A Theorem Proving Assistant

Joe Duffin

School of Computer Science
University College Dublin

20th March 2017

Overview

- What is Theorem Proving
- What I Built
- How does it work

What is a Theorem?

A Theorem is a proposition which is not necessarily self-evident but can be proved with a chain of reasoning.

Theorem ($\vee$ *zero*)

$P \vee true \equiv true$

What is Theorem Proving?

Proof of $\vee zero$

$$
\begin{array}{ll}
 & P \vee true \\
\equiv & \{(X ::= P).(0)\} \\
 & P \vee (P \equiv P) \\
\equiv & \{(X, Y, Z := P, P, P).(1)\} \\
 & P \vee P \equiv P \vee P \\
\equiv & \{(X := P).(2)\} \\
 & P \equiv P \\
\equiv & \{(X := P).(0)\} \\
 & true
\end{array}
$$

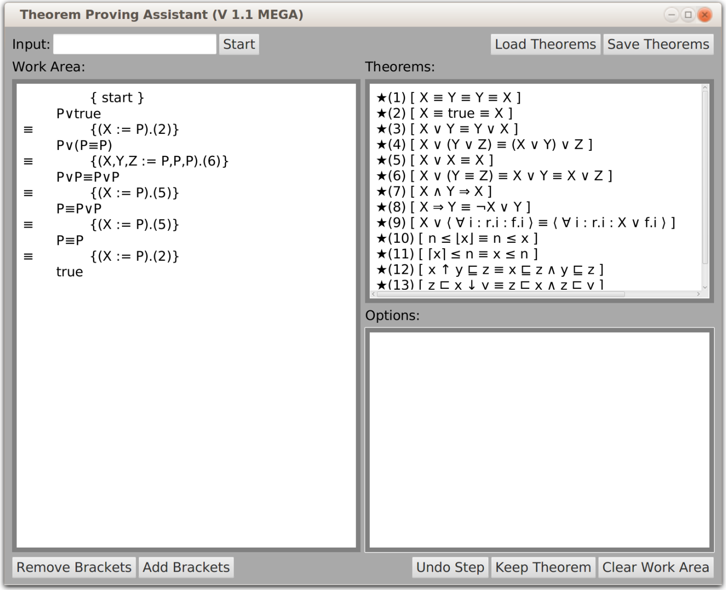
Theorems

$(0)[X \equiv X \equiv true]$

$(1)[X \vee (Y \equiv Z) \equiv X \vee Y \equiv X \vee Z]$

$(2)[X \vee X \equiv X]$

What I Built

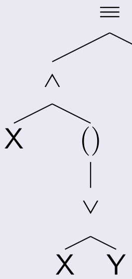

Theorem Proving Assistant (V 1.1 MEGA)
Input:
Start
Load Theorems
Save Theorems
Work Area:
{ start }
P∨true
≡ {(X := P).(2)}
P∨(P≡P)
≡ {(X,Y,Z := P,P,P).(6)}
P∨P≡P∨P
≡ {(X := P).(5)}
P≡P∨P
≡ {(X := P).(5)}
P≡P
≡ {(X := P).(2)}
true
Theorems:
★(1) [X ≡ Y ≡ Y ≡ X]
★(2) [X ≡ true ≡ X]
★(3) [X ∨ Y ≡ Y ∨ X]
★(4) [X ∨ (Y ∨ Z) ≡ (X ∨ Y) ∨ Z]
★(5) [X ∨ X ≡ X]
★(6) [X ∨ (Y ≡ Z) ≡ X ∨ Y ≡ X ∨ Z]
★(7) [X ∧ Y ⇒ X]
★(8) [X ⇒ Y ≡ ¬X ∨ Y]
★(9) [X ∨ ⟨ ∀ i : r.i : f.i ⟩ ≡ ⟨ ∀ i : r.i : X ∨ f.i ⟩]
★(10) [n ≤ ⌊x⌋ ≡ n ≤ x]
★(11) [⌈x⌉ ≤ n ≡ x ≤ n]
★(12) [x ↑ y ⊑ z ≡ x ⊑ z ∧ y ⊑ z]
★(13) [z ⊏ x ↓ y ≡ z ⊏ x ∧ z ⊏ y]
Options:
Remove Brackets
Add Brackets
Undo Step
Keep Theorem
Clear Work Area

How it Works - Expression Representation

String Representation ($Absorbtion0$)

$X \wedge (X \vee Y) \equiv X$

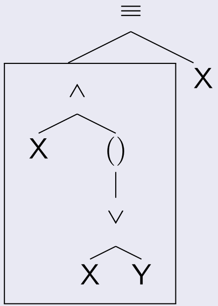
Tree Representation ($Absorbtion0$)



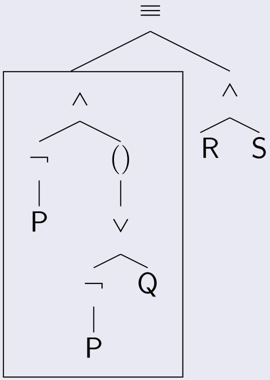
- Syntax trees are used to represent expressions.

