Automated Theorem Proving

-- A Deep Learning Approach

Industrial Sponsor: Real AI

Gwang Hyeon CHOI, Seung Yong MOON, Zheng PAN, Justin SUN

fppt.com

Outline

- Introduction
- Metamath: The Foundation
- Holophrasm: An Overview
- Future Work

fppt.com

Outline

- Introduction
- Metamath: The Foundation
- Holophrasm: An Overview
- Future Work

fppt.com

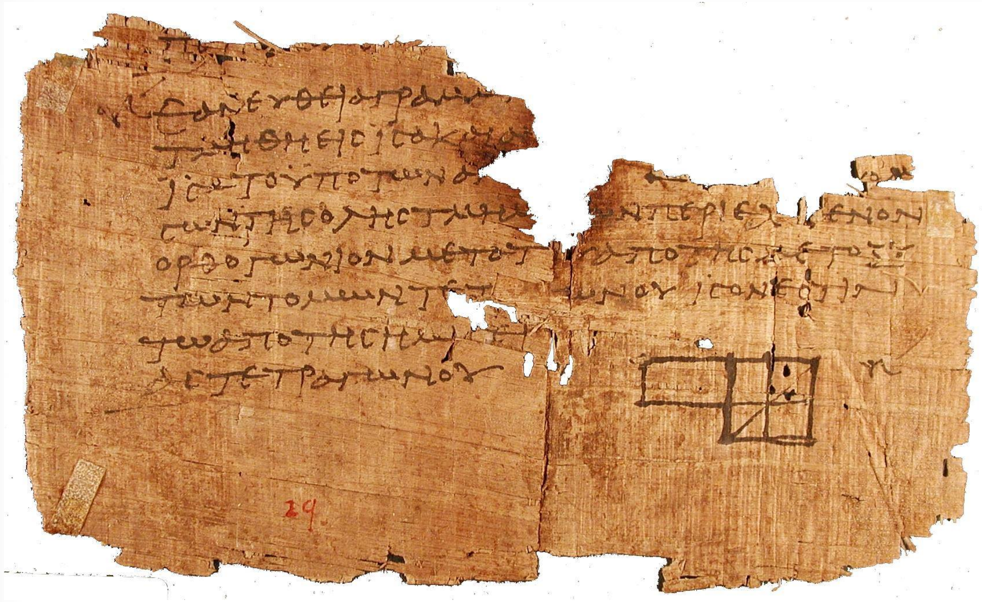
Company Background

Real AI

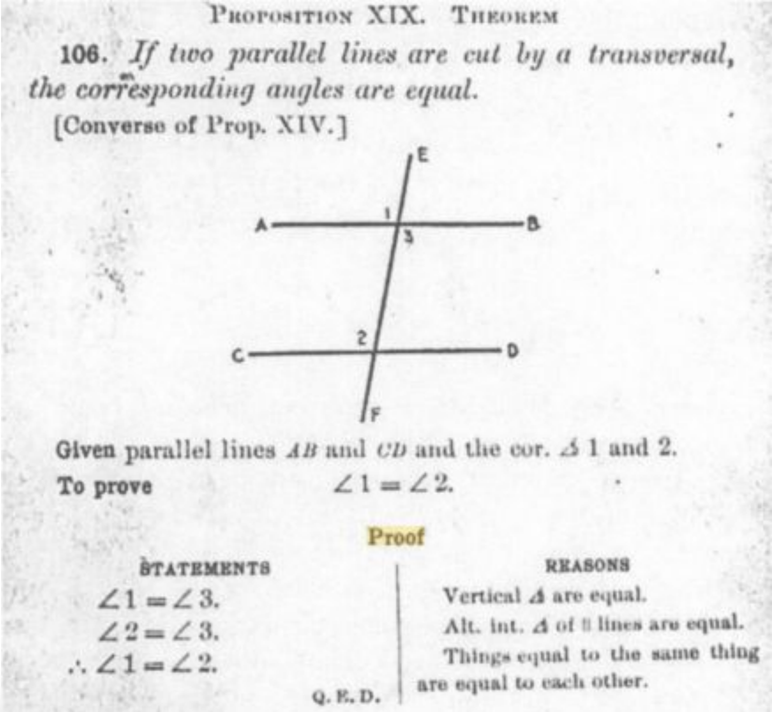
- Research & development of safe and beneficial artificial general intelligence
- Altruistic intelligence for problem solving

fppt.com

Proving Math Theorems ...



https://en.wikipedia.org/wiki/Mathematical_proof#/media/File:P._Oxy._I_29.jpg



PROPOSITION XIX. THEOREM

106. *If two parallel lines are cut by a transversal, the corresponding angles are equal.*

[Converse of Prop. XIV.]

Given parallel lines AB and CD and the cor. $\angle$s 1 and 2.

To prove $\angle 1 = \angle 2$.

Proof

STATEMENTS	REASONS
$\angle 1 = \angle 3$.	Vertical $\angle$s are equal.
$\angle 2 = \angle 3$.	Alt. int. $\angle$s of ∥ lines are equal.
$\therefore \angle 1 = \angle 2$.	Things equal to the same thing are equal to each other.

Q. E. D.

Deep Learning

- Language translation
- Image recognition
- Alphago

.
.
.

- **Mathematics??**

https://www.theguardian.com/technology/2017/may/23/alphago-google-ai-beats-ke-jie-china-go

fppt.com

Outline

- Introduction
- Metamath: The Foundation
- Holophrasm
- Future Work

fppt.com

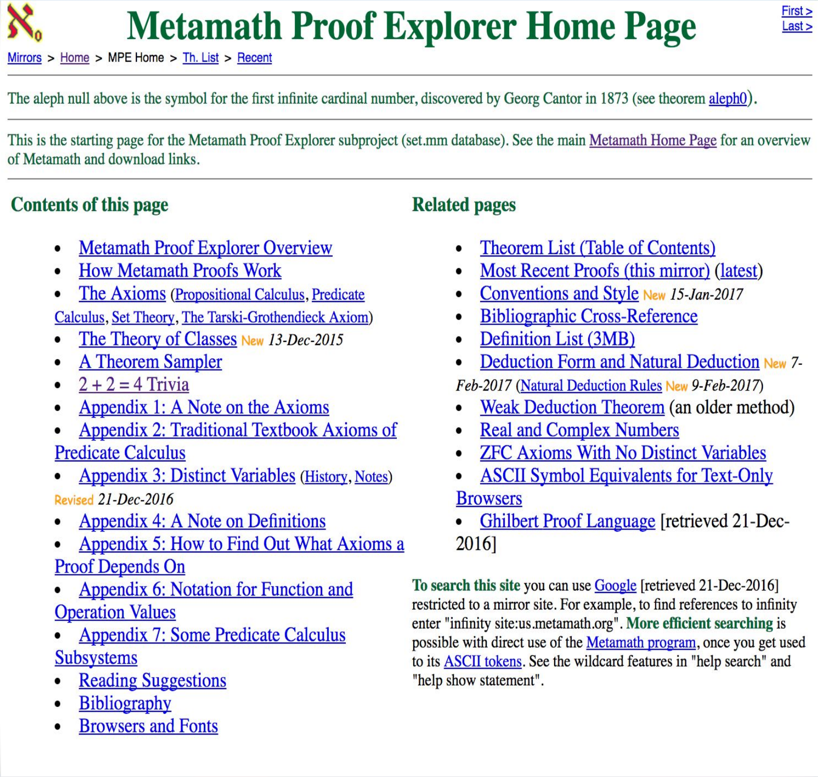
Formal Systems

- Enables computers to work on proofs
- Metamath, Mizar, HOL Light

fppt.com

Metamath

- Over 19,000 proofs
- Built from 22 axioms
  - Logic + ZFC axioms



Metamath Proof Explorer Home Page

First >
Last >

Mirrors > Home > MPE Home > Th. List > Recent

The aleph null above is the symbol for the first infinite cardinal number, discovered by Georg Cantor in 1873 (see theorem aleph0).

This is the starting page for the Metamath Proof Explorer subproject (set.mm database). See the main Metamath Home Page for an overview of Metamath and download links.

