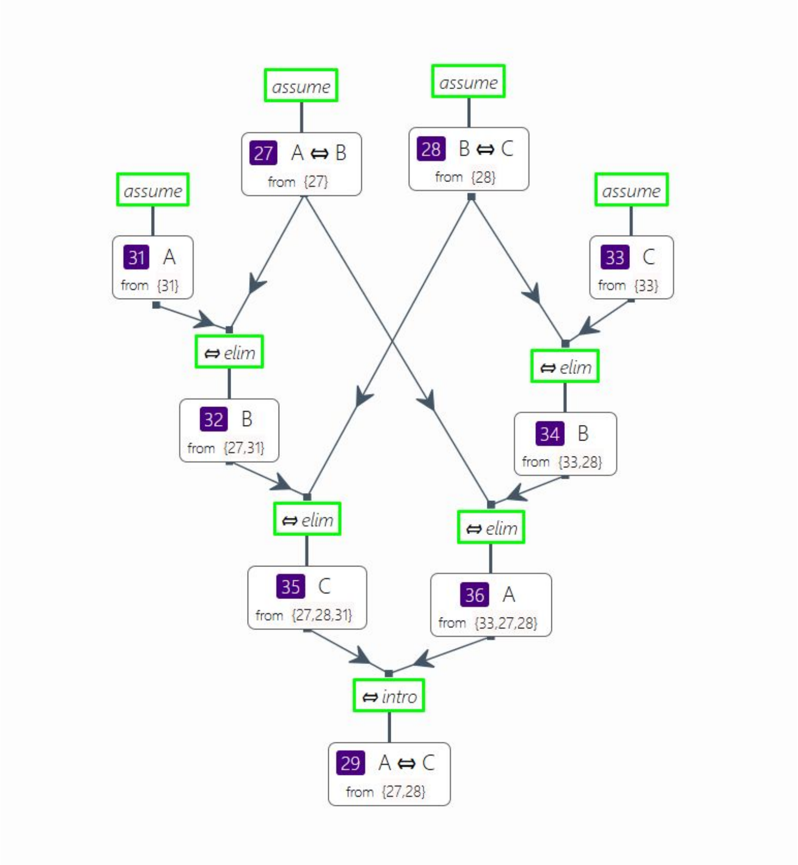
Modus Ponens (If Elim) but you can use either side as the antecedent and the other will automatically become the consequent.



$$\frac{\Gamma \vdash \phi \quad \Sigma \vdash \phi \leftrightarrow \psi}{\Gamma \cup \Sigma \vdash \psi} \leftrightarrow E_l \qquad \frac{\Gamma \vdash \psi \quad \Sigma \vdash \phi \leftrightarrow \psi}{\Gamma \cup \Sigma \vdash \phi} \leftrightarrow E_r$$

Iff Elim Example: Transitivity of Iff





Hyperslate Terms

Definition (Given): A *given* is an assumption you are allowed to use to prove a goal, it is allowed to appear on the “from:” section of your goal.

Definition (Derived): Anything formula you build from your assumptions and givens from the top down.

Definition (Theorem): A theorem is something you can prove with no givens. Its “from:” section will be empty e.g. “from:{}”

Definition (Goal): a *goal* is something we want to prove, it may be a theorem (that depends on no assumptions) or it may depend on *given assumptions.*
It will always be the bottom node in a proof.

Definition (Subgoal)*:* A *subgoal* is a smaller goal produced by breaking the goal down using introduction rules.

Common Proof Finding Heuristics and Strategies

1. Work top down and bottom up, meet in the middle
2. Prove conjunction goals by splitting them into two subgoals
3. Prove disjunction goals by selecting one disjunct as a subgoal
4. Prove implications by assuming and the antecedent and proving the consequent
5. Introduce any assumption on any node
6. Prove negations by assuming the positive and deriving a contradiction
7. Prove biconditionals by assuming each side and proving the other side.
8. Try proof by contradiction (On Anything)
9. Try proof by cases when given a disjunction

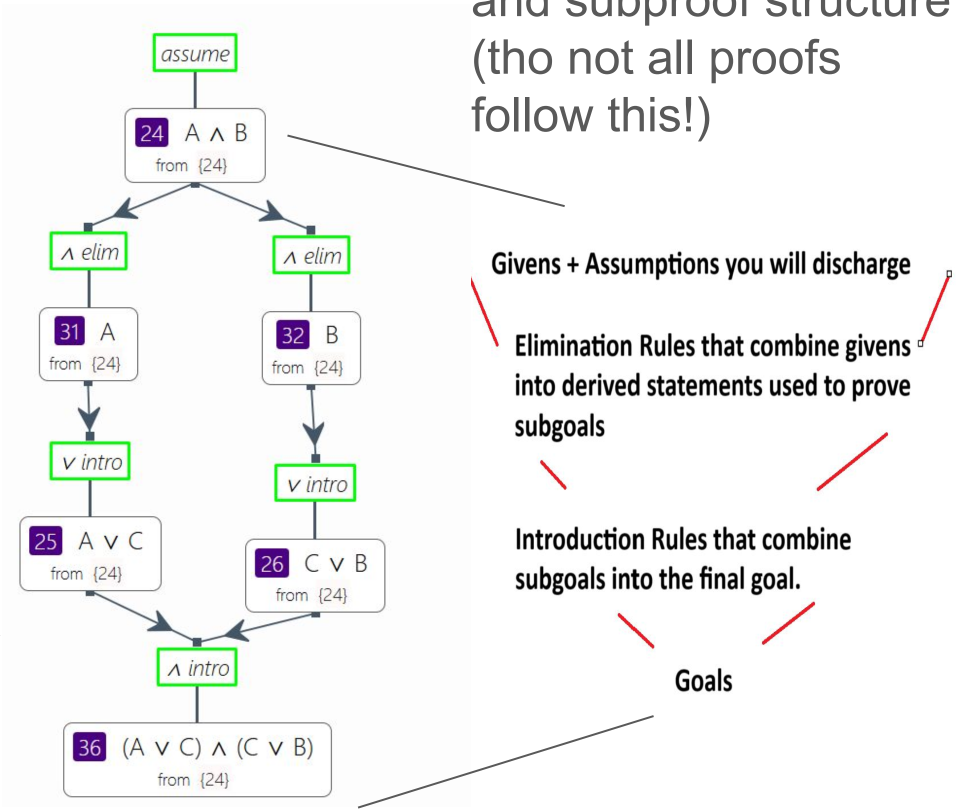
(Examples on the following slides)

General Tips

- Work both top down AND bottom up, meet in the middle!!!!
  - Use elimination rules from the top to destruct assumptions and givens.
  - Use Introduction rules from the bottom to destruct goals into multiple simpler goals.

- Goal is a theorem that can have no assumptions / You have no Givens?
  - You have to work from the bottom up, you can work top down after making your own assumptions that you will later discard.

