

# Combining Axiom Injection and Knowledge Base Completion for Efficient Natural Language Inference

Masashi Yoshikawa<sup>◆</sup>, Koji Mineshima<sup>★</sup>, Hiroshi Noji<sup>♥</sup>, Daisuke Bekki<sup>★</sup>

◆ Nara Institute of Science and Technology

★ Ochanomizu University

♥ Artificial Intelligence Research Center, AIST

AAAI-33 2019/1/31



# Recognizing Textual Entailment

a.k.a. Natural Language Inference

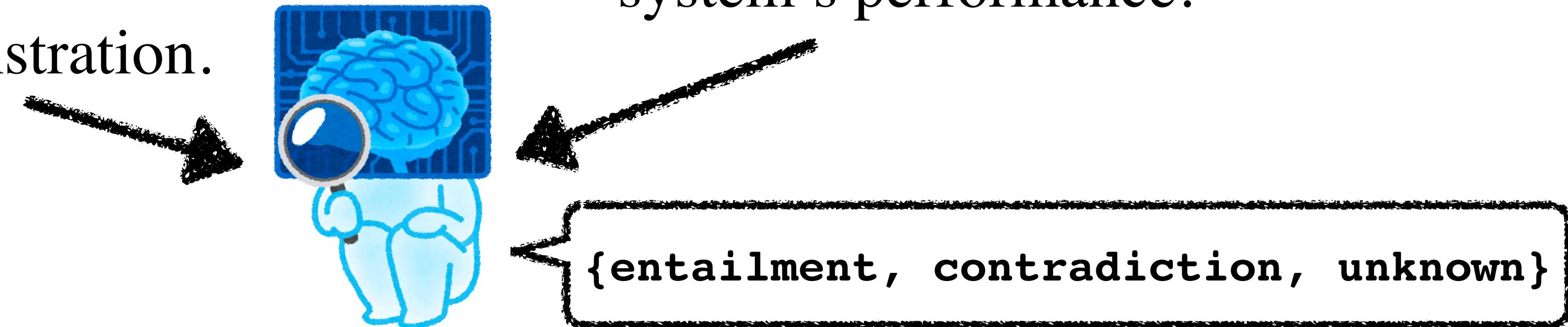
## Premise(s)

P1: Clients at the demonstration were all impressed by the system's performance.

P2: Smith was a client at the demonstration.

## Hypothesis

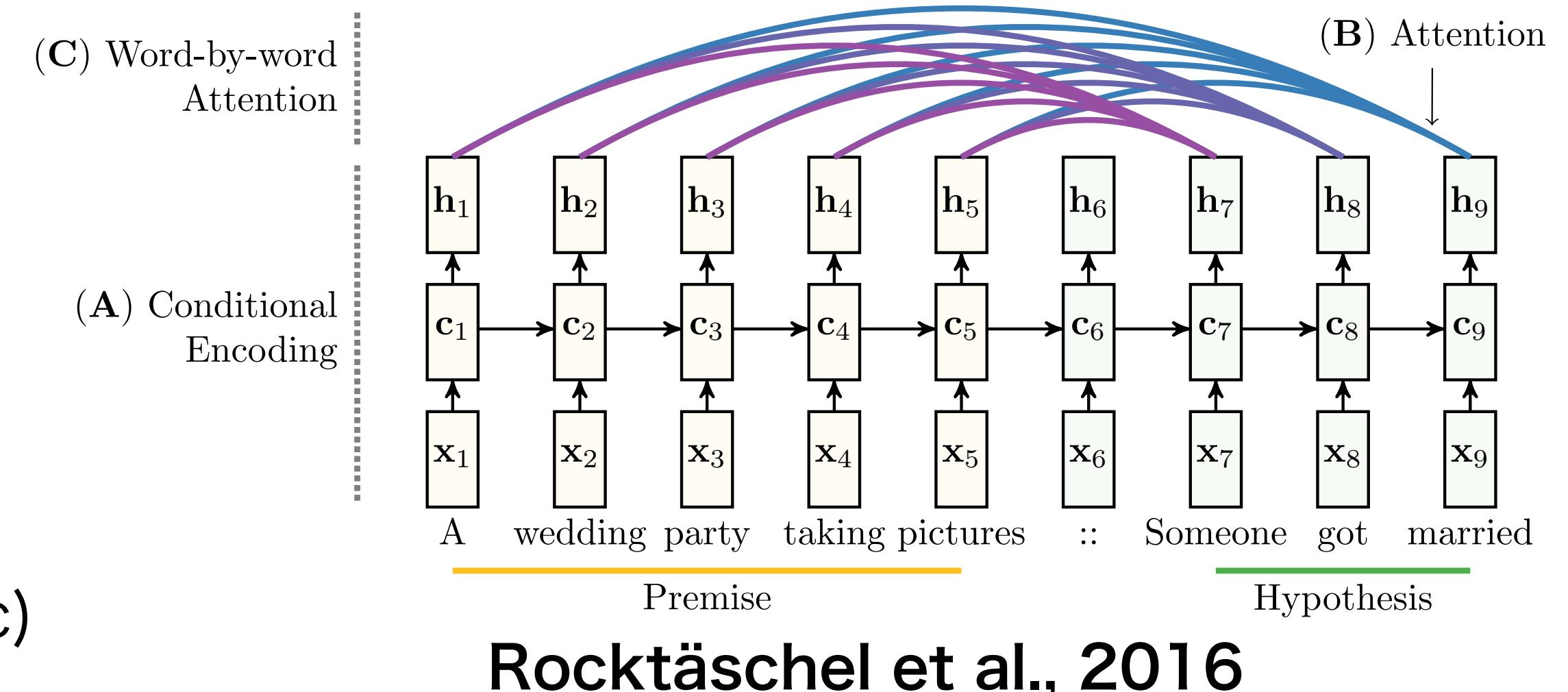
H: Smith was impressed by the system's performance.