Contents of this page

Related pages

- Theorem List (Table of Contents)
- Most Recent Proofs (this mirror) (latest)
- Conventions and Style New 15-Jan-2017
- Bibliographic Cross-Reference
- Definition List (3MB)
- Deduction Form and Natural Deduction New 7-Feb-2017 (Natural Deduction Rules New 9-Feb-2017)
- Weak Deduction Theorem (an older method)
- Real and Complex Numbers
- ZFC Axioms With No Distinct Variables
- ASCII Symbol Equivalents for Text-Only Browsers
- Ghilbert Proof Language [retrieved 21-Dec-2016]

To search this site you can use Google [retrieved 21-Dec-2016] restricted to a mirror site. For example, to find references to infinity enter "infinity site:us.metamath.org". **More efficient searching** is possible with direct use of the Metamath program, once you get used to its ASCII tokens. See the wildcard features in "help search" and "help show statement".

Dissecting a proof: t=t

- Two axioms:

$$\text{(A1)} \qquad (t = r \to (t = s \to r = s))$$

$$\text{(A2)} \qquad (t + 0) = t$$

- Modus Ponens:

$$\text{(MP)} \qquad ((P \wedge (P \to Q)) \to Q)$$

fppt.com

Dissecting a proof: t=t

- Required 'tools'
- Assertion

```
th1 $p |- t = t
```

- Proof

```
tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
tt weq tt tze tpl tt weq tt tt weq wim tt a2
tt tze tpl tt tt a1 mp mp
```

```
$( Declare the constant symbols we will use $)
    $c 0 + = -> ( ) term wff |- $.
$( Declare the metavariables we will use $)
    $v t r s P Q $.
$( Specify properties of the metavariables $)
    tt $f term t $.
    tr $f term r $.
    ts $f term s $.
    wp $f wff P $.
    wq $f wff Q $.
$( Define "term" (part 1) $)
    tze $a term 0 $.
$( Define "term" (part 2) $)
    tpl $a term ( t + r ) $.
$( Define "wff" (part 1) $)
    weq $a wff t = r $.
$( Define "wff" (part 2) $)
    wim $a wff ( P -> Q ) $.
$( State axiom a1 $)
    a1 $a |- ( t = r -> ( t = s -> r = s ) ) $.
$( State axiom a2 $)
    a2 $a |- ( t + 0 ) = t $.
    ${
       min $e |- P $.
       maj $e |- ( P -> Q ) $.
$( Define the modus ponens inference rule $)
       mp  $a |- Q $.
    $}
$( Prove a theorem $)
    th1 $p |- t = t $=
  $( Here is its proof: $)
       tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
       tt weq tt tze tpl tt weq tt tt weq wim tt a2
       tt tze tpl tt tt a1 mp mp
     $.
```

Dissecting a proof: t=t

- Reverse Polish Notation (RPN)

Usual notation	RPN
2+3	23+
4+6-8	46+8-

- Hypotheses: operands
- Theorems: operators

fppt.com

Dissecting a proof: t=t

Proof	**tt tze tpl tt weq … a1 mp mp**

Stack

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	**tt** **tze** **tpl** **tt** **weq** … **a1** **mp** **mp**

`tt $f term t $.`

A hypothesis

Stack

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	**tt tze tpl tt weq … a1 mp mp**

`term t`

Stack

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	**tt** $\boxed{\textbf{tze}}$ **tpl tt weq … a1 mp mp**

```
tze $a term 0 $.
```

An axiom with a CONSTANT

```
term t
```

Stack

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	**tt tze tpl tt weq … a1 mp mp**