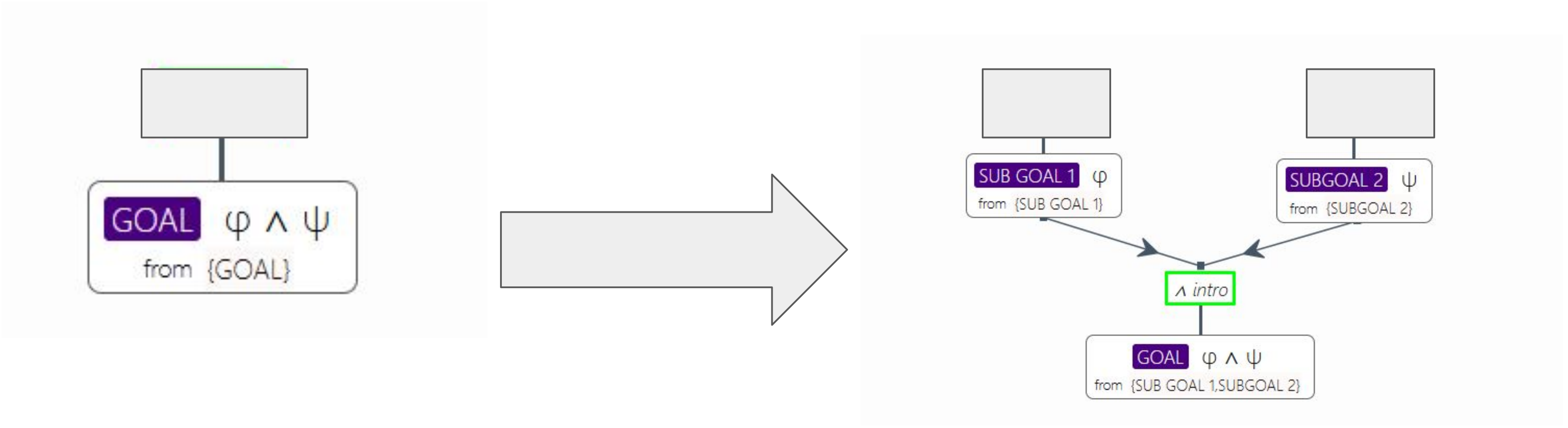
A prototypical proof and subproof structure (tho not all proofs follow this!)



Proving Conjunction Goals (Bottom Up)

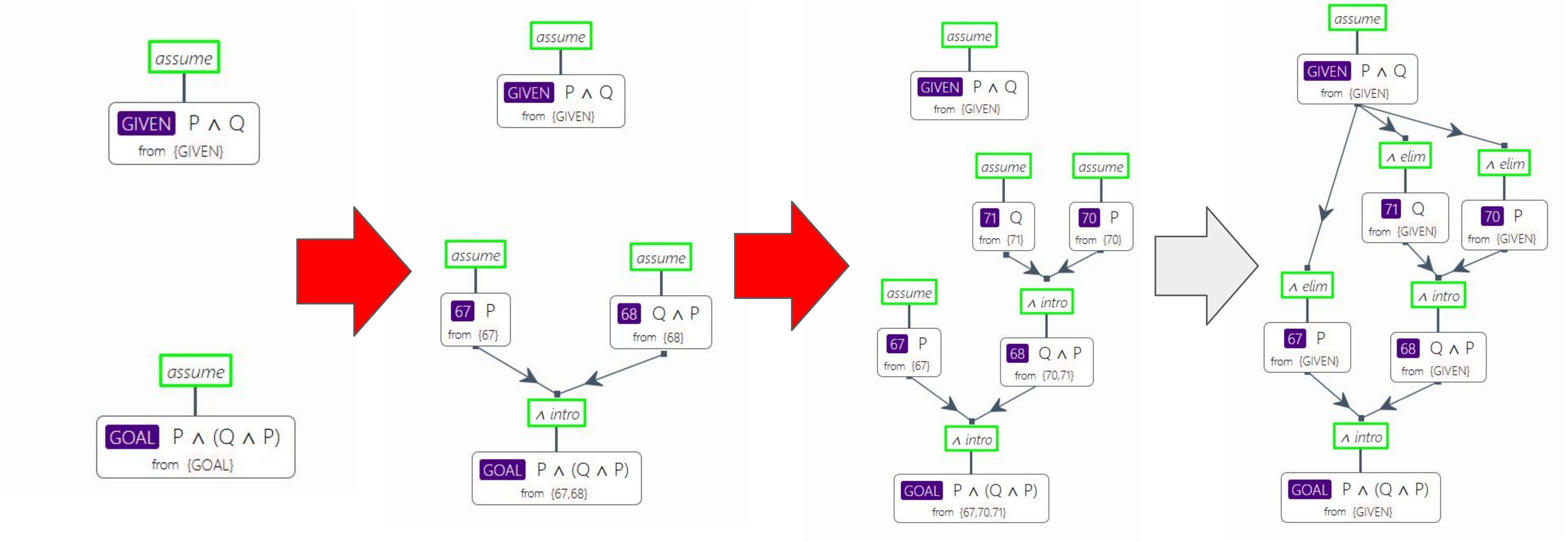
If a goal is a conjunction, use conjunction introduction to split it into two subgoals for you to prove!

You can now use the subgoals as your new goals.



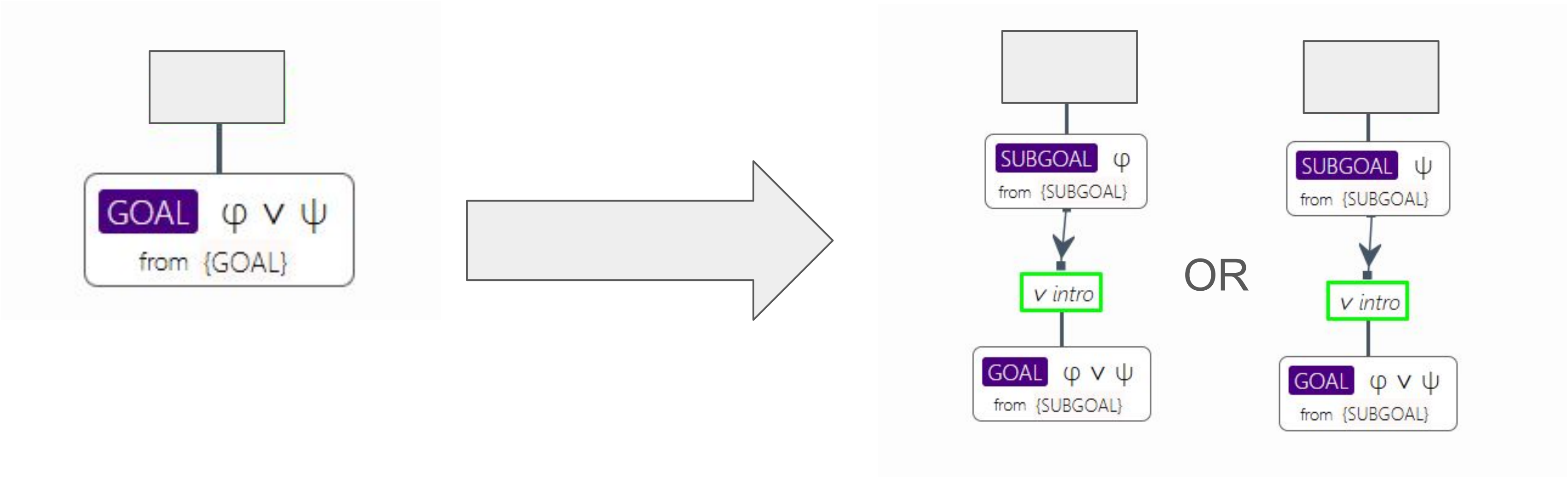
Proving Conjunction Goals Example

From Last Class: Given P and Q prove P and (Q and P)



Proving Disjunction Goals (Bottom Up)

If a goal is a disjunction, use disjunction introduction to pick one of the sides as a subgoal! You should pick the side intelligently! pick the side that looks easier to prove as your new subgoal, if you pick wrong it may be impossible to prove!