- A testbed to evaluate if a machine can reason as we do
  - lexical, logical, syntactic phenomena, etc.
- Elemental technology for improving other NLP tasks
  - Question answering, reading comprehension, etc.

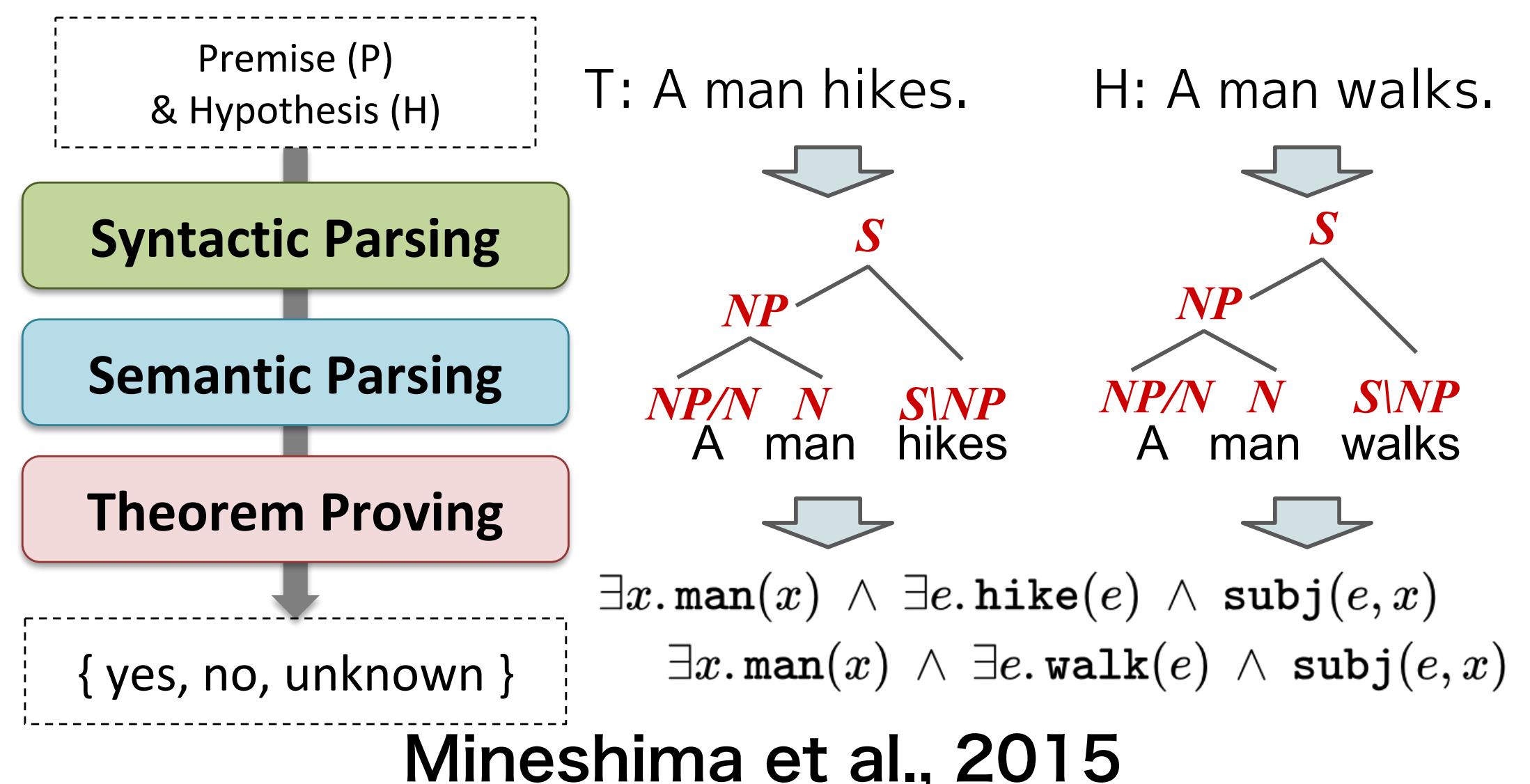
# Approaches to RTE

- Machine learning (Rocktäschel et al., 2016, etc.)
  - e.g. Neural Networks
- Logic (Mineshima et al., 2015, Abzianidze 2017, etc)



- Traditional pipeline systems
- Theorem prover (e.g. Coq)  

- Ours: logic-based, extended by ML! (Hybrid)



# ccg2lambda (Mineshima et al., 2015)

Premise (P)  
& Hypothesis (H)