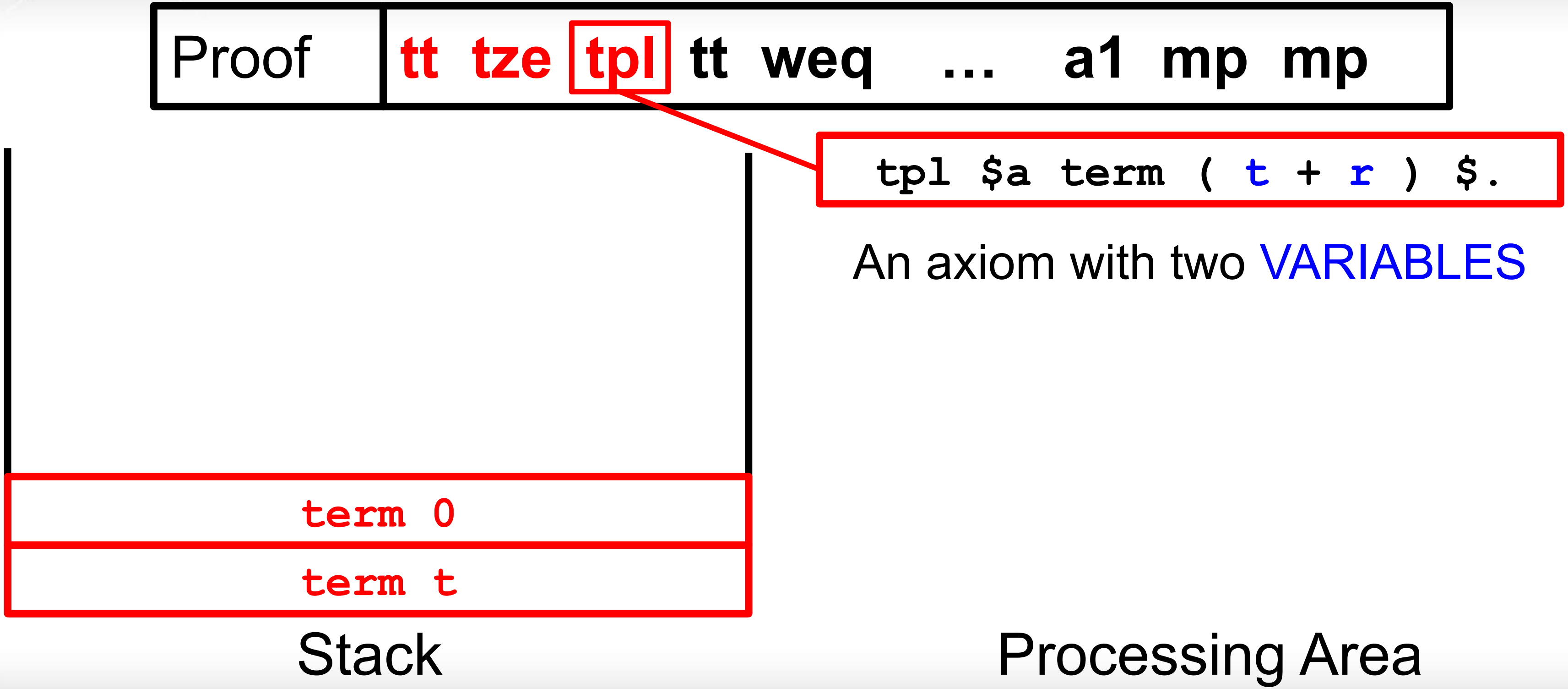
```
term 0
term t
```

Stack

Processing Area

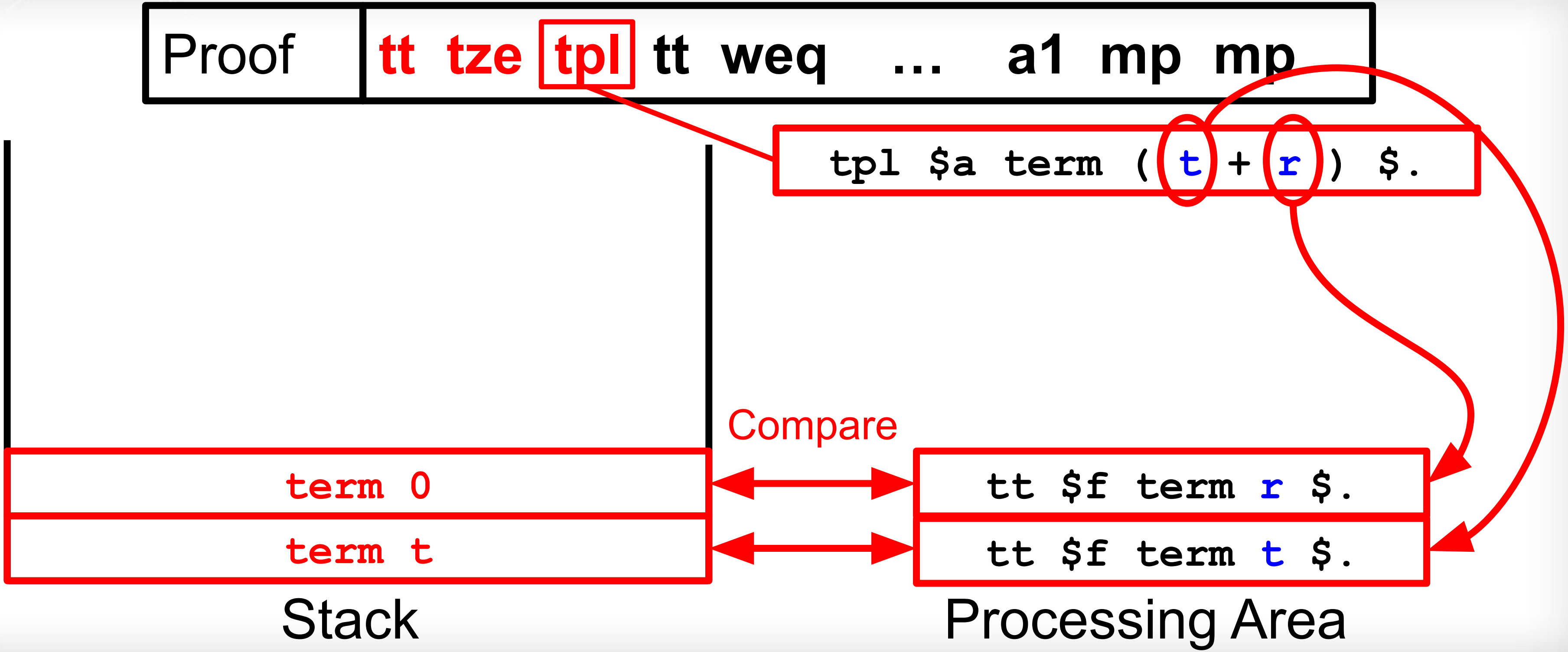
fppt.com

Dissecting a proof: t=t



fppt.com

Dissecting a proof: t=t



Dissecting a proof: t=t

Proof	tt tze tpl tt weq ... a1 mp mp

```
tpl $a term ( t + r ) $.
```

Perform Substitution

```
Replace t by t
Replace r by 0
```

```
term 0
term t
```

Stack

```
term ( t + 0 )
```

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	tt tze tpl tt weq … a1 mp mp

Remove the objects which were used to perform the substitution

Stack

```
term ( t + 0 )
```

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	tt tze tpl tt weq … a1 mp mp

```
term ( t + 0 )
```

Stack

Processing Area

fppt.com

Dissecting a proof: t=t

Proof	tt tze tpl tt weq … a1 mp mp

```
term t
term ( t + 0 )
```

Stack

fppt.com

Dissecting a proof: t=t

Proof	tt tze tpl tt weq … a1 mp mp

```
wff ( t + 0 ) = t
```

Stack

fppt.com

Dissecting a proof: t=t

Proof	**tt tze tpl tt weq … a1 mp mp**

`|- ( ( t + 0 ) = t -> ( ( t + 0 ) = t -> t = t ) )`
`|- ( t + 0 ) = t`
`wff ( ( t + 0 ) = t -> t = t )`
`wff ( t + 0 ) = t`
`|- ( t + 0 ) = t`
`wff t = t`
`wff ( t + 0 ) = t`

Stack

fppt.com

Dissecting a proof: t=t

Proof	tt tze tpl tt weq ... a1 mp **mp**

|- ((t + 0) = t -> t = t))

|- (t + 0) = t

wff t = t

wff (t + 0) = t

Stack

fppt.com

Dissecting a proof: t=t

Proof	tt tze tpl tt weq … a1 mp mp

|- t = t

Stack

fppt.com

Outline

- Introduction
- Metamath: The Foundation
- Holophrasm: An Overview
- Future Work

fppt.com

Holophrasm

Daniel P.Z. Whalen (2016), Holophrasm: a neural Automated Theorem Prover for higher-order logic. : arxiv:1608.02644

Github : https://github.com/dwhalen/holophrasm

Automated theorem prover based on metamath, python, and RNN.

Found proofs for 388 of 2720, or 14.3% of the test propositions

fppt.com

Proof Tree

```
$( Declare the constant symbols we will use $)
    $c 0 + = -> ( ) term wff |- $.
$( Declare the metavariables we will use $)
    $v t r s P Q $.
$( Specify properties of the metavariables $)
    tt $f term t $.
    tr $f term r $.
    ts $f term s $.
    wp $f wff P $.
    wq $f wff Q $.
$( Define "term" (part 1) $)
    tze $a term 0 $.
$( Define "term" (part 2) $)
    tpl $a term ( t + r ) $.
$( Define "wff" (part 1) $)
    weq $a wff t = r $.
$( Define "wff" (part 2) $)
    wim $a wff ( P -> Q ) $.
$( State axiom a1 $)
    a1 $a |- ( t = r -> ( t = s -> r = s ) ) $.
$( State axiom a2 $)
    a2 $a |- ( t + 0 ) = t $.
    ${
       min $e |- P $.
       maj $e |- ( P -> Q ) $.
$( Define the modus ponens inference rule $)
       mp  $a |- Q $.
    $}
$( Prove a theorem $)
    th1 $p |- t = t $=
  $( Here is its proof: $)
       tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
       tt weq tt tze tpl tt weq tt tt weq wim tt a2
       tt tze tpl tt tt a1 mp mp
     $.
```

fppt.com

Proof Tree

```
$( State axiom a1 $)
    a1 $a |- ( t = r -> ( t = s -> r = s ) ) $.
$( State axiom a2 $)
    a2 $a |- ( t + 0 ) = t $.
    ${
       min $e |- P $.
       maj $e |- ( P -> Q ) $.
$( Define the modus ponens inference rule $)
       mp  $a |- Q $.
    $}
$( Prove a theorem $)
    th1 $p |- t = t $=
  $( Here is its proof: $)
       tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
       tt weq tt tze tpl tt weq tt tt weq wim tt a2
       tt tze tpl tt tt a1 mp mp
     $.
```

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$
$(t = s \Rightarrow r = s)$

```
$( State axiom a1 $)
    a1 $a |- ( t = r -> ( t = s -> r = s ) ) $.
```

fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(\, t = s \Rightarrow r = s \,)$

Axiom a2

Assertion : $\vdash (\, t + 0 \,) = t$

```
$( State axiom a2 $)
    a2 $a |- ( t + 0 ) = t $.
```

fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

```
  ${
     min $e |- P $.
     maj $e |- ( P -> Q ) $.
$( Define the modus ponens inference rule $)
     mp  $a |- Q $.
  $}
```

fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash t = t$

```
$( Prove a theorem $)
    th1 $p |- t = t $=
```

fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

$\boxed{\text{Assertion : } \vdash t = t}$

fppt.com

Proof Tree

$\alpha = \alpha$

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + \mathbf{0}) = t$

Modus Ponens(mp)

Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

$\boxed{\text{Assertion : } \vdash \alpha = \alpha}$

fppt.com

Proof Tree

$\alpha = \alpha$

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$

```
$( Prove a theorem $)
    th1 $p |- t = t $=
  $( Here is its proof: $)
       tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
       tt weq tt tze tpl tt weq tt tt weq wim tt a2
       tt tze tpl tt tt a1 mp mp
    $.
```

fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

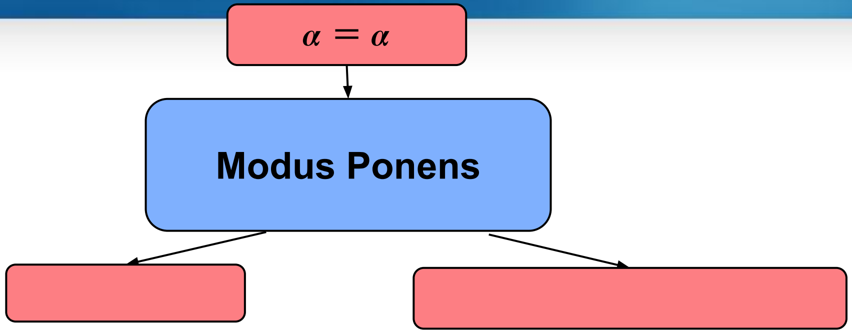
Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$



Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow (t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