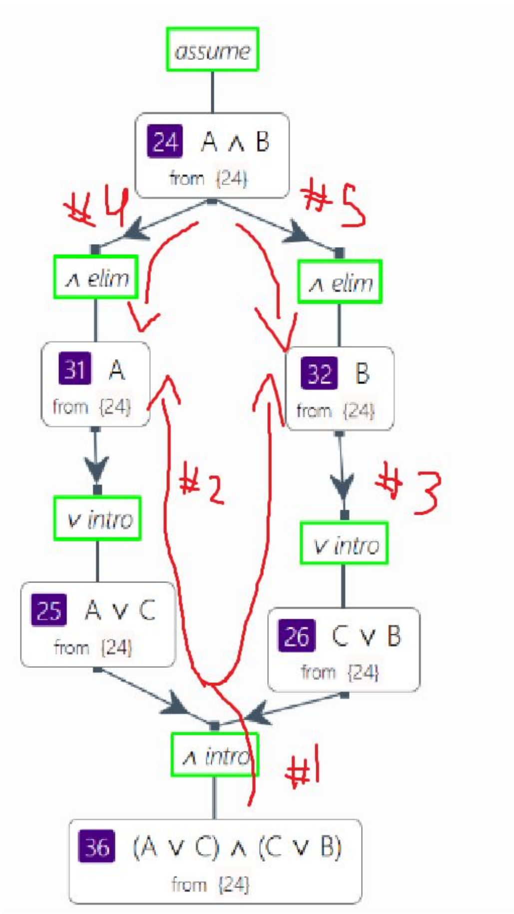
Proving Disjunction Goals Example

From Last Class: A and B proves (A or C) and (C or B)

Start by splitting our goal into subgoals with and intro.

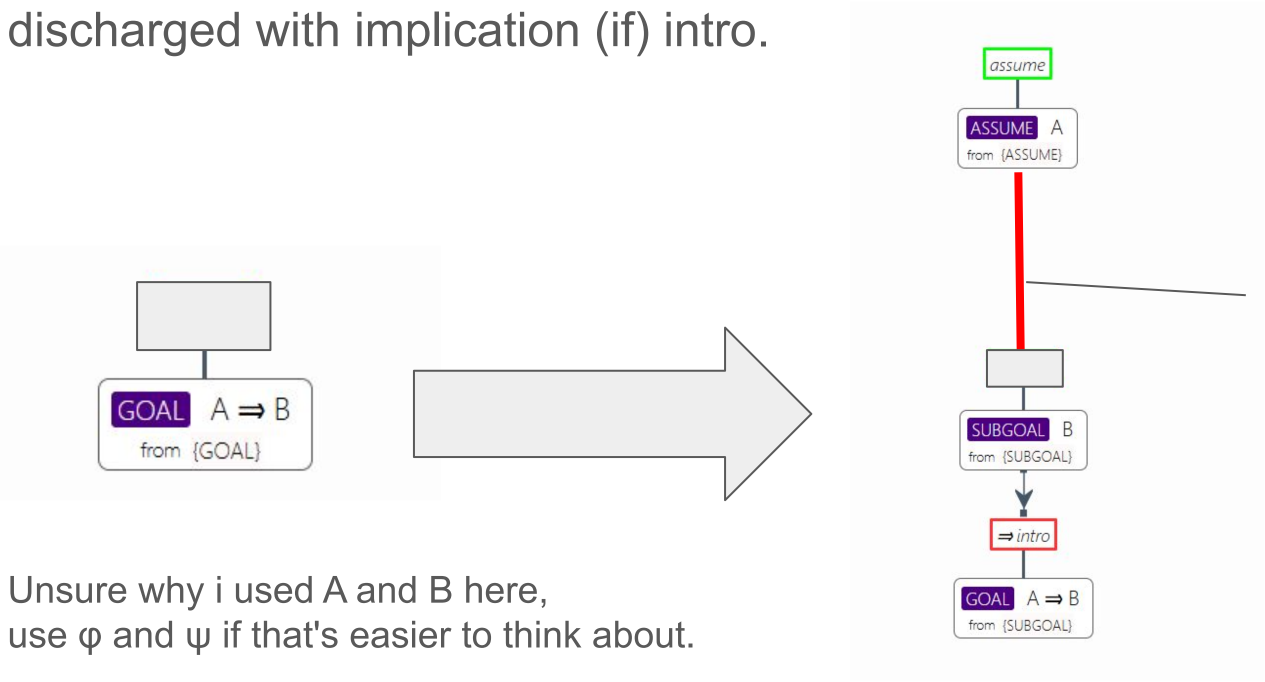
Now split each subgoal into something we can prove from A and B. Choose A, B respectively (can't prove C from A and B, so don't chose it).

Now get A, B from A and B with conjunction elim.



Proving Implication Goals (Bottom Up + Top Down)

If a goal is an implication (A => B), assume the antecedent (A) and prove the consequent (B). The assumption will be passed down your proof chain and discharged with implication (if) intro.

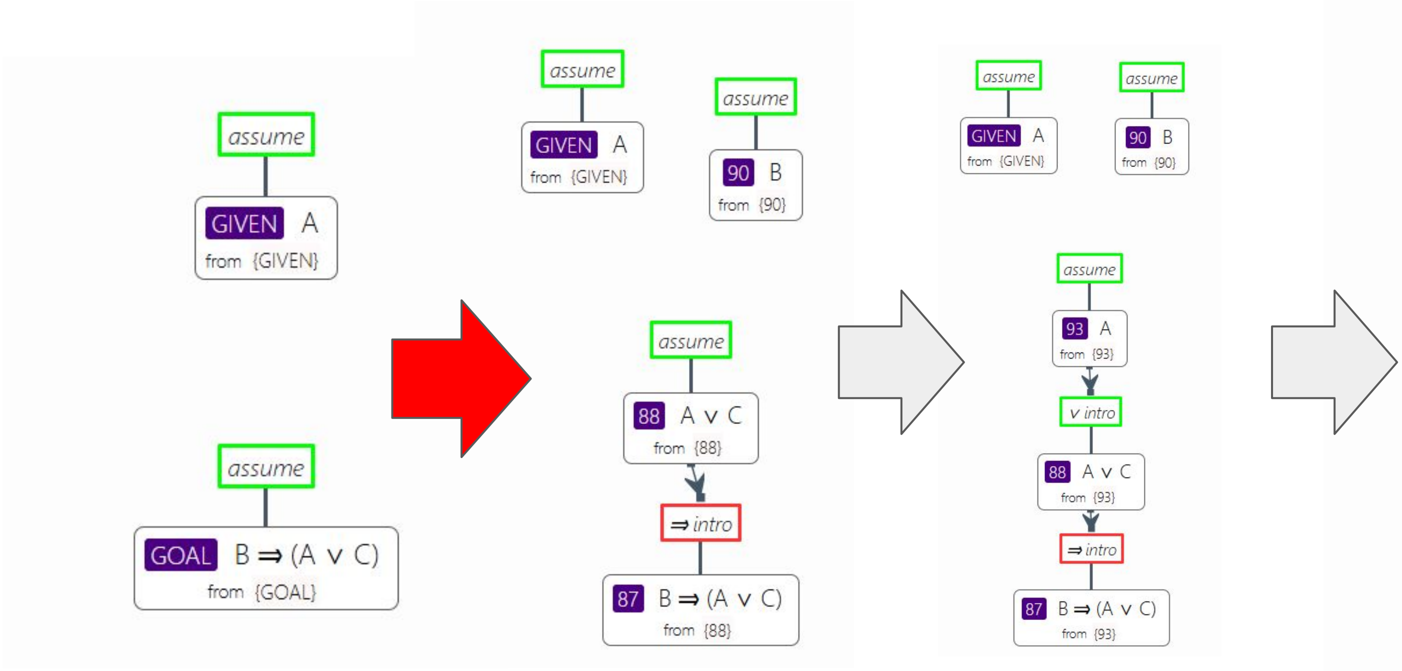


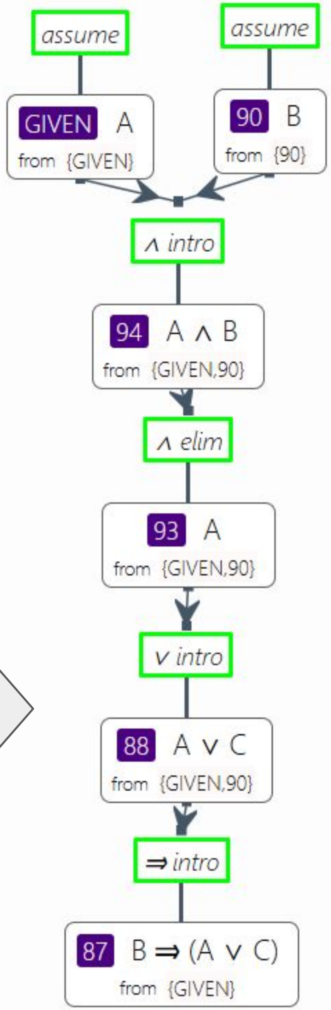
It's your job now to use A to prove B. Doing so will make the "=> intro" green.

