Learning Language in Logic

⚠ Personally biased account ⚠

Michael Kohlhase

School of Computer Science	Fachbereich Informatik
Carnegie Mellon University	Universität des Saarlandes
Pittsburgh, USA	Saarbrücken, Germany

http://www.cs.cmu.edu/~kohlhase

NSH 4623, 8-5749, come see me!

Learning Language in Logic

- is a developing research area at the intersection of
  - computational logic (my area),
  - natural language processing (my hobby)
  - machine learning (related to language acquisition?).
- Overview of the talk:
  - What is (computational) logic (basic introduction)
  - logic and natural language semantics (Where are we, and why?)
  - inductive logic programming (learning with logical methods)
  - some applications (if we can get to it)

What is Logic?

- Logic studies formal languages, inference and their relation with the world
  - Formal language: set of formulae ($2+3/7$, $\forall x.x+y=y+x$, $A\vee\neg A$)
  - Formula: sequence/tree of symbols ($x, y, f, g, p, 1, \pi, \in, \neg, \wedge, \forall, \exists$)
  - Inference: rules to transform (sets of) formulae ($A, A\Rightarrow B\vdash B$)
  - Models: things we understand (e.g. number theory)
  - Interpretation: maps formulae into models ($[\![\texttt{three plus five}]\!]=8$)
  - Validity: $\mathcal{M}\models A$, iff $[\![A]\!]_{\mathcal{M}}=\mathtt{T}$ ($\texttt{five greater three}$ is valid)
  - Entailment: $A\models B$, iff $\mathcal{M}\models B$ for all $\mathcal{M}\models A$.
- Syntax: formulae, inference (just a bunch of symbols)
- Semantics: models, interpr., validity, entailment (math. structures)
- Important Question: What is the relation between syntax and semantics?

A simple System: Prop. Logic with Hilbert-Calculus

- Formulae: build up from prop. variables: $P, Q, R, \ldots$ and implication: $\Rightarrow$
- Semantics: $[\![P]\!]_\varphi = \varphi(P)$ given, $[\![\mathbf{A} \Rightarrow \mathbf{B} = \mathtt{T}$, iff $[\![\mathbf{A}]\!] = \mathtt{F}$ or $[\![\mathbf{B}]\!] = \mathtt{T}$.
- Axioms: $P \Rightarrow P$, $P \Rightarrow Q \Rightarrow P$, $(P \Rightarrow Q \Rightarrow R) \Rightarrow (P \Rightarrow Q) \Rightarrow P \Rightarrow R$
- Inference Rules: $\dfrac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{A}}{\mathbf{B}}\, MP$ $\qquad \dfrac{\mathbf{A}}{[\mathbf{B}/P]\mathbf{A}}\, Subst$
- Example Proof: (Tertium non datur)

1 $(\mathbf{A} \Rightarrow (\mathbf{A} \Rightarrow \mathbf{A}) \Rightarrow \mathbf{A}) \Rightarrow (\mathbf{A} \Rightarrow (\mathbf{A} \Rightarrow \mathbf{A})) \Rightarrow \mathbf{A} \Rightarrow \mathbf{A}$ [Ax3 with $P = \mathbf{A}$, $Q = (\mathbf{A} \Rightarrow \mathbf{A})$ and $R = \mathbf{A}$]

2 $\mathbf{A} \Rightarrow ((\mathbf{A} \Rightarrow \mathbf{A}) \Rightarrow \mathbf{A})$ [Ax2 with $P = \mathbf{A}$ and $Q = (\mathbf{A} \Rightarrow \mathbf{A})$]

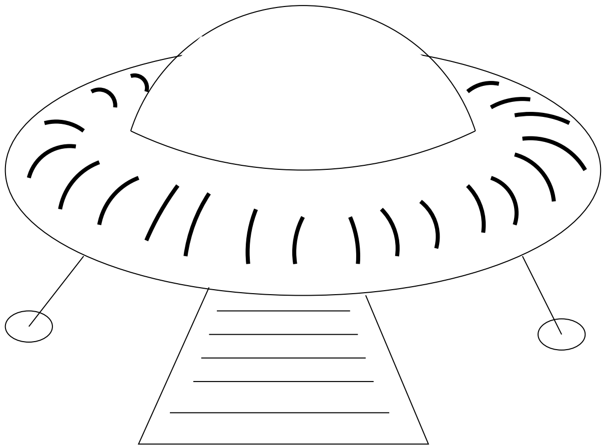
3 $(\mathbf{A} \Rightarrow (\mathbf{A} \Rightarrow \mathbf{A})) \Rightarrow \mathbf{A} \Rightarrow \mathbf{A}$ [MP 2 3]

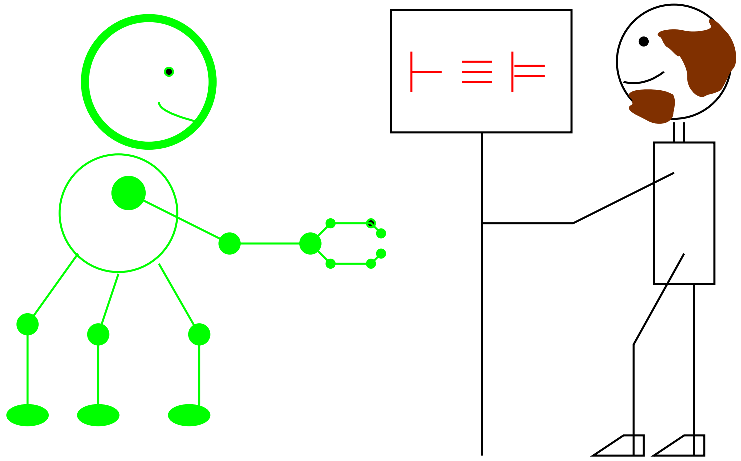
4 $\mathbf{A} \Rightarrow \