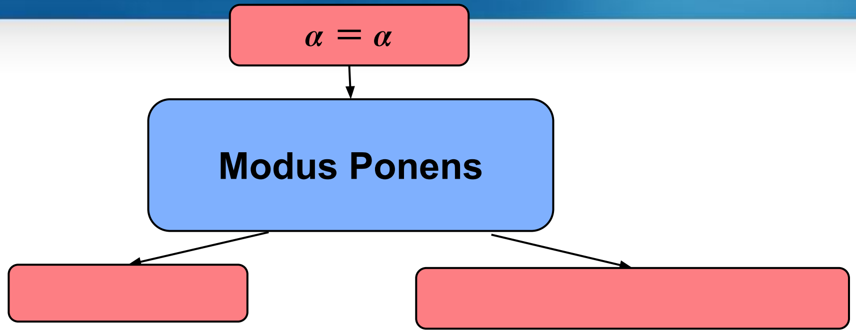
Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$




```
$( Here is its proof: $)
    tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
    tt weq tt tze tpl tt weq tt tt weq wim tt a2
    tt tze tpl tt tt a1 mp mp
```

fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$
$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

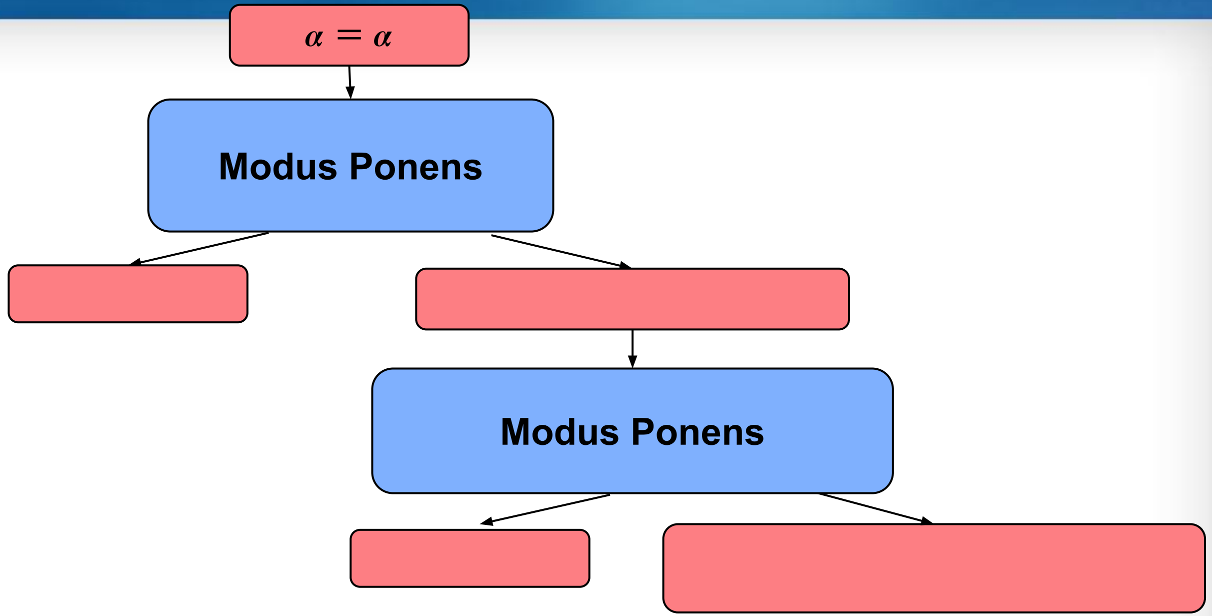
Modus Ponens(mp)

Hypotheses : $\vdash P$
$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$



fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$
$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + \mathbf{0}) = t$

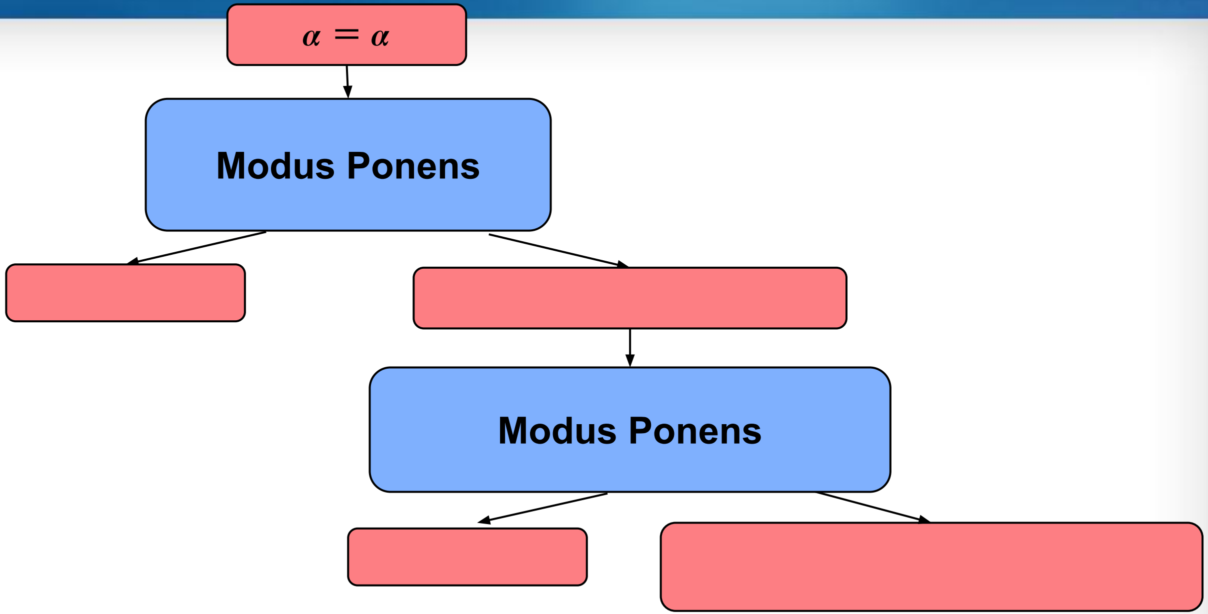
Modus Ponens(mp)

Hypotheses : $\vdash P$
$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$




```
$( Here is its proof: $)
    tt tze tpl tt weq tt tt weq tt a2 tt tze tpl
    tt weq tt tze tpl tt weq tt tt weq wim tt a2
    tt tze tpl tt tt a1 mp mp
```

Proof Tree

Axiom a1

Assertion : $\vdash t = r \Rightarrow$

$(t = s \Rightarrow r = s)$

Axiom a2

Assertion : $\vdash (t + 0) = t$

Modus Ponens(mp)

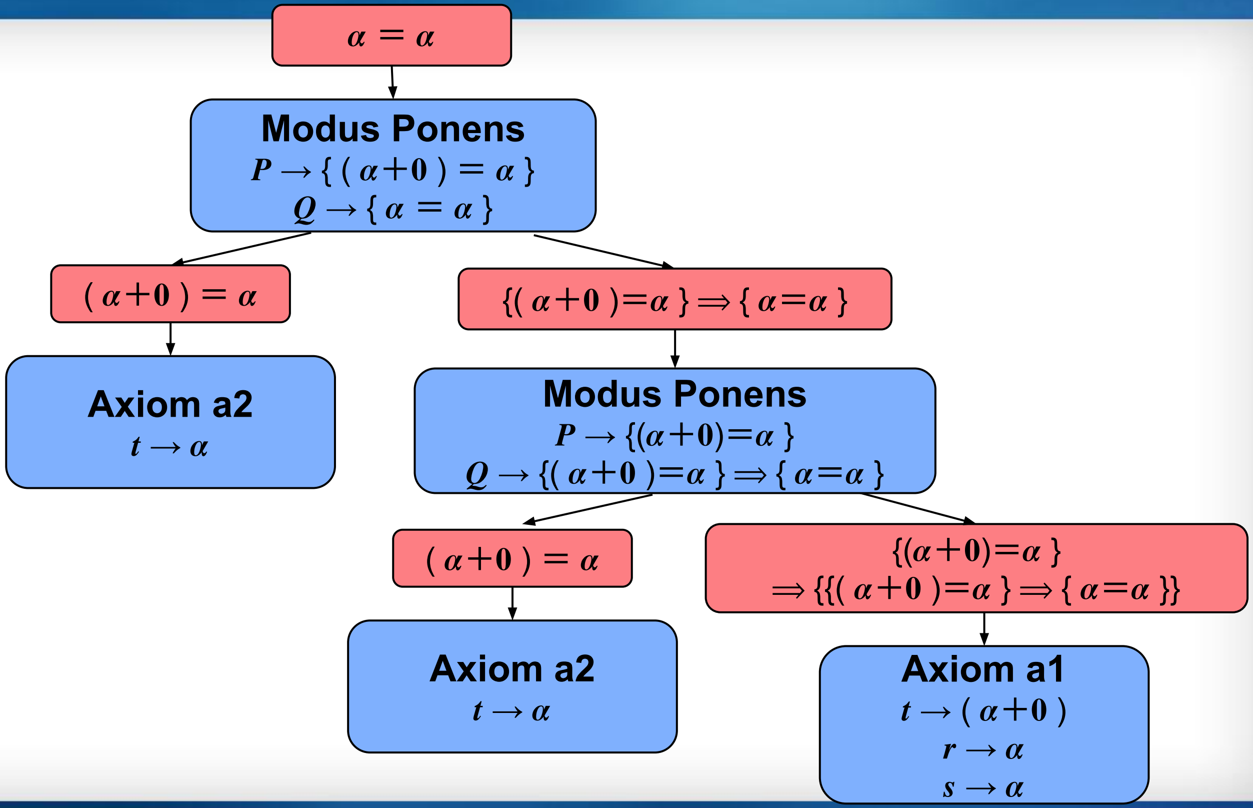
Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$