Unsure why i used A and B here,
use φ and ψ if that's easier to think about.

Proving Implication Goals Example

You will *frequently* need this technique where you add an arbitrary assumption to a node.



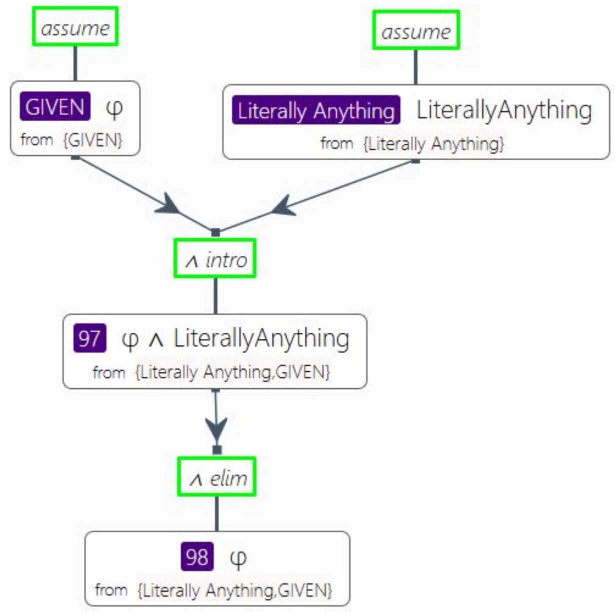


Introduce Any Assumption On Any Node

Many rules (=> intro, ¬ intro, ¬elim, <=> intro) require a specific premise to be in the set of assumptions.

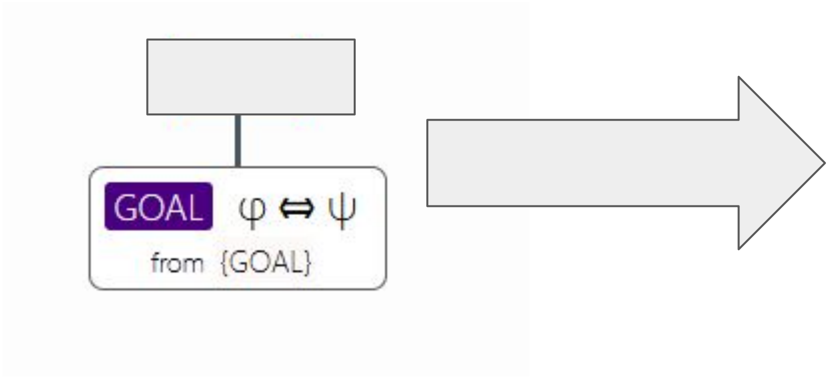
You can add ANY premice into a set of assumptions of any node using this technique.

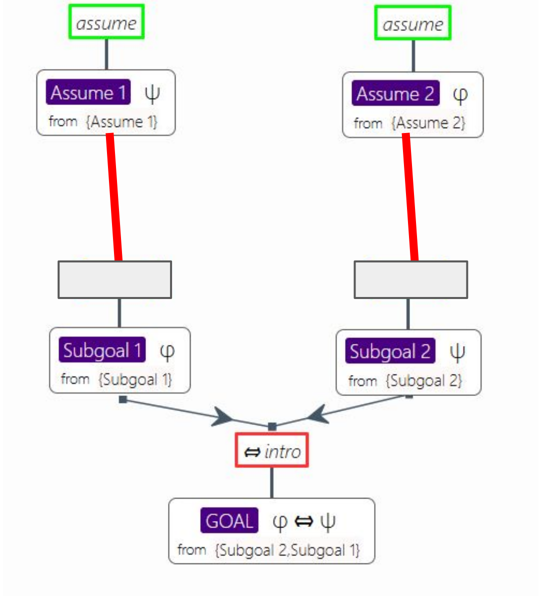
Think about it from a natural language perspective. If we have “A follows from B” well then obviously “A follows from B and C”, adding the extra assumption does not change the fact that we can use B to get A. To do this in Hyperslate we use conjunction intro followed by conjunction elim.



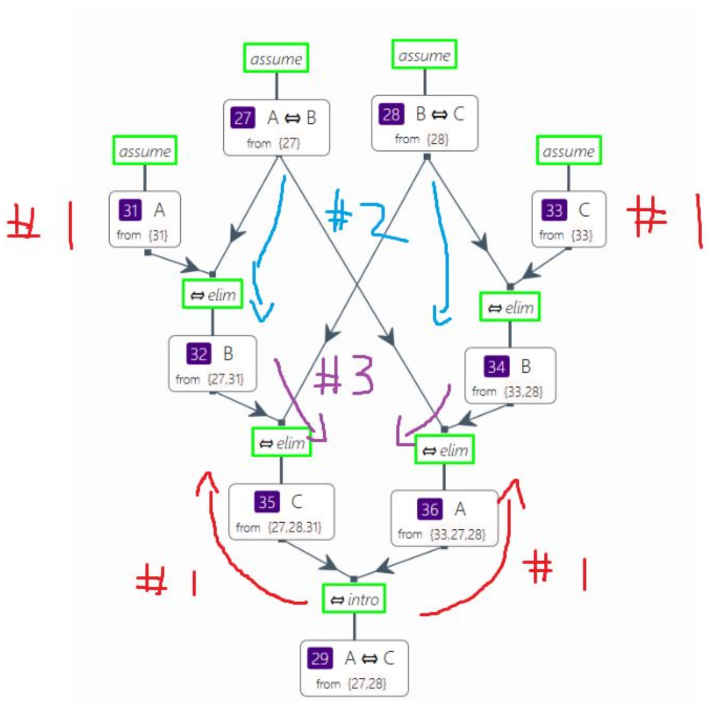
Proving Iff Goals (Bottom Up + Top Down)

If a goal is a biconditional (A ⇔ B), assume A and prove B as one proof, then assume B and prove A as a separate proofs. The first proof (proving B from A) may not use the B assumption and the second proof (proving A from the B) may not use the A assumption. Use ⇔ intro.



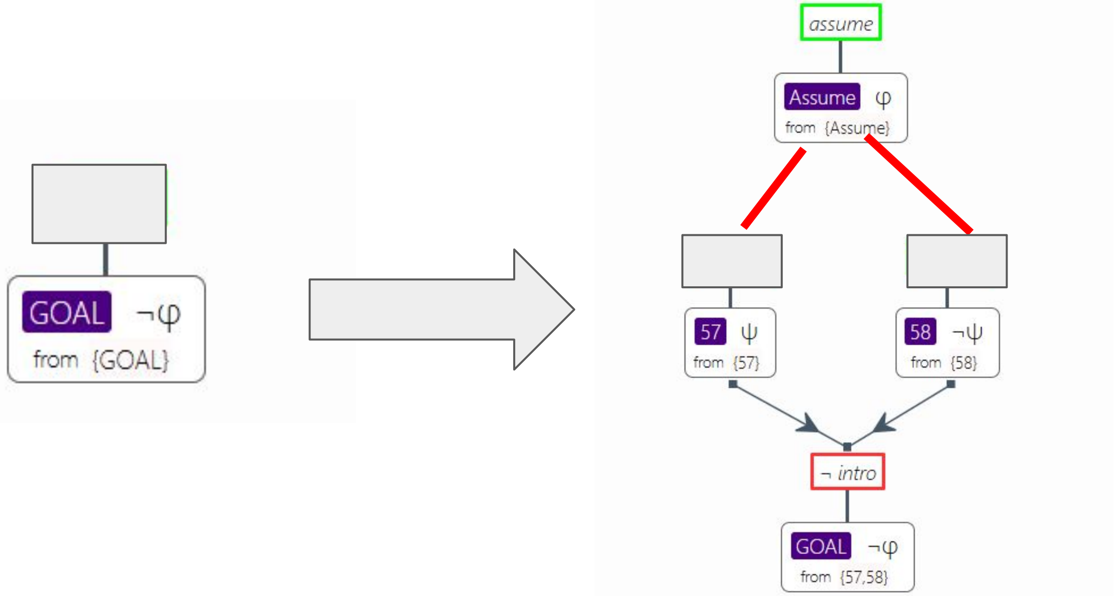


Proving Iff Goals Example



Proving Negated Goals (Bottom Up + Top Down)