## Syntactic Parsing

CCG Derivations

## Semantic Parsing

Logical Formulas

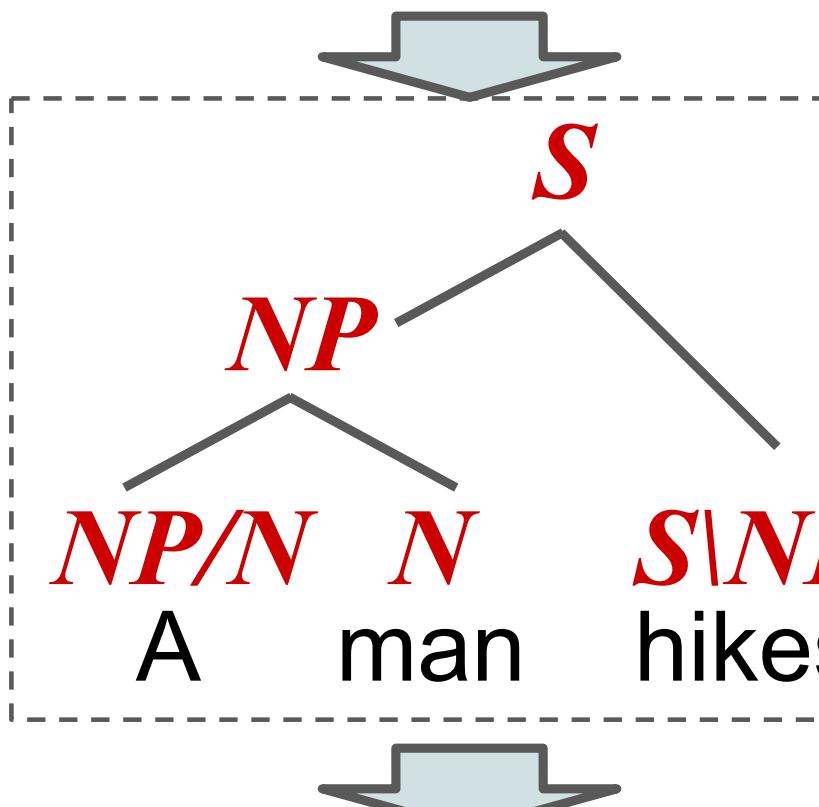
## Theorem Proving

{ yes, no, unknown }

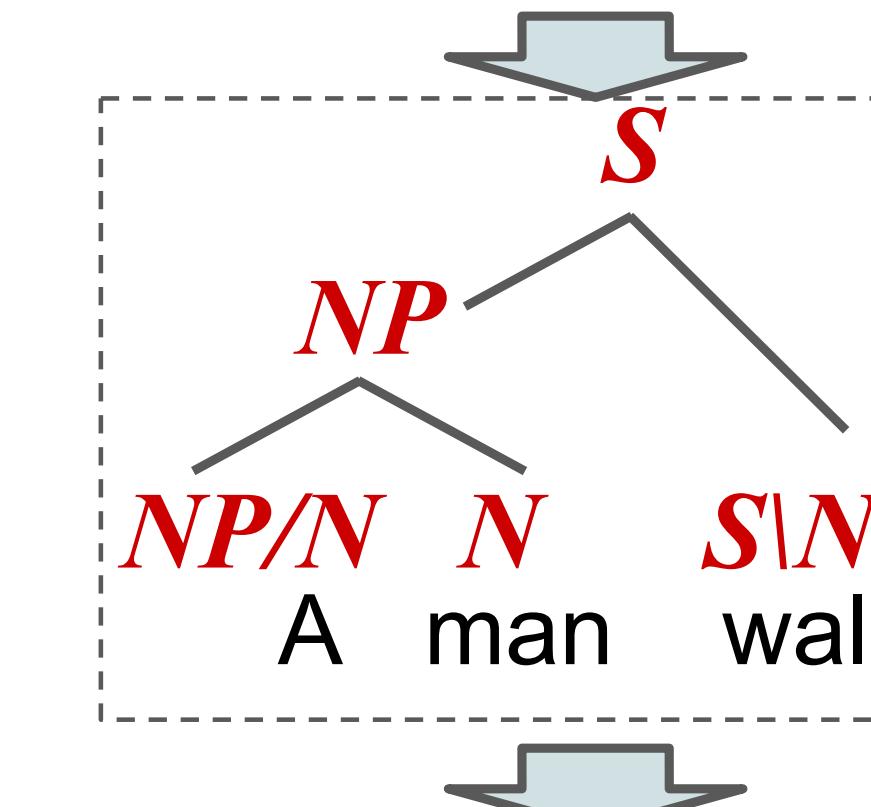
👍 Unsupervised

👍 Captures linguistic phenomena  
- 83.6 % accuracy in SICK

P: A man hikes.



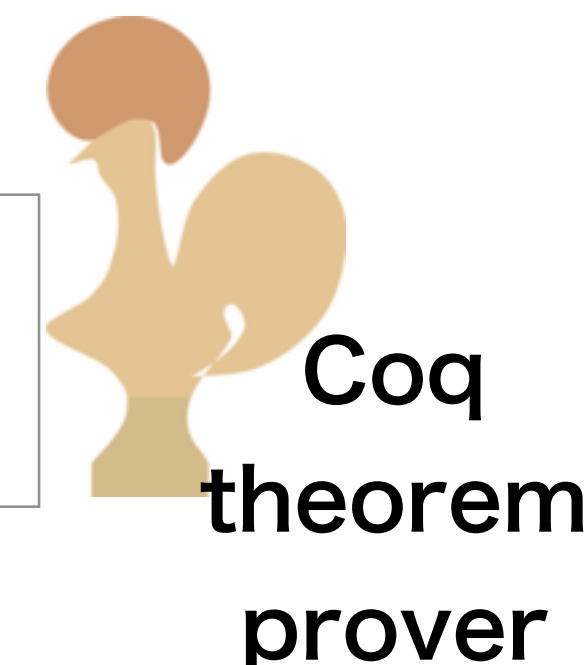
H: A man walks.



$\exists x. \text{man}(x) \wedge \exists e. \text{hike}(e) \wedge \text{subj}(e, x)$

$\exists x. \text{man}(x) \wedge \exists e. \text{walk}(e) \wedge \text{subj}(e, x)$

```
Coq < Theorem t1:  
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->  
exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).  
Coq < Proof. ccg2lambda. Qed.
```