fppt.com

Proof Tree

Axiom a1

Assertion : $\vdash t=r \Rightarrow (t=s \Rightarrow r=s)$

Axiom a2

Assertion : $\vdash (t+0) = t$

Modus Ponens(mp)

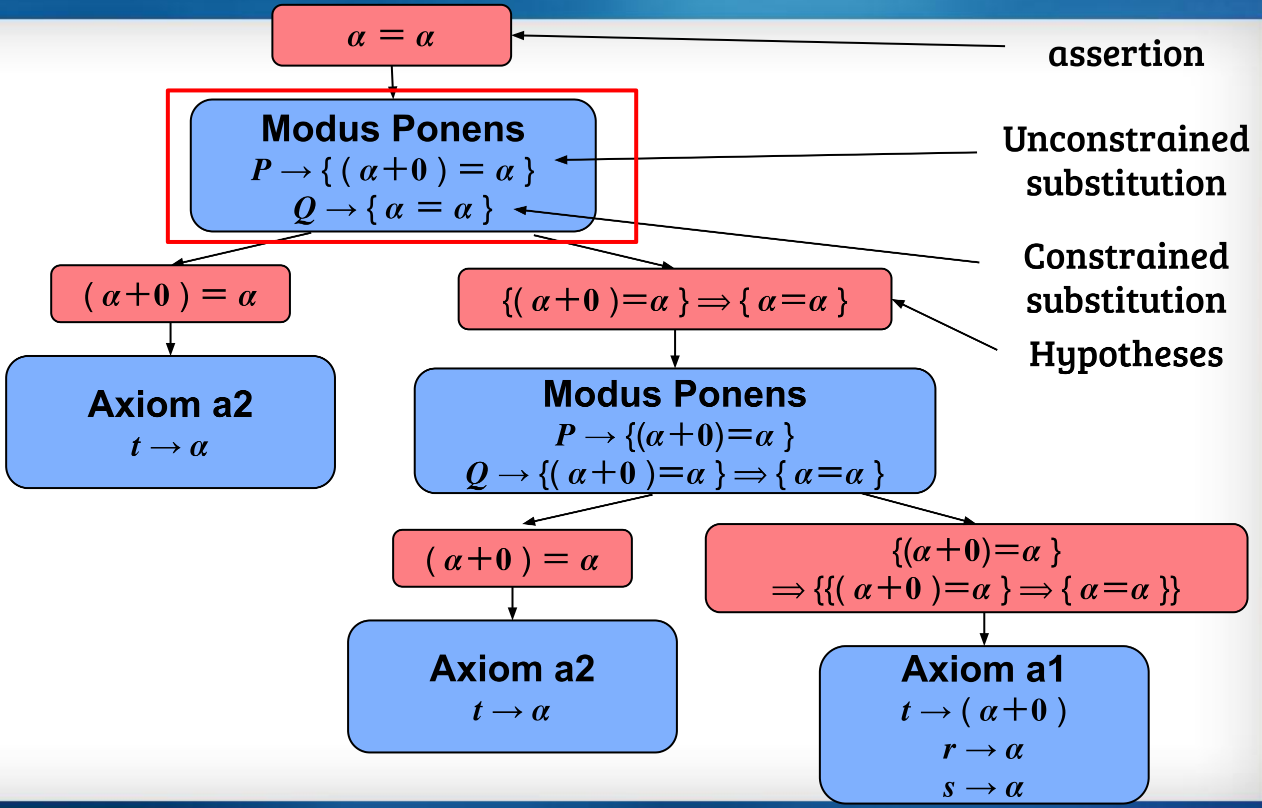
Hypotheses : $\vdash P$

$\vdash P \Rightarrow Q$

Assertion : $\vdash Q$

Theorem th1

Assertion : $\vdash \alpha = \alpha$



Proof Tree

So this theorem consists of

1. Assertion : $\alpha = \alpha$
2. Hypotheses(& axioms) : a1, a2, mp
3. Free variables : α, t, P, …
4. Pairs of disjoint variables : (t, s), (P, Q), …
   (restricting substitutions)

fppt.com

Proof Tree

Q. How to find this proof tree?

A. Of course, Deep Learning!

Holophrasm

- Given a theorem, find a proof tree using deep learning.
- 3 neural networks
  - Relevance
  - Generative
  - Payoff

fppt.com

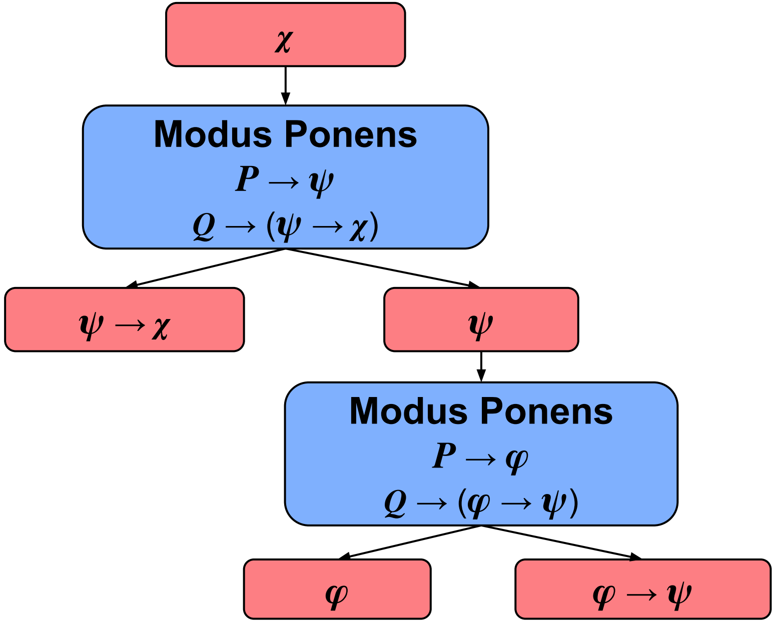
Holophrasm

Theorem mp2b

Hypotheses	Assertion
$\vdash \varphi$	$\vdash \chi$
$\vdash \varphi \rightarrow \psi$	
$\vdash \psi \rightarrow \chi$	

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \rightarrow Q$	



fppt.com

Holophrasm

Assertion

χ

Hypotheses

$\vdash \varphi$

$\vdash \varphi \to \psi$

$\vdash \psi \to \chi$

Theorem mp2b

Hypotheses	Assertion
$\vdash \varphi$	$\vdash \chi$
$\vdash \varphi \to \psi$	
$\vdash \psi \to \chi$	