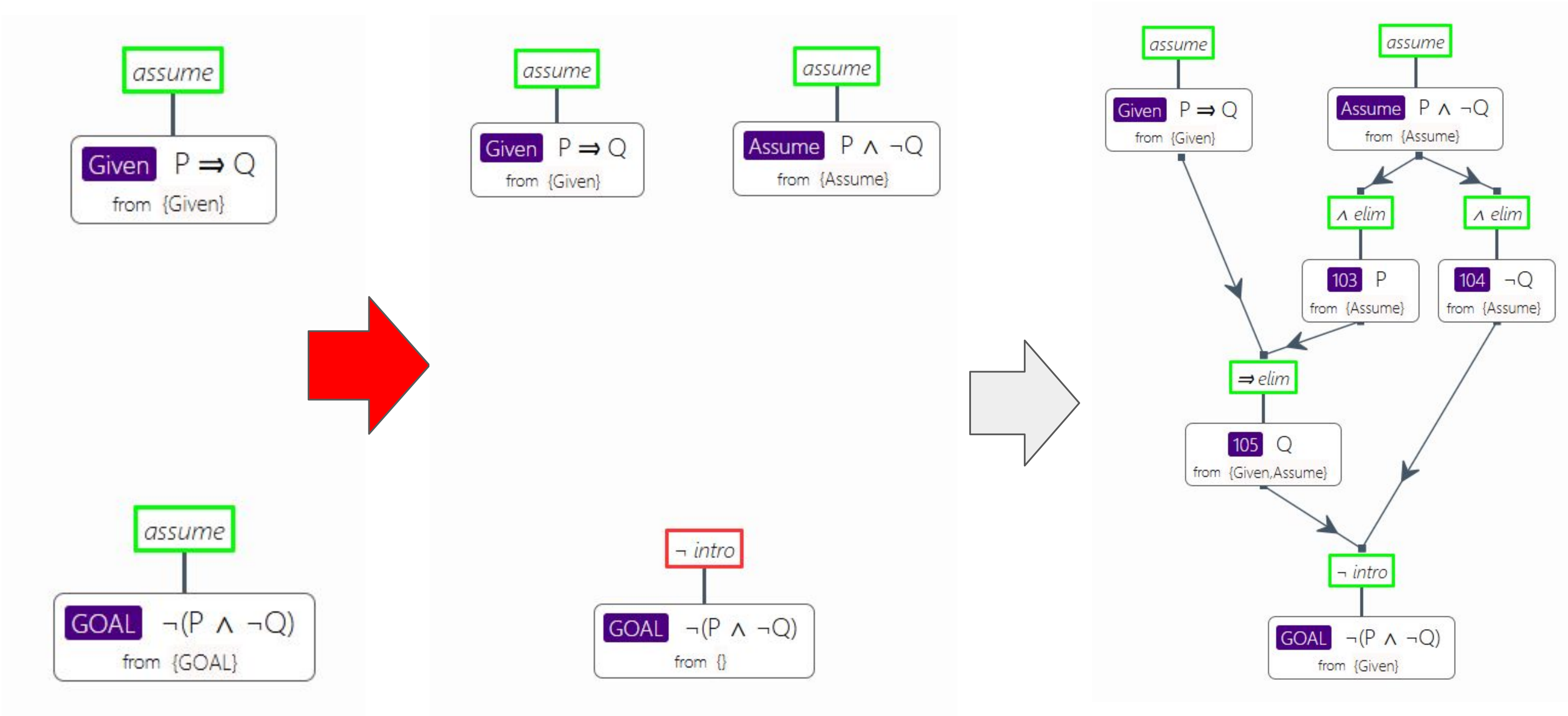
If a goal is negated (¬φ), assume its non-negated form (φ). Using other givens and φ (or sometimes by directly using φ if there are no other implications) prove a contradiction and apply not intro.



Now, use φ and other givens to prove two formulae: ψ and ¬ψ, which you can use to apply not introduction.

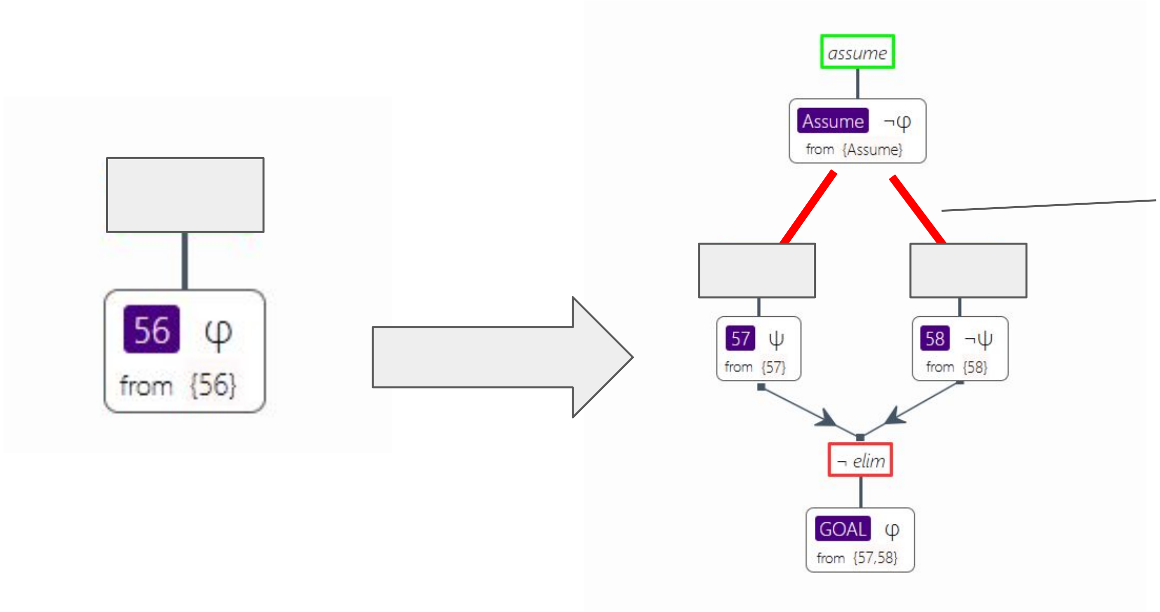
Finding these formulae may require a bit of reasoning about what you can derive from φ and other givens.