result: unknown

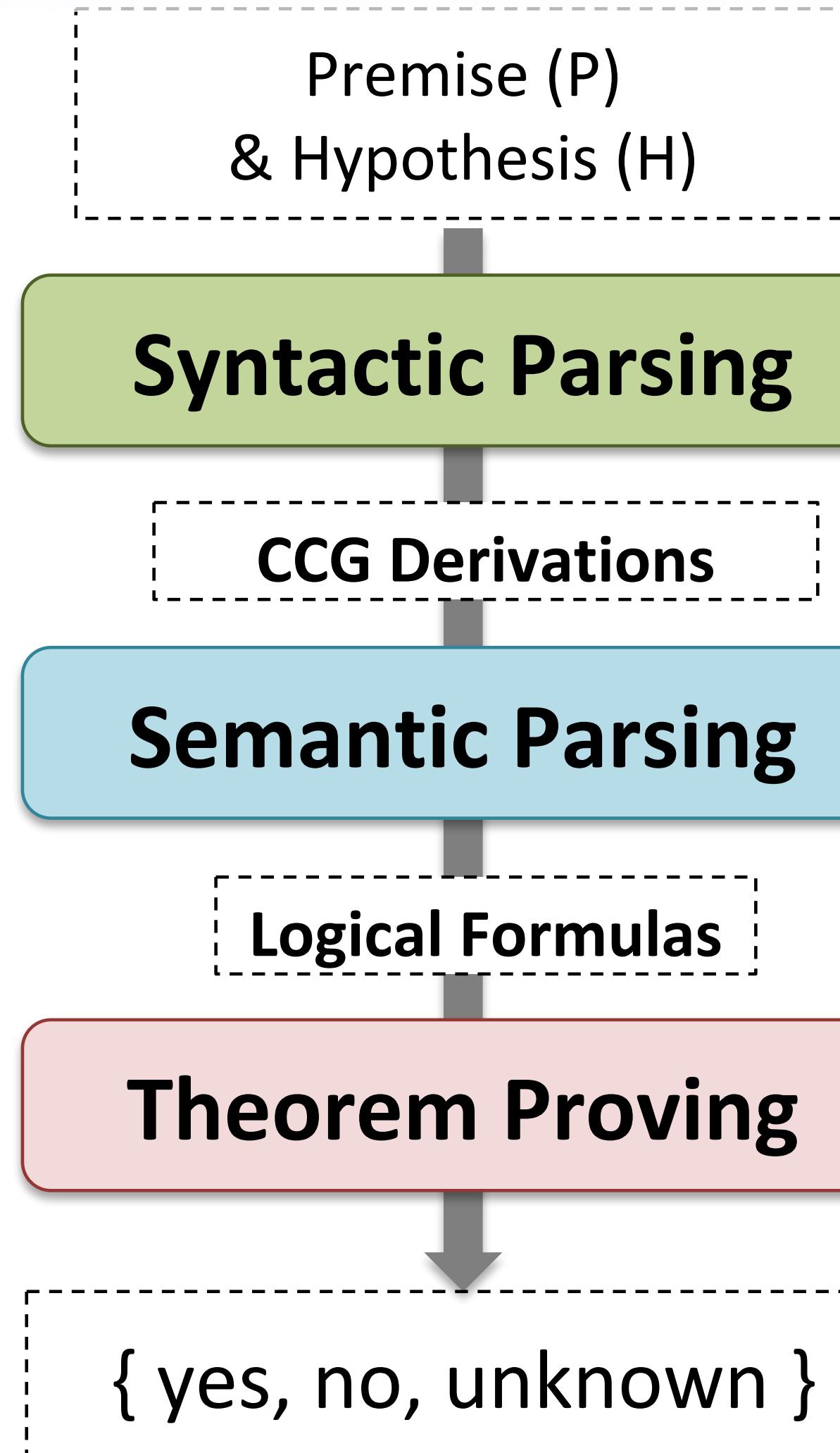


How to handle external knowledge?

e.g.  $\forall x. \text{hike}(x) \rightarrow \text{walk}(x)$

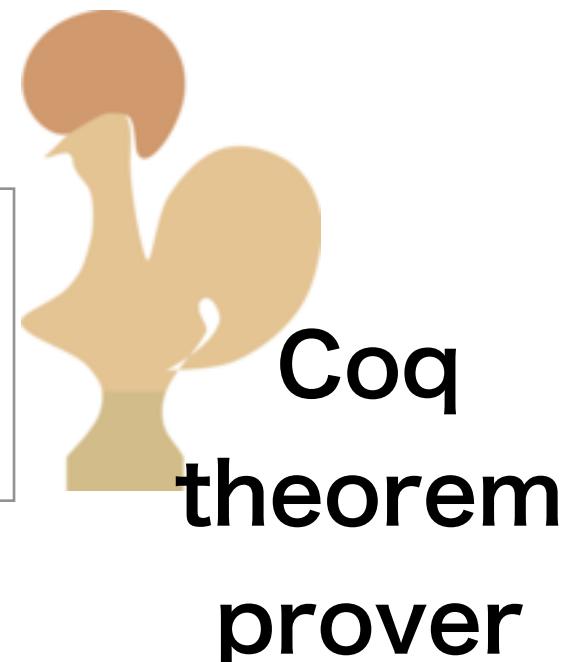
- Use WordNet as axioms blows up  
the search space of theorem proving!