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \to Q$	

fppt.com

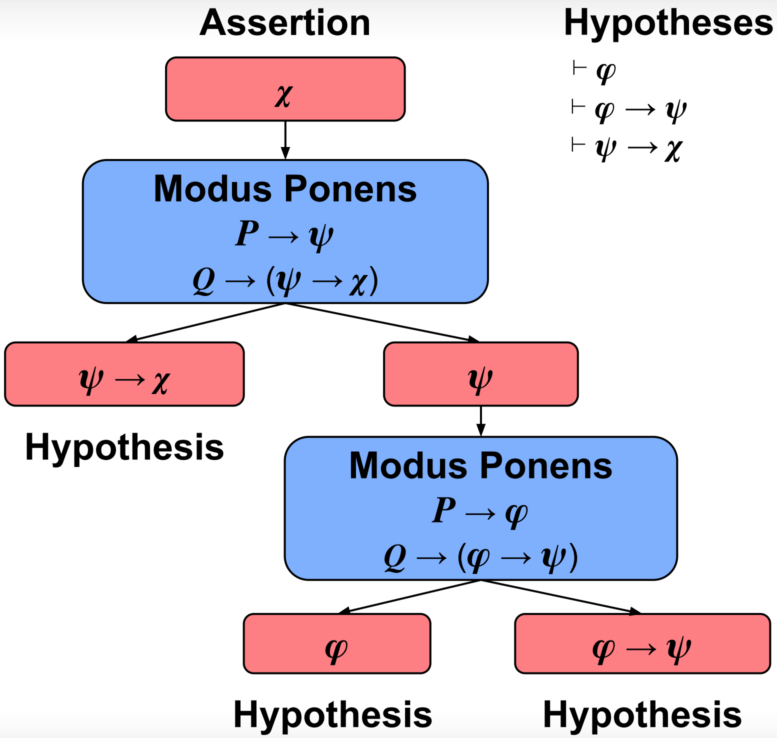
Holophrasm

Theorem mp2b

Hypotheses	Assertion
$\vdash \varphi$	$\vdash \chi$
$\vdash \varphi \to \psi$	
$\vdash \psi \to \chi$	

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \to Q$	



fppt.com

Relevance network

Assertion

$\boxed{\chi}$

Hypotheses

$\vdash \varphi$

$\vdash \varphi \rightarrow \psi$

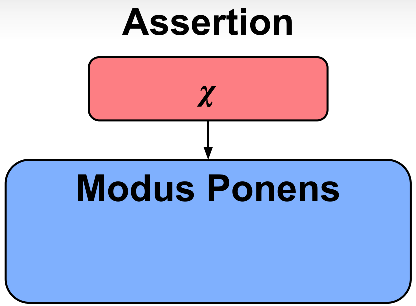
$\vdash \psi \rightarrow \chi$

- Given an assertion, find the next proposition that will be used to prove the assertion.

fppt.com

Relevance network

- Given an assertion, find the next proposition that will be used to prove the assertion.



Hypotheses

$\vdash \varphi$

$\vdash \varphi \to \psi$

$\vdash \psi \to \chi$

fppt.com

Relevance network

- Inputs
  - an assertion and a set of hypotheses of theorem we want to prove

ex)

Theorem mp2b

Hypotheses	**Assertion**
$\vdash \varphi$	$\vdash \chi$
$\vdash \varphi \to \psi$	
$\vdash \psi \to \chi$	

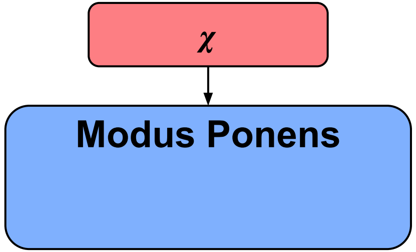
- Uses RNN
  - GRU

fppt.com

Generative network

- Given an assertion, and the proposition, find the substitutions for the unconstrained variables of the proposition.

Assertion



Hypotheses

$\vdash \varphi$
$\vdash \varphi \rightarrow \psi$
$\vdash \psi \rightarrow \chi$

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \rightarrow Q$	

fppt.com

Generative network

- Given an assertion, and the proposition, find the substitutions for the unconstrained variables of the proposition.

Assertion

χ

Modus Ponens

$P \rightarrow \psi$

$Q \rightarrow (\psi \rightarrow \chi)$

Hypotheses

$\vdash \varphi$

$\vdash \varphi \rightarrow \psi$

$\vdash \psi \rightarrow \chi$

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \rightarrow Q$	

fppt.com

Generative network

- Inputs
  - a set of hypotheses of the theorem we want to prove
  - a set of hypotheses of the proposition

ex)

Hypotheses of mp2b	**Hypotheses of modus Ponens**
$\vdash \varphi$	$\vdash P$
$\vdash \varphi \rightarrow \psi$	$\vdash P \rightarrow Q$
$\vdash \psi \rightarrow \chi$	

- Uses sequence to sequence model
  - GRU

Payoff network

- repeat the previous procedures on the remaining assertions to find a proof tree.

Assertion

χ

↓

Modus Ponens

$P \to \psi$

$Q \to (\psi \to \chi)$

Hypotheses

$\vdash \varphi$

$\vdash \varphi \to \psi$

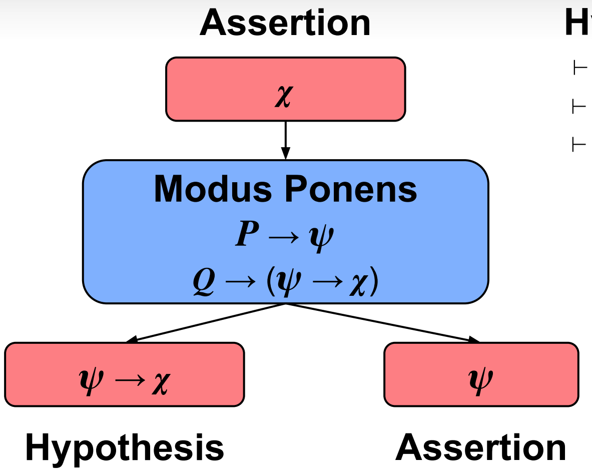
$\vdash \psi \to \chi$

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \to Q$	

Payoff network

- repeat the previous procedures on the remaining assertions to find a proof tree.



Hypotheses

$\vdash \varphi$

$\vdash \varphi \to \psi$

$\vdash \psi \to \chi$

Axiom Modus Ponens

Hypotheses	Assertion
$\vdash P$	$\vdash Q$
$\vdash P \to Q$	

fppt.com

Payoff network

- repeat the previous procedures on the remaining assertions to find a proof tree.