Proving Negated Goals Example



Proof By Contradiction (Bottom Up + Top Down)

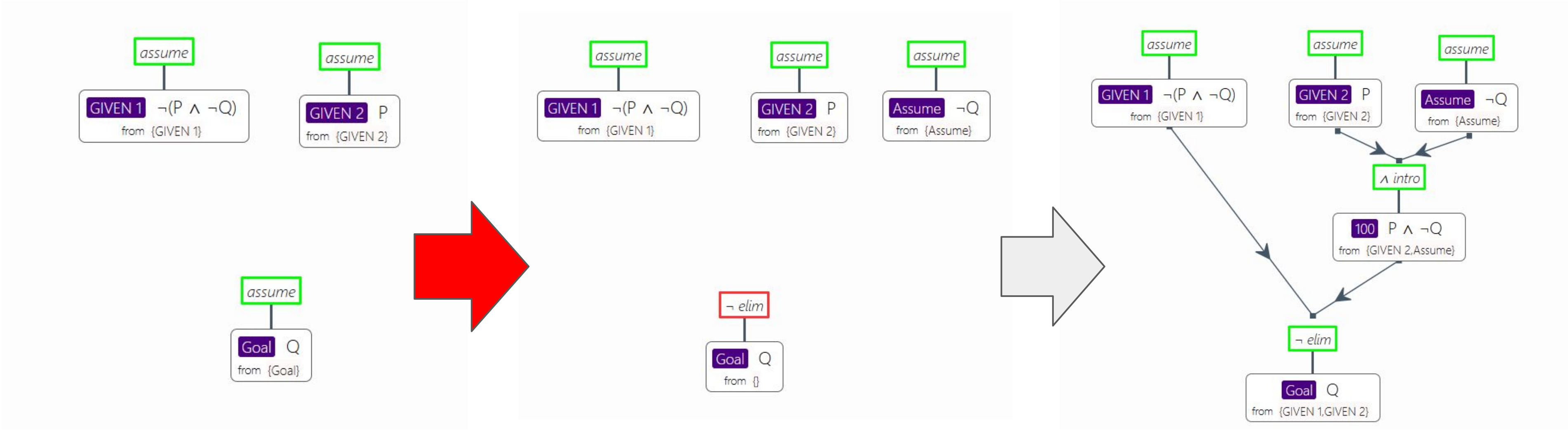
Only try this AFTER you have tried and failed with the other previous rules that split the goal based on its operator. If your goal is ANY formula φ, assume its negation (¬φ) and derive a contradiction with other givens to use not elimination.



Now, use ¬φ and other givens to prove two formulae: ψ and ¬ψ, which you can use to apply not elimination.

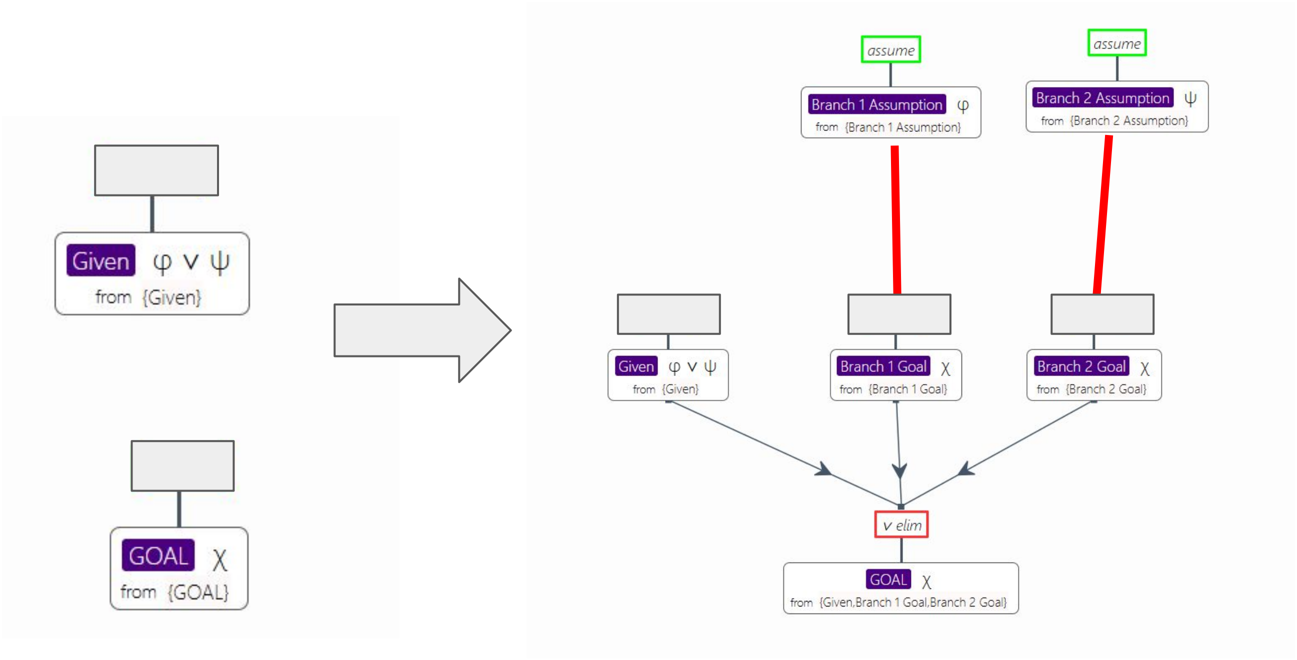
Finding these formulae may require a bit of reasoning about what you can derive from ¬φ and other givens.

Proof By Contradiction Example



Proof By Cases (Given Disjunction)

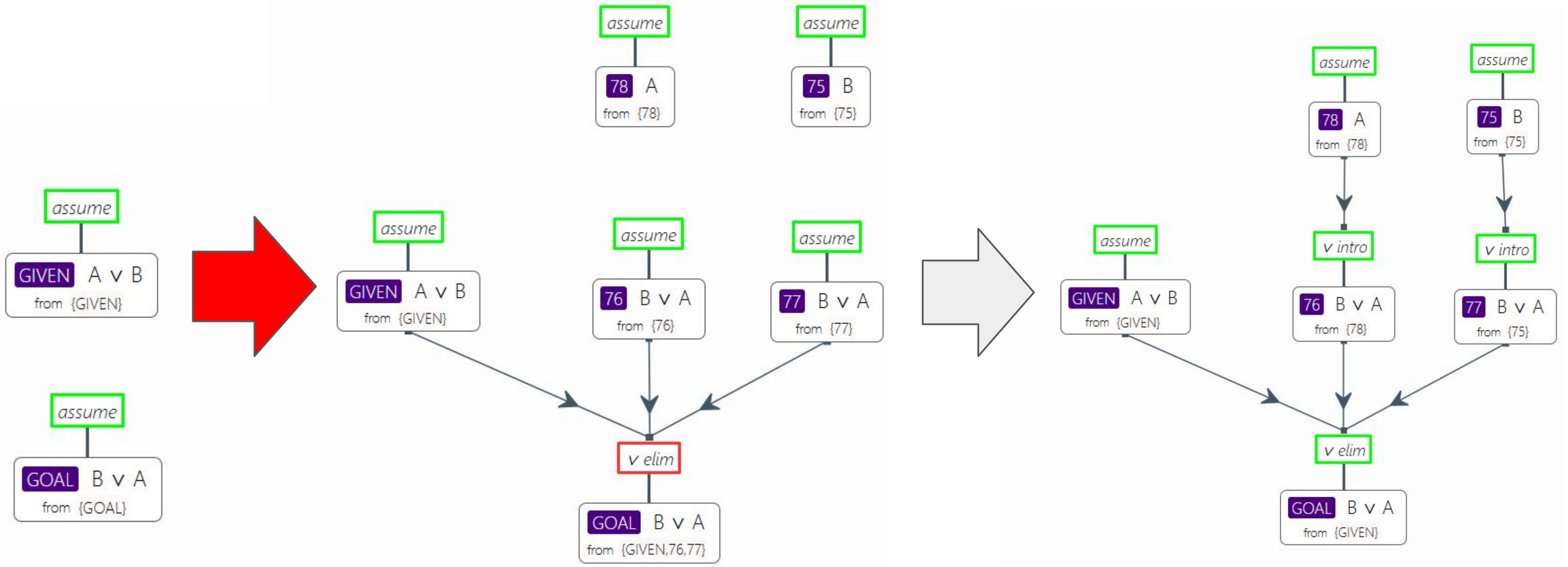
If you are given a disjunction or derive one while building a proof from the top down, there is a chance you will need to apply disjunction elimination and perform a proof by cases. These generally take the following form.