# "Abduction" Mechanism (Martínez-Gómez et al., 2017)



```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

result: **unknown**



More steps when the 1st theorem proving is unsuccessful

1. Search KBs (e.g. WordNet) for useful lexical relations
2. Rerun Coq with additional axioms

# "Abduction" Mechanism (Martínez-Gómez et al., 2017)

- Promising approach to handling external knowledge within a logic-based system

- (However,) **Practical issues:**



- We want to **add more knowledge** to increase the coverage of reasoning
- We want the **KBs to be compact** for efficient inference & memory usage
- Do not want to run Coq again and again for real applications 😞
- Ideally, the mechanism should be tightly integrated with the inference for efficiency

👉 We solve these issues by:

1. Replacing search on KBs by techniques of "Knowledge Base Completion"
2. Developing "**abduction**" Coq plugin



# 1. Extending Abduction Mechanism with KBC

- Knowledge Base Completion:

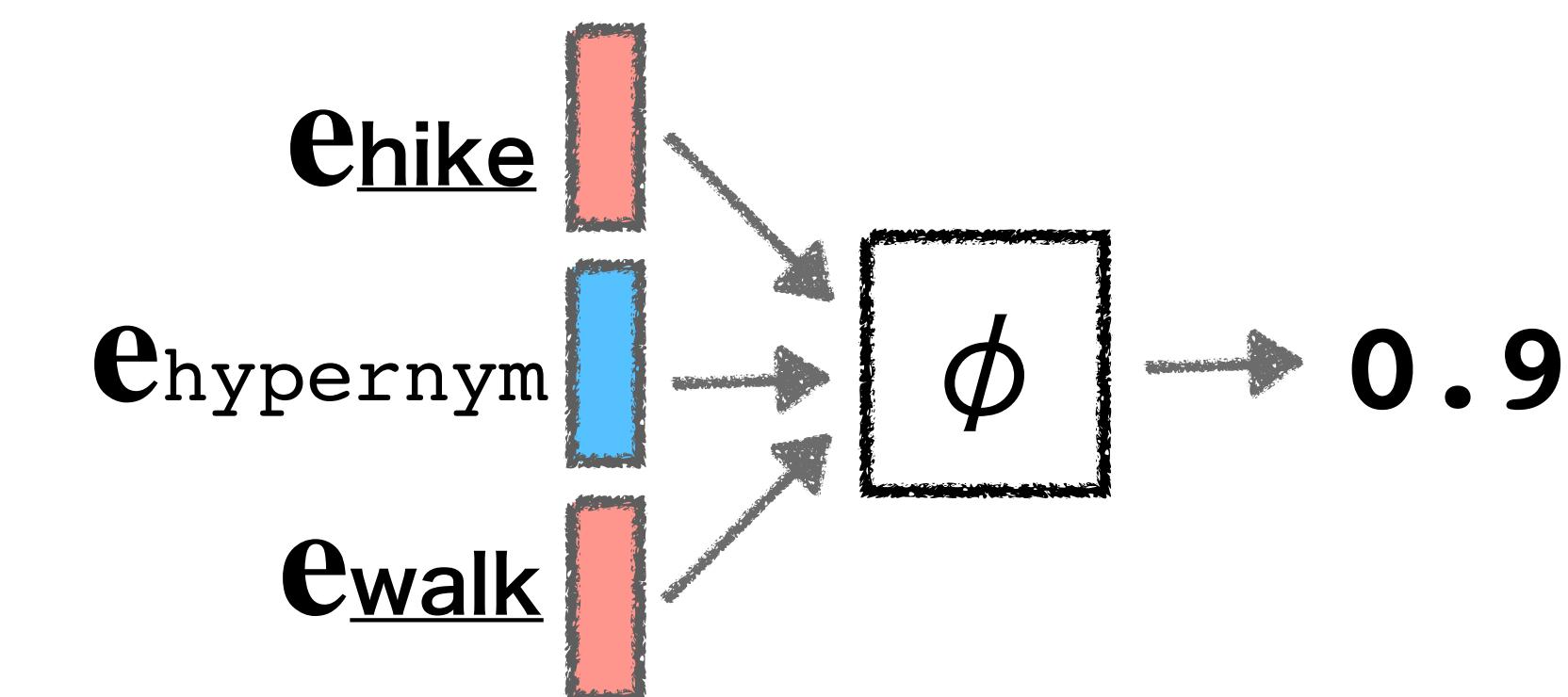
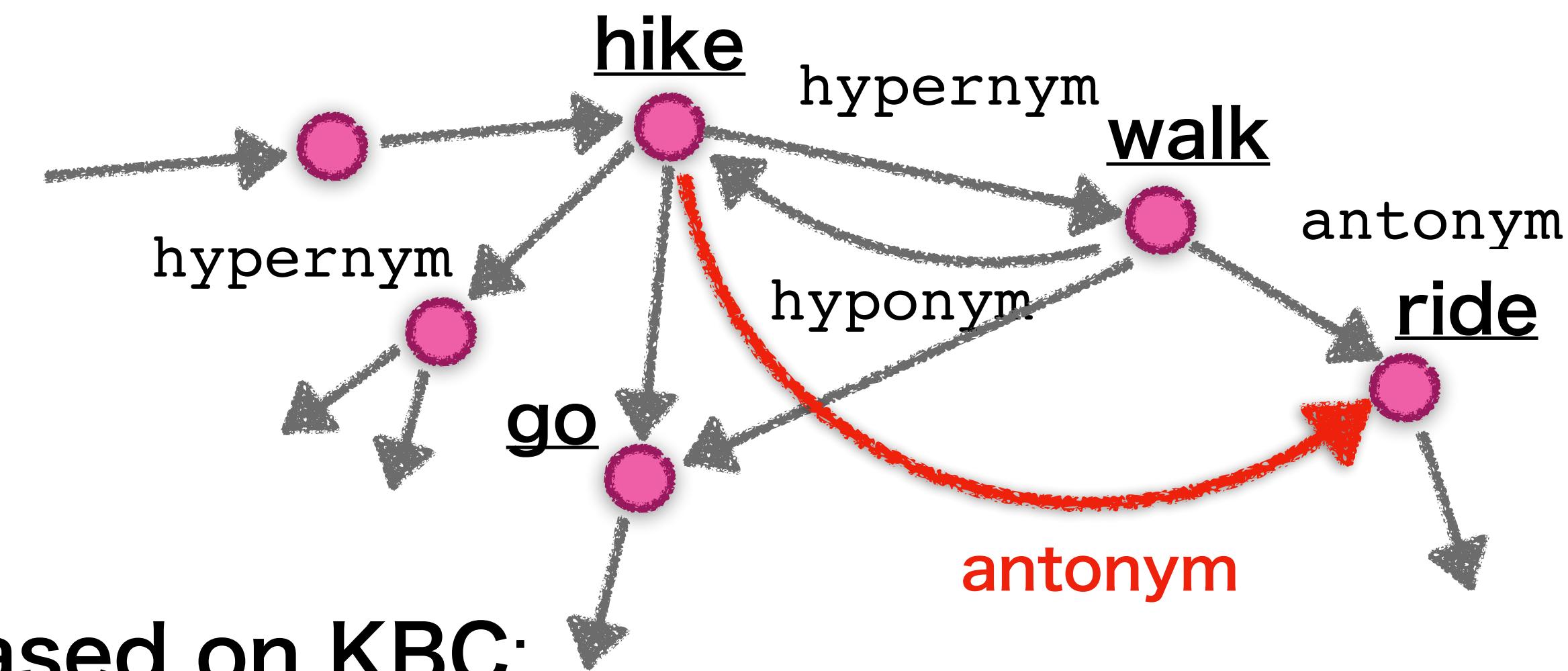
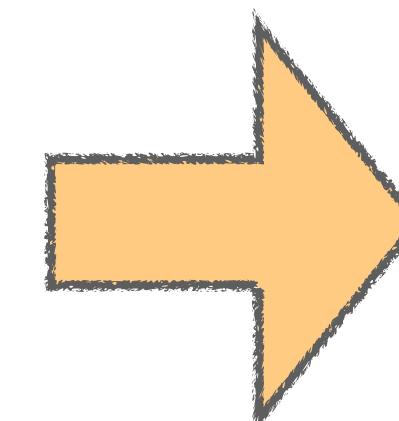
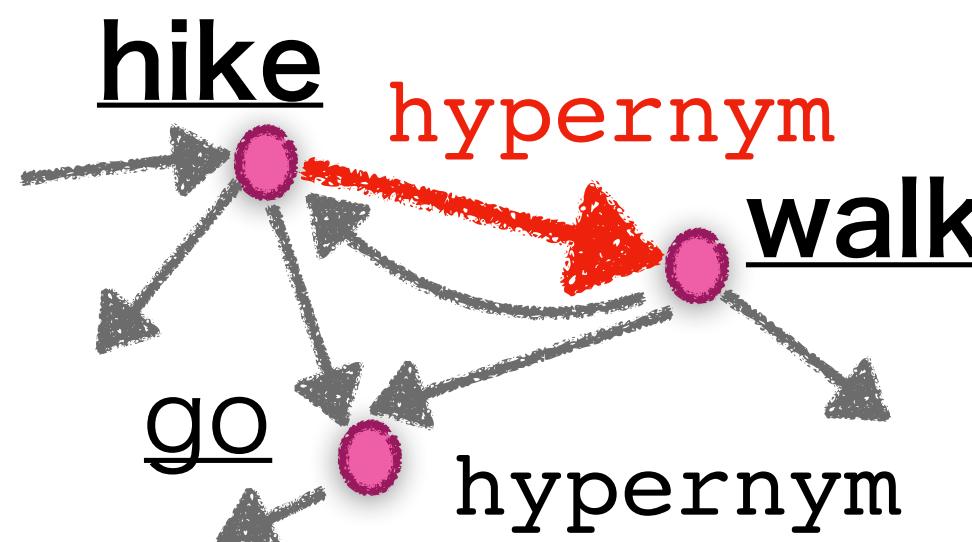
- A task to complement missing relations

- recent huge advancement

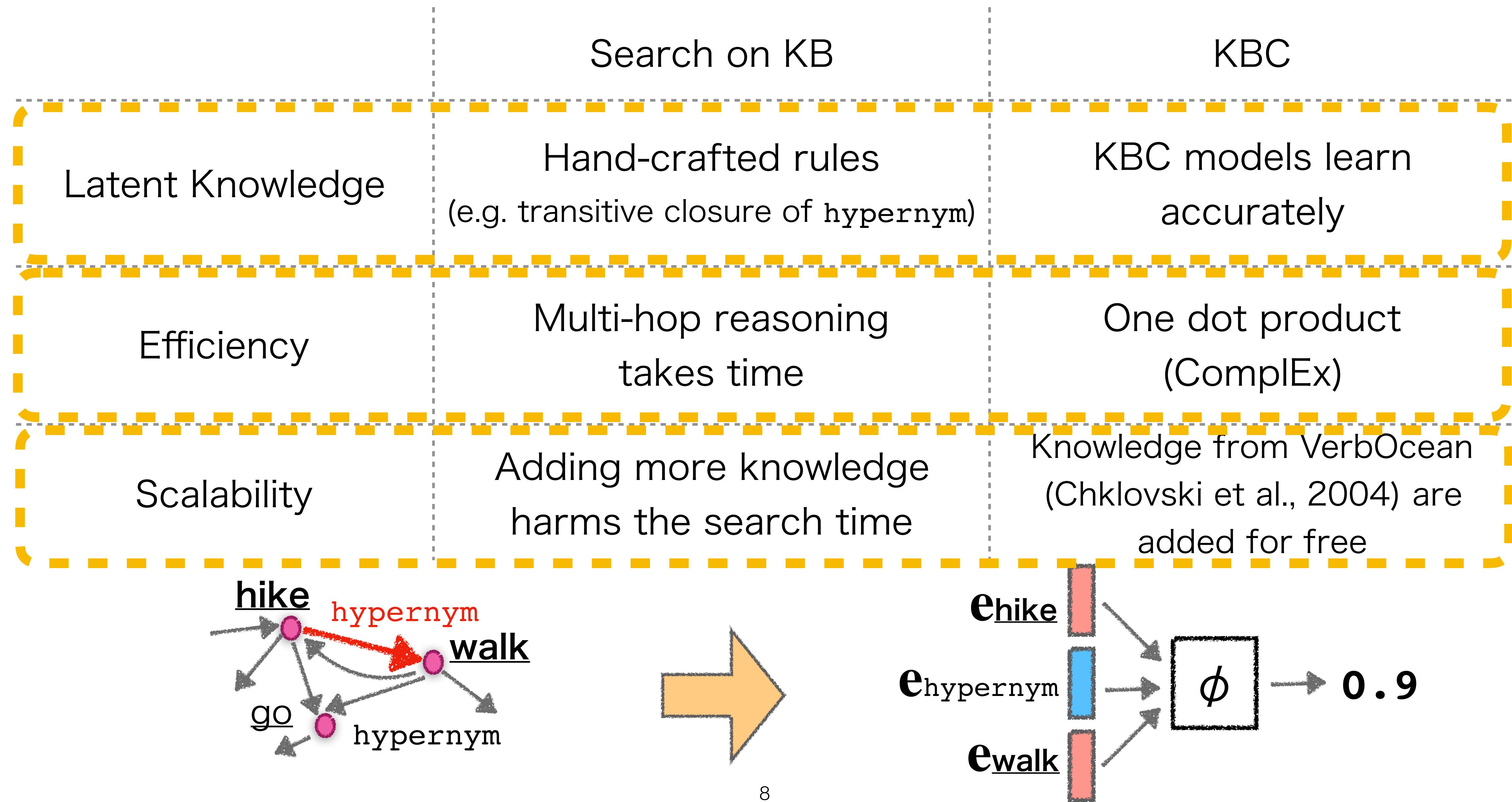
- We propose an **abduction mechanism based on KBC**:

- If  $(s, r, o)$  is missing, use it as axiom if  $\phi(s, r, o) \geq \delta$  (threshold)