Axiom Modus Ponens

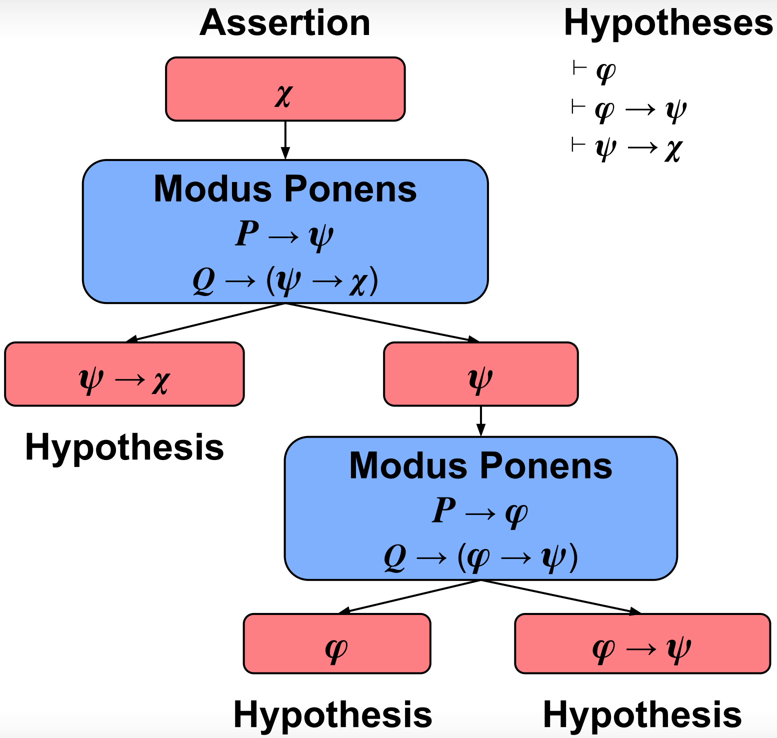
Hypotheses

$\vdash P$

$\vdash P \to Q$

Assertion

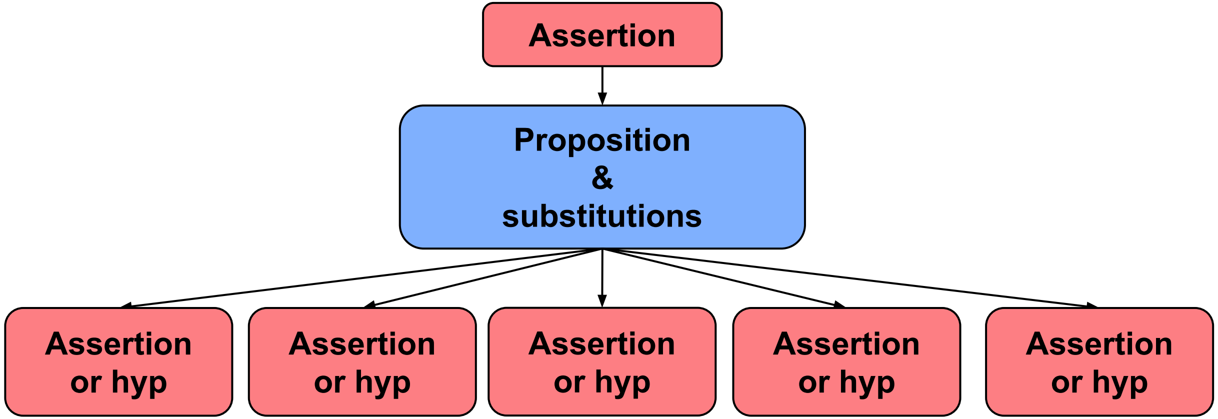
$\vdash Q$



fppt.com

Payoff network

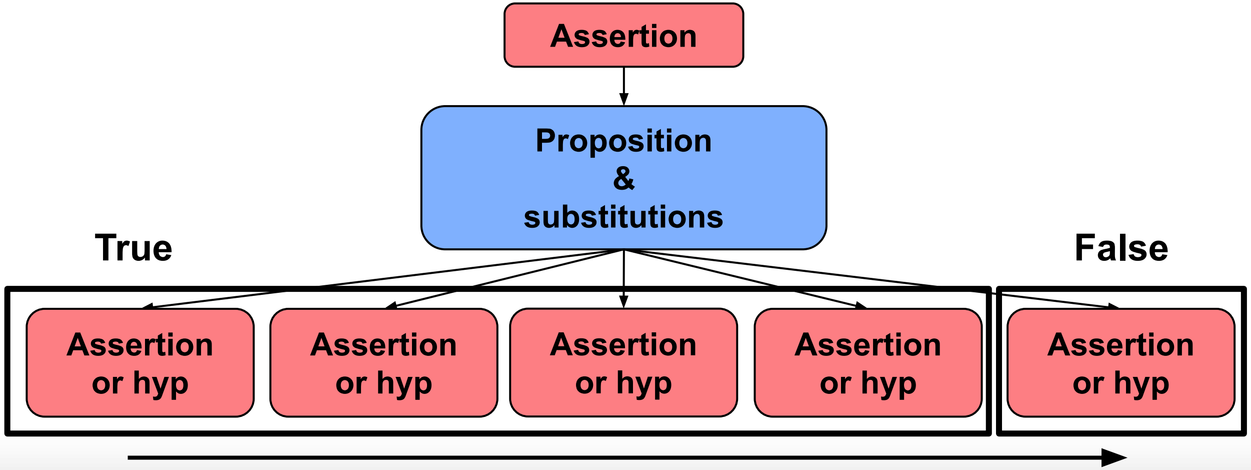
- What if there are too many assertions?



fppt.com

Payoff network

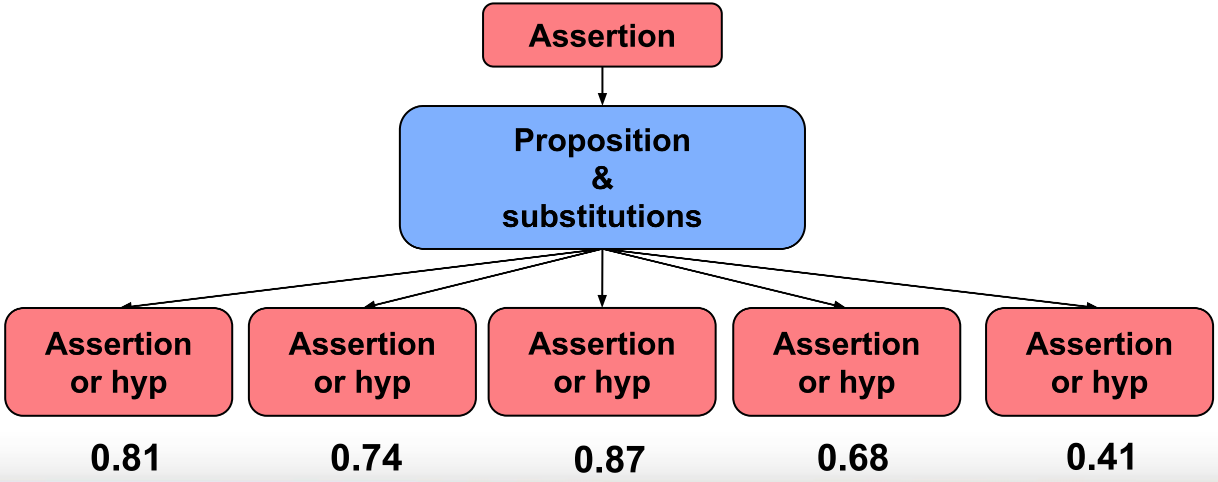
- What if there are too many assertions?



fppt.com

Payoff network

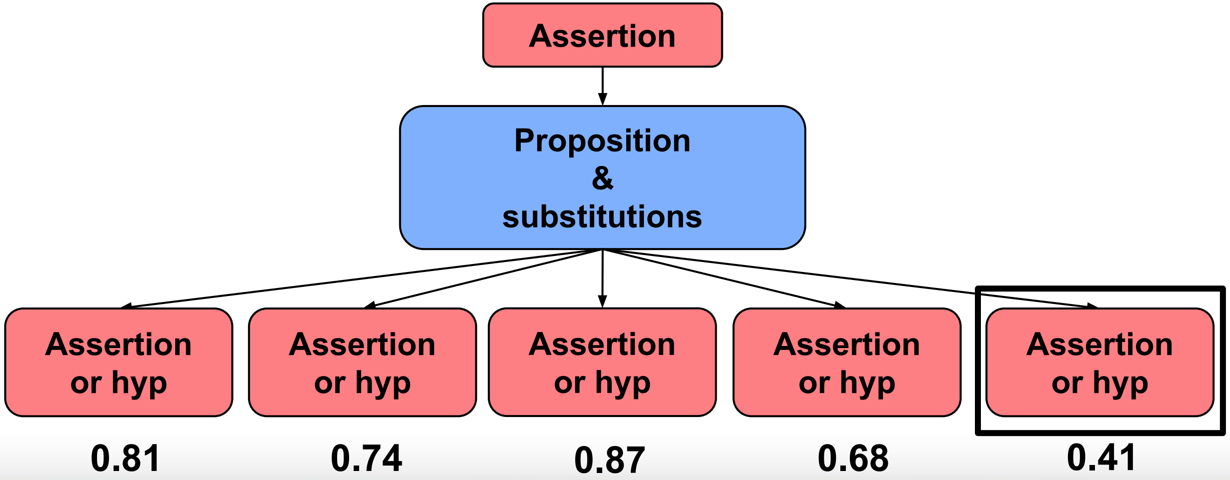
- Given a set of assertions, find the least promising assertion (least probable to be proved)



fppt.com

Payoff network

- Given a set of assertions, find the least promising assertion (least probable to be proved)



fppt.com

Payoff network

- Input: an assertion and a set of hypotheses of theorem
- Uses RNN
  - GRU
  - Bidirectional network

fppt.com

Outline

- Introduction
- Metamath: The Foundation
- Holophrasm: An Overview
- Future Work