How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$



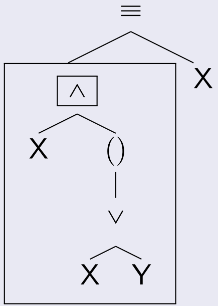
User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



Look Up Table

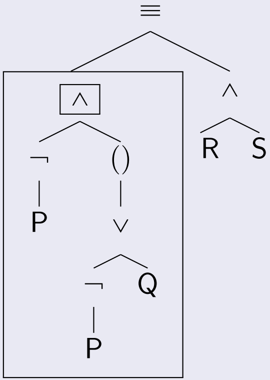
How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$



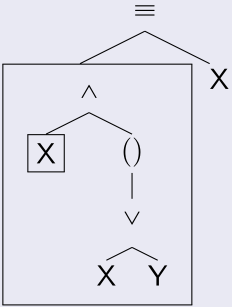
Look Up Table

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



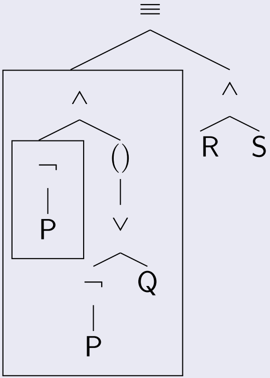
How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$



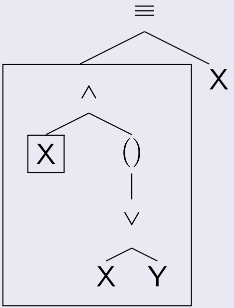
Look Up Table

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$

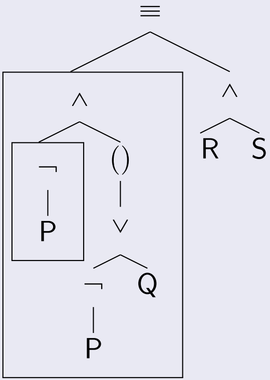


How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$



User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$

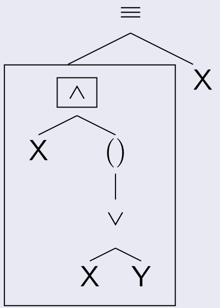


Look Up Table

$\boxed{X := \neg P}$

How it Works - Pattern Matching

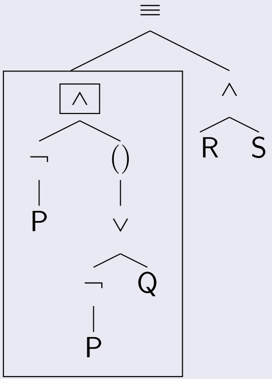
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

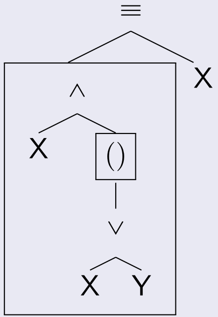
$X \coloneqq \neg P$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

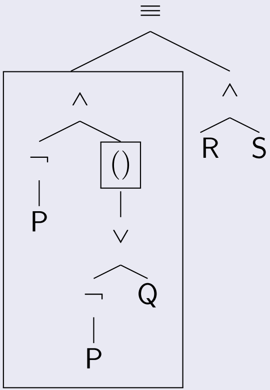
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

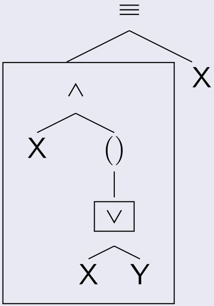
$X \coloneqq \neg P$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

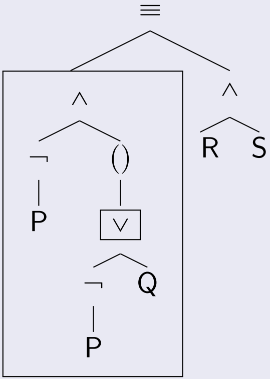
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