- ComplEx (Trouillon et al., 2016):  $\phi(s, r, o) = \sigma(Re(\langle \mathbf{e}_s, \mathbf{e}_r, \mathbf{e}_o \rangle))$ ,  $\forall \mathbf{e}_v \in \mathbb{C}^n$

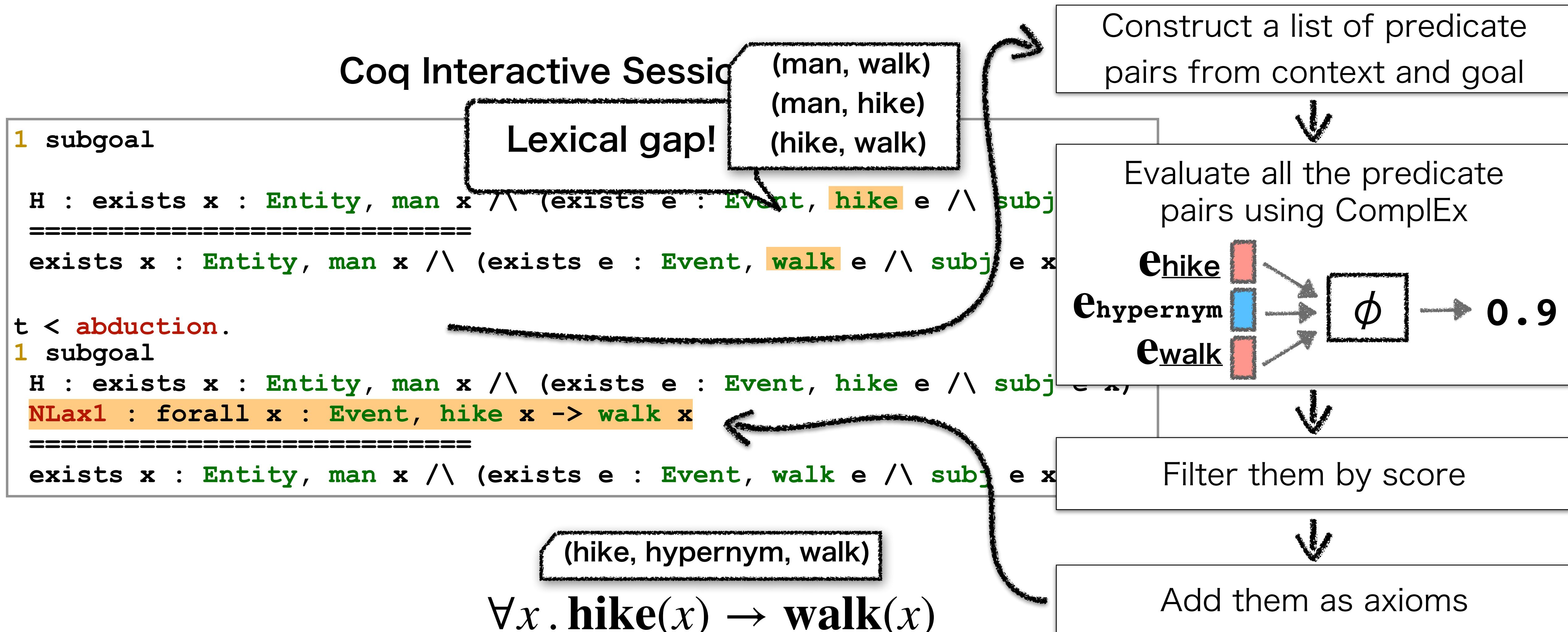


# 1. Extending Abduction Mechanism with KBC





## 2. Faster Reasoning with "abduction" Coq plugin



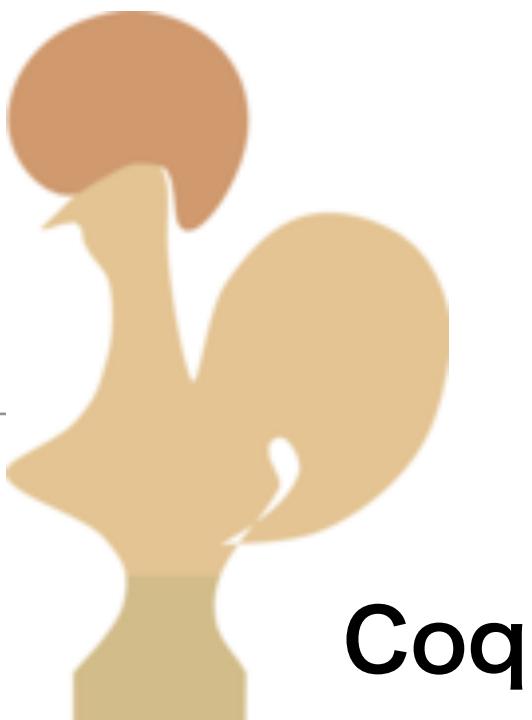
**Semantic Parsing****Logical Formulas****Theorem Proving**

{ yes, no, unknown }

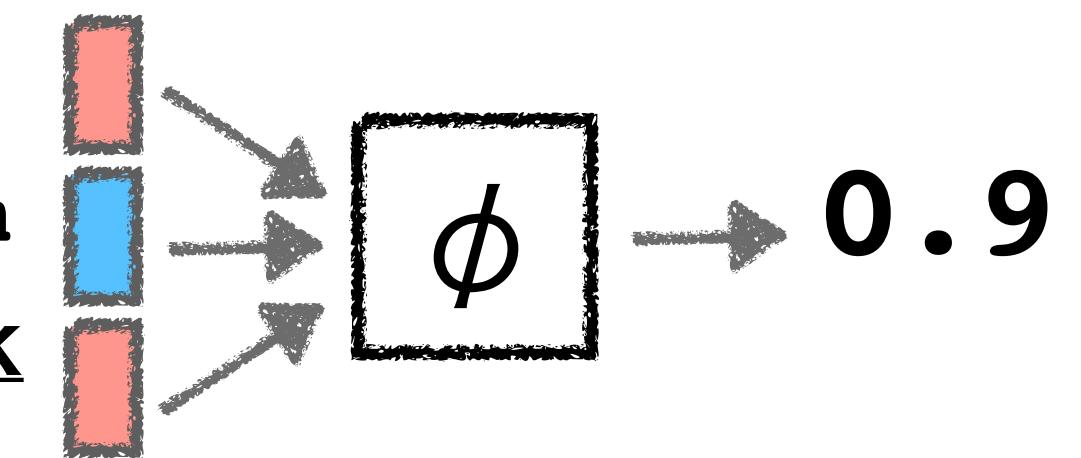
# Summary so far...

$$\begin{array}{c} \text{A man hikes} \\ \downarrow \\ \exists x. \text{man}(x) \wedge \exists e. \text{hike}(e) \wedge \text{subj}(e, x) \\ \text{A man walks} \\ \downarrow \\ \exists x. \text{man}(x) \wedge \exists e. \text{walk}(e) \wedge \text{subj}(e, x) \end{array}$$