Given φ or ψ with a goal χ. Assume φ and prove χ. Then assume ψ and prove χ, use disjunction elimination with the conclusions and the original φ or ψ.

Proof By Cases Example

From last class: Swap disjunct order, Given A or B prove B or A. Start by noticing we have a disjunctive given! Trying proof by cases is a good idea!



Now: Personalized Problems!

Selmer has uploaded two personalized problems to Hyperslate, due by the end of class at 6 PM today.

Use the strategies just discussed to help complete these proofs. Each person has different (but potentially similar) proofs. Feel free to work with your neighbors and help each other solve them.

I will solve mine live to help walk people through the thought process on one set of them, feel free to ignore me and work on your own proofs. I will also take basic questions about inference rules & strategy, but I am not allowed to solve your proof for you.

You can access this slide deck at: https://bit.ly/4etaD83

Zeroth Order Logic (ZOL)

Zeroth Order Logic adds new features to Propositional Logic (PL):

1) *Terms,* a way to represent objects in logical formulae

2) *Predicates* which allow and relate terms and have a truth value

3) The special *equality predicate* and associated inference rules, allowing us to assert that two terms are equivalent.

Warning: Some sources like Wikipedia will tell you that Zeroth Order Logic is Propositional Logic. This is due to contentions over definitions, some people do not consider the logic we present here to be its own logic, and consider its features to only be available in *First Order Logic* (which we will learn about later). In this class ZOL and PL are not the same, ZOL is a superset of PL.

Terms: Examples from Math

Let’s think about math formulae: some things are terms some things are formulae.

Definition (Terms):

Terms may be constants: 1,2,3,5,6,7, … a, b, c.

Terms may be variables: x,y,z, …

Terms might be functions that combine multiple terms: 1 + 2, 2+4, f(5), g(x,y)

TERMS DO NOT HAVE A TRUTH VALUE

“Is 7 + 2 true or false” is not a sensical statement.

Predicates

Predicates take terms into formulae by assigning a truth value based on the value of the term.

1 + 2 = 3 is a statement with a truth value (its true)

Even(67)   is a statement with a truth value (its true)

f(a + b) = f(a) + f(b) is a statement with a truth value (depending on f)

Animal(x) is a statement that is true if x is an animal.

Naming Conventions

We will typically use lowercase letters at the start of the alphabet or numbers to denote constants: a, b, c, d, …   1, 2, 3, …

We will typically use lowercase letters at the end of the alphabet to denote variables: x, y, z,

For functions we will typically use lowercase letters starting at f (f, g, h), or lowercase words. Using + and * as funcs has special support in Hyperslate to render in infix notation.

We will use uppercase letters or capital words for predicates. The equality predicate (=) has special support and is rendered infix. < and > are also rendered infix.

Exercises

Is the following an constant, function, predicate, or logical connective? For each, specify if they are used at the formula or term level.

- The number 1?
- Addition (+)?
- Equality (=)?
- Conjunction (∧)?
- Divides (|)?
- The string “hello world!”?
- Not equals (≠)?

The Expressive Power of ZOL

We have the power to say more things than ever now:

$\text{Llama}(x) \Rightarrow \text{Animal}(x)$ “if x is a llama then x is an animal”

$\text{Even}(x) \Leftrightarrow \text{Divides}(2, x)$ “x is even if and only if 2 divides x”

$(x = y \land y = z) \Rightarrow x = z$ “if x = y and y = z then x = z (equality is transitive)”

$x = 2 \Rightarrow x + x = x^2$ “if x = 2 then x + x = x ^ 2”

$\text{TA}(x, \text{iblai}) \Rightarrow x = \text{james}$ “if x is the TA of IBLAI, then x is James”

$\text{Hom}(f) \Leftrightarrow f(x + y) = f(x) + f(z)$ “f is a homomorphism on an additive group iff …

Open vs Closed formulae

What is the truth value of of: "x + 2 = 5"? Impossible to determine! The truth value depends on whatever x is.

Definition (Open Formula):
We call a ZOL formulae with variables in terms an "open formulae". Variables in these formulae are called *free variables* (When you learn FOL later on we will have *bound variables*)

What about 1 + 2 = 4 and Even(66)?

Definition (Closed Formula): A formulae with no variables whose truth value can be decided is a closed formulae. Note that to actually determine the truth value of a formulae like this requires givens that define how functions and predicates behave.

The Formal Syntax of Hyperslate ZOL

Name ::= <Any word or character>

Constant ::= Name

Function ::= Name “(“ TermList “)” | Term “+” Term | Term “×” Term

Term ::= Constant | Function

TermList ::= (Term “,”)* Term <A comma separated list of terms, * = Kleene star>

Predicate ::= Name “(“ TermList “)” | Term “=” Term

Atom ::= Name

Formula ::= Atom | Predicate | Formula ∧ Formula |

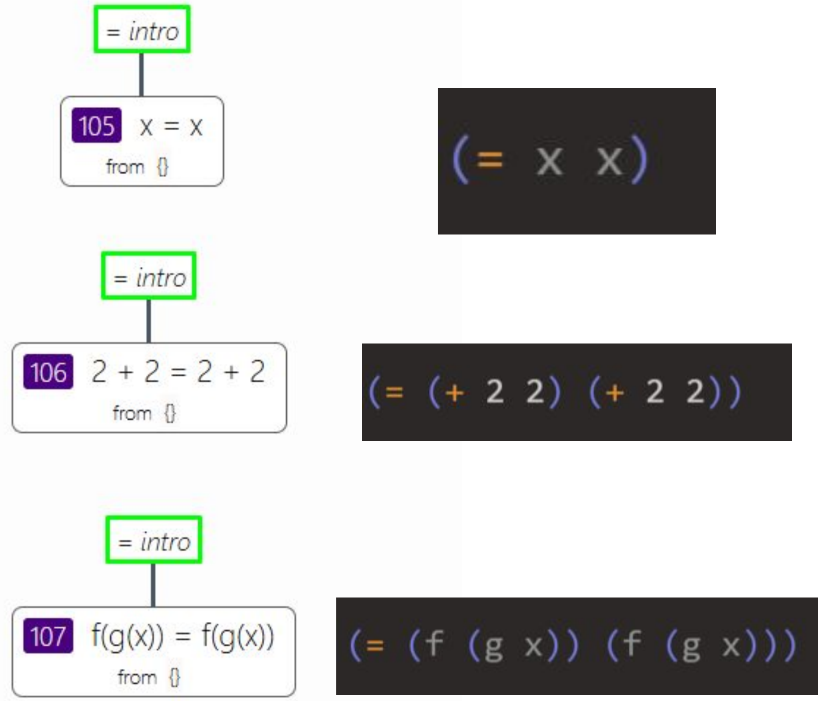
Equality Introduction

Premices: None

Conclusions: x = x from {}

You can always say given nothing
that something being equal to itself
is a theorem (note how “from” is empty!)

Don’t “assume” this, if you assume it you
are saying “given x = x, i can prove x = x”
instead of “x = x is a theorem”

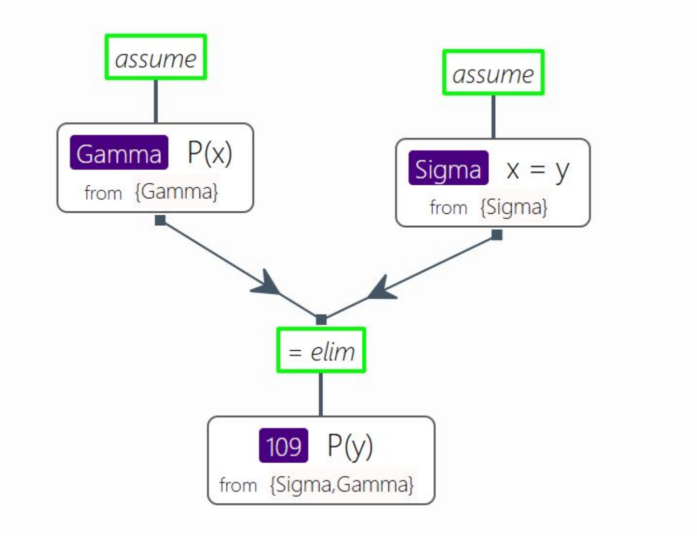


Equality Elimination

You can use equality to substitute any term with an equivalent term. In the following notation "φ[x]" is "A formula φ containing a term x" and "φ[x/y]" is "φ after replacing all 'x's with 'y's "

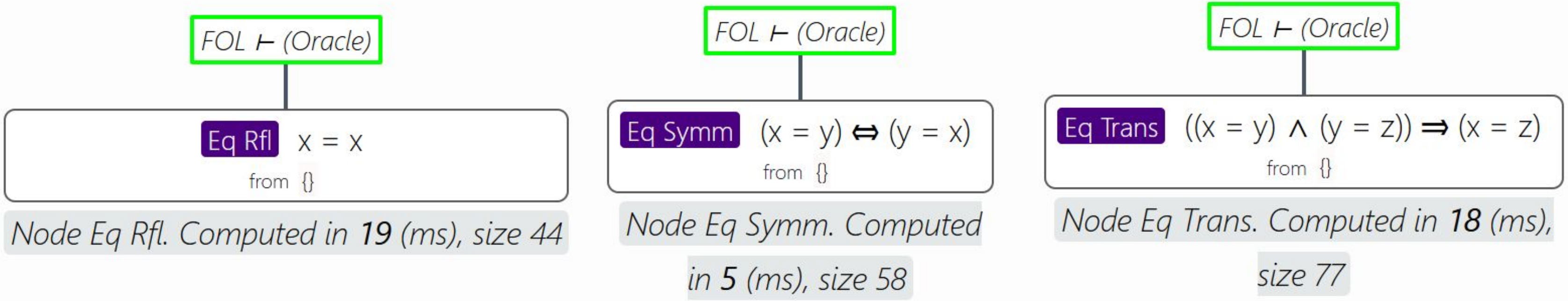
Premicies: "φ[x] is provable from Γ"
"x = y is provable from Σ"

Conclusions: "φ[x/y] is provable from Γ and Σ"

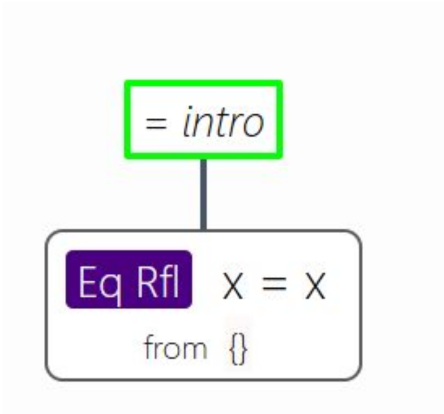


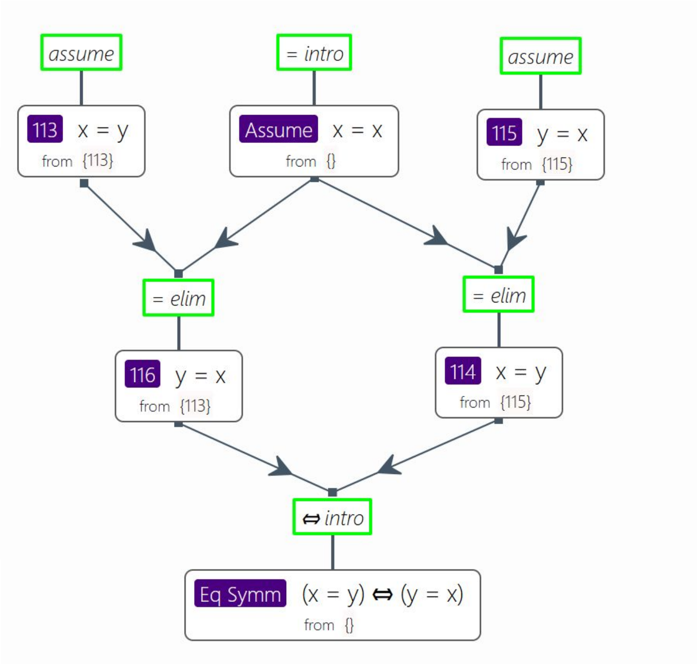
Equality Rules Exercises

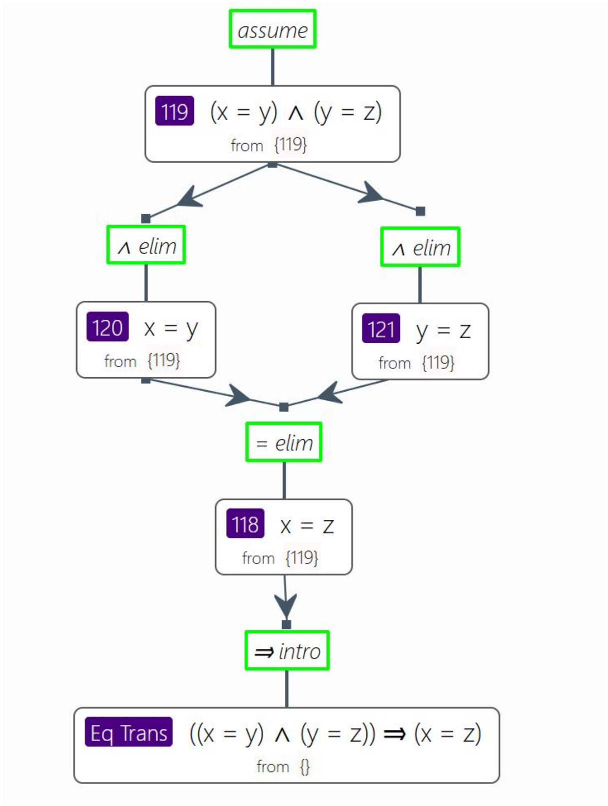
Prove equality is reflexive, transitive, and symmetric.



Equality Rules Exercise Solutions







Rest of Class : Try these

Addition Property

FOL $\vdash$ (Oracle)

Addition Property $(a = b) \Rightarrow (a + c = b + c)$

from {}

*Node Addition Property. Computed in **6** (ms), size 76*

Green Cheese Moon

PC $\vdash$ (Oracle)

89 $\neg(P \Rightarrow (Q \Rightarrow P)) \Rightarrow G$

from {}

*Node 89. Computed in **4** (ms), size 80*

Full Demorgan's Laws

PC $\vdash$ (Oracle)

Demorgans $(\neg(A \wedge B) \Leftrightarrow (\neg A \vee \neg B)) \wedge (\neg(A \vee B) \Leftrightarrow (\neg A \wedge \neg B))$

from {}

*Node Demorgans. Computed in **11** (ms), size 133*

Lean4 congr

FOL $\vdash$ (Oracle)

126 $((a = b) \wedge (f(a) = g(a))) \Rightarrow (f(a) = g(b))$

from {}

*Node 126. Computed in **8** (ms), size 97*

Distribution of Or over And

PC $\vdash$ (Oracle)

91 $(P \vee (Q \wedge R)) \Leftrightarrow ((P \vee Q) \wedge (P \vee R))$

from {}

*Node 91. Computed in **15** (ms), size 88*

If you finish early, go back to previous problems (https://bit.ly/4etaD83), can ChatGPT provide correct natural deduction proofs for them? Can Bard? In your prompt feel free to include the list of rules in our natural deduction system in the ReadMe of lazyslate (https://github.com/RAIRLab/lazyslate). Email me your results, oswalj@rpi.edu