$X := \neg P$

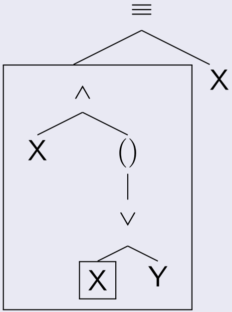
User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

Rule:

$X \wedge (X \vee Y) \equiv X$

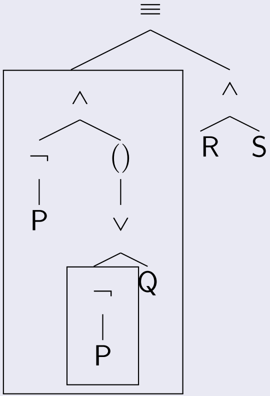


Look Up Table

$X := \neg P$

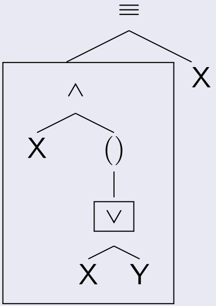
User Expression:

$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

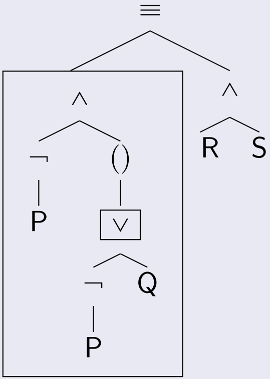
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

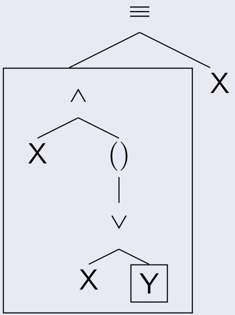
$X \coloneqq \neg P$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

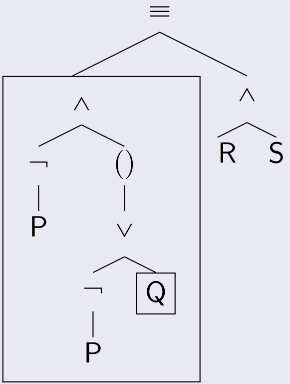
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

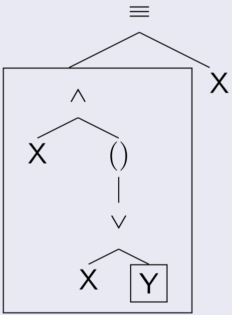
$X := \neg P$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

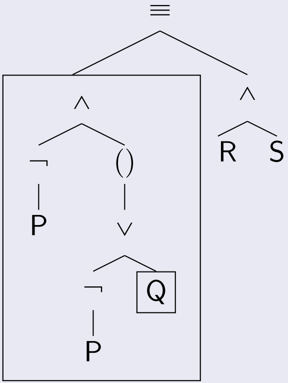
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

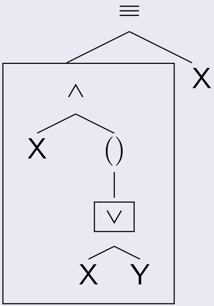
$X := \neg P$
$\boxed{Y := Q}$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

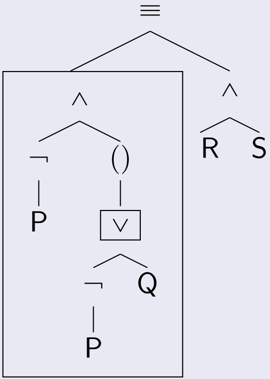
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

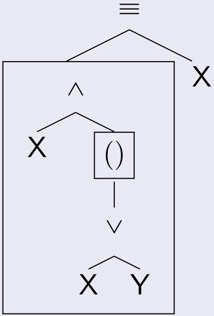
$X := \neg P$
$Y := Q$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$

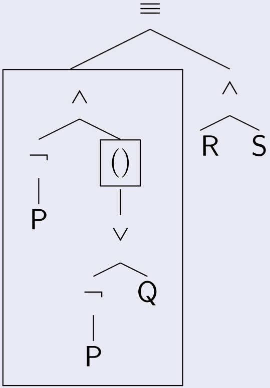


Look Up Table

$X := \neg P$

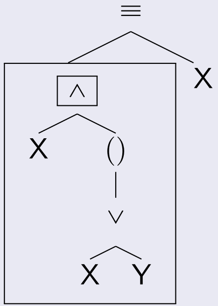
$Y := Q$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