```
Coq < Theorem t1:
(exists x : Entity, man x /\ (exists e : Event, hike e /\ subj e x)) ->
exists x : Entity, man x /\ (exists e : Event, walk e /\ subj e x).
Coq < Proof. ccg2lambda. Qed.
```

**+abduction****result: yes**

$e_{\text{hike}}$   
 $e_{\text{hypernym}}$   
 $e_{\text{walk}}$



- 👍 Efficient and scalable abduction mechanism
- 👍 No need to rerun Coq in abduction
  - Our method is applicable to other logic-based systems
    - e.g. Modern Type Theory (Bernandy and Chatzikyriakidis, 2017)

# Experiments

- SICK RTE dataset (Marelli et al., 2014)
- Metrics: accuracy and processing time
- ComplEx is trained on logistic loss:  $\sum_{((s,r,o),t) \in \mathcal{D}} t \log f(s, r, o) + (1 - t) \log(1 - f(s, r, o))$
- The training data is constructed using WordNet
  - synonym, antonym, hyponym, hypernyms, etc.
  - The trained ComplEx model achieves MRR of 77.68%

P: A flute is being played in a lovely way by a girl.

H: One woman is playing a flute.

syntactic

logical

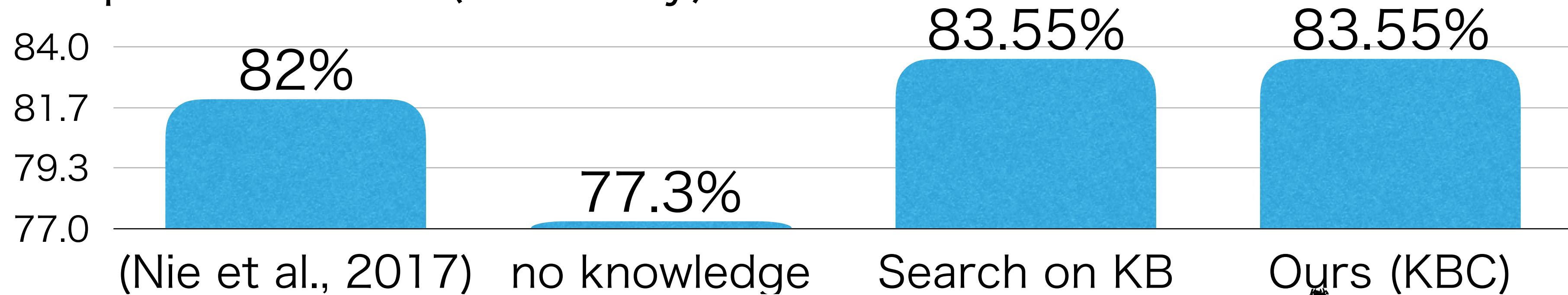
lexical  
phenomena



entailment

# Experimental Results on SICK

- RTE performance (accuracy)



Achieves the same accuracy, improving significantly from "no knowledge" case

- Processing speed (second per a problem)



- Baselines: Search on KB (Martínez-Gómez et al., 2017), NN

Our method halves the time to process an RTE problem!

# Thank you!

- A KBC-based axiom injection for logic-based RTE systems
  - Efficient, scalable, and it provides latent knowledge
- **abduction** tactic for further faster reasoning
- **Come to my poster (#1319) for other topics:**
  - Adding other KB (VerbOcean) without losing efficiency
  - Evaluating learned latent knowledge in terms of RTE (LexSICK dataset)
- **All the codes, dataset and slides are available:**
  - <https://masashi-y.github.io>