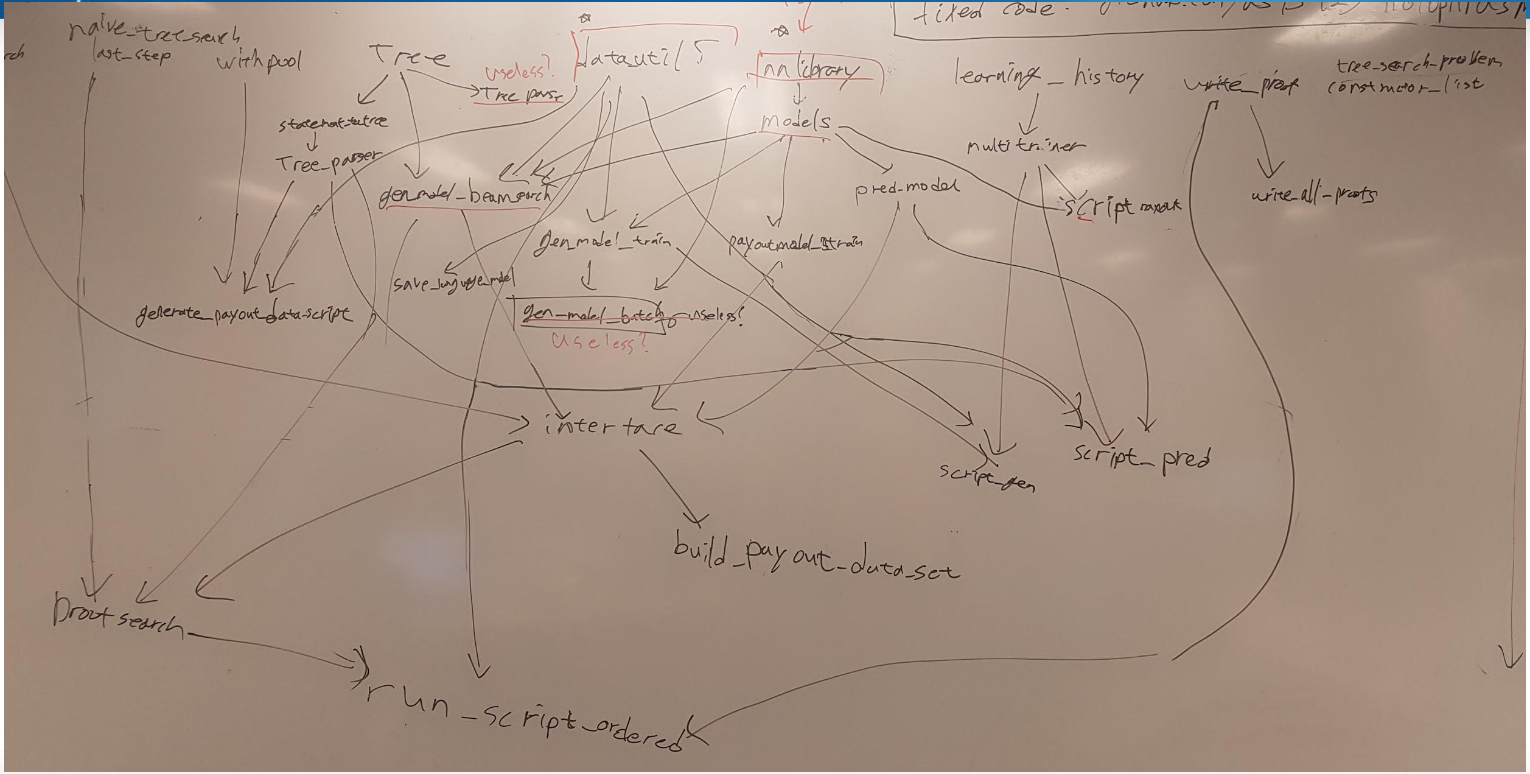
fpt.com

Future Work

- Reconstruct Holophrasm and make descriptions

fppt.com

Future Work


fixed code
naive_tree_search
last_step
with pool
Tree
useless?
Tree_parse
datautils
nn library
models
learning_history
write_proof
tree-search-problem
constructor_list
Tree_parser
gen_model_beamsearch
gen_model_train
pred-model
multi trainer
write_all_proofs
save_language_model
generate_payout_data_script
gen_model_batch
useless?
Useless?
interface
script_gen
script_pred
build_payout_data_set
run_script_ordered

Future Work

Accelerating Deep Learning Research with the Tensor2Tensor Library

Monday, June 19, 2017

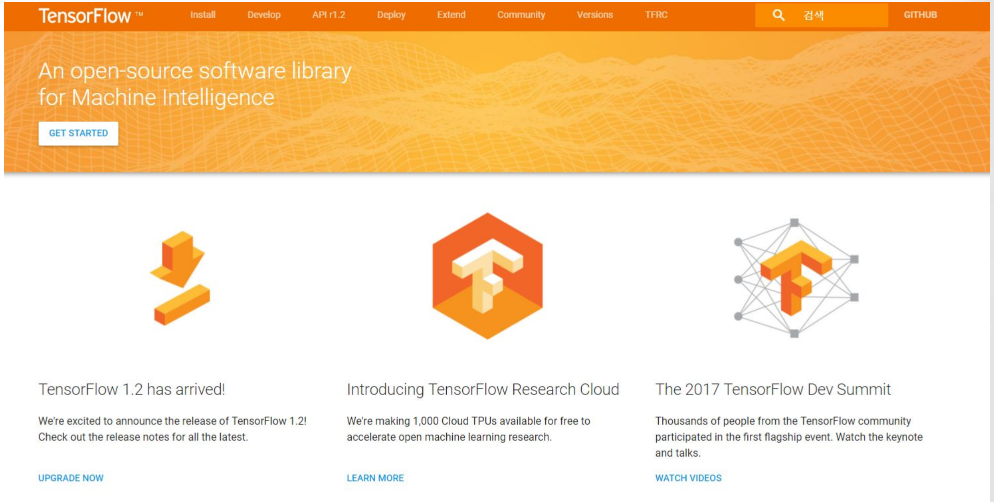
Posted by Łukasz Kaiser, Senior Research Scientist, Google Brain Team

Deep Learning (DL) has enabled the rapid advancement of many useful technologies, such as machine translation, speech recognition and object detection. In the research community, one can find code open-sourced by the authors to help in replicating their results and further advancing deep learning. However, most of these DL systems use unique setups that require significant engineering effort and may only work for a specific problem or architecture, making it hard to run new experiments and compare the results.

Today, we are happy to release Tensor2Tensor (T2T), an open-source system for training deep learning models in TensorFlow. T2T facilitates the creation of state-of-the art models for a wide variety of ML applications, such as translation, parsing, image captioning and more, enabling the exploration of various ideas much faster than previously possible. This release also includes a library of datasets and models, including the best models from a few recent papers (*Attention Is All You Need*, *Depthwise Separable Convolutions for Neural Machine Translation* and *One Model to Learn Them All*) to help kick-start your own DL research.

Translation Model	Training time	BLEU (difference from baseline)
Transformer (T2T)	3 days on 8 GPU	**28.4 (+7.8)**
SliceNet (T2T)	6 days on 32 GPUs	26.1 (+5.5)
GNMT + Mixture of Experts	1 day on 64 GPUs	26.0 (+5.4)
ConvS2S	18 days on 1 GPU	25.1 (+4.5)
GNMT	1 day on 96 GPUs	24.6 (+4.0)
ByteNet	8 days on 32 GPUs	23.8 (+3.2)
MOSES (phrase-based baseline)	N/A	20.6 (+0.0)

BLEU scores (higher is better) on the standard WMT English-German translation task.



fppt.com

Reference

Norman Megill. "Metamath: A Computer Language for Pure Mathematics" (PDF). Lulu Press, Morrisville, North Carolina. ISBN 978-1-4116-3724-5. p. xi

Daniel P.Z. Whalen(2016). "Holophrasm: a neural Automated Theorem Prover for higher-order logic". : arxiv:1608.02644

Norman Megill. "Demo0.mm" Jan 1, 2004, Retrieved July 13, 2017

"Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano – WildML". *Wildml.com*. Retrieved July 13, 2017.
"Accelerating Deep Learning Research with the Tensor2Tensor Library". *Research.googleblog.com*. Retrieved July 14, 2017.

"TensorFlow", *TensorFlow.org.* Retrieved July 14, 2017

fppt.com

Thank you for listening!

fppt.com

Thank you for listening!

Any questions?

fppt.com