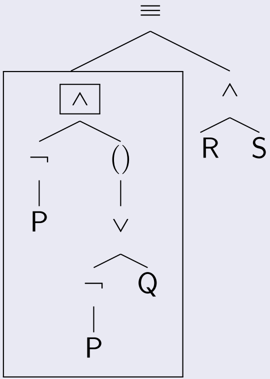
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

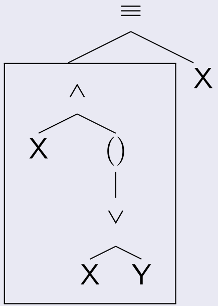
$X := \neg P$
$Y := Q$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

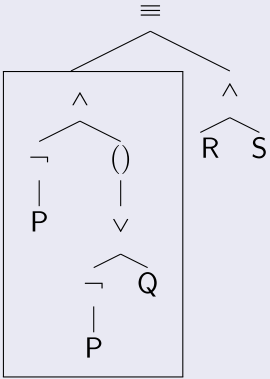
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

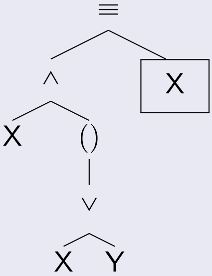
$X := \neg P$
$Y := Q$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$



How it Works - Pattern Matching

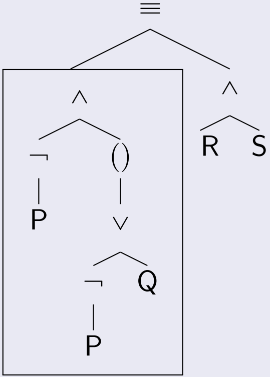
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

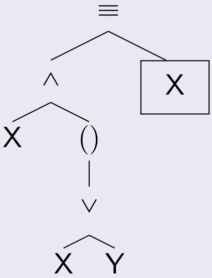
$X := \neg P$
$Y := Q$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$

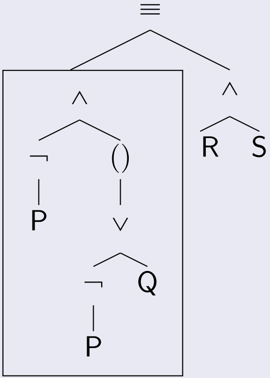


How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$



User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$

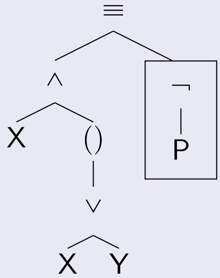


Look Up Table

$\boxed{X := \neg P}$
$Y := Q$

How it Works - Pattern Matching

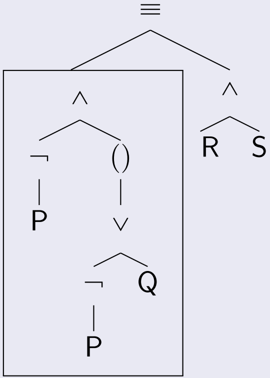
Rule:
$X \wedge (X \vee Y) \equiv X$



Look Up Table

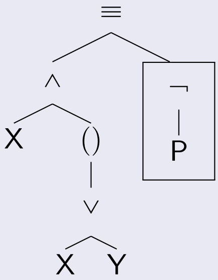
$\boxed{X := \neg P}$
$Y := Q$

User Expression:
$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$

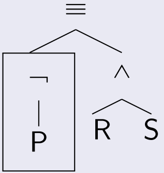


How it Works - Pattern Matching

Rule:
$X \wedge (X \vee Y) \equiv X$



User Expression:
$\neg P \equiv R \wedge S$



Look Up Table

$X := \neg P$
$Y := Q$

How it Works - Pattern Matching

Previous Expression

$\neg P \wedge (\neg P \vee Q) \equiv R \wedge S$

Look Up Table

$X := \neg P$

$Y := Q$

New Expression

$\neg P \equiv R \wedge S$

- The new user expression and lookup table are used to generate the hint and next line of the proof.

The Step

$$
\begin{array}{ll}
 & \neg P \wedge (\neg P \vee Q) \equiv R \wedge S \\
\equiv & \quad \{(X, Y ::= \neg P, Q).Abs0\} \\
 & \neg P \equiv R \wedge S
\end{array}
$$

Versatility - Supported Calculi

- Boolean
- Floor/Ceiling
- Max/Min
- Lattice Theory
- Quantified Notation

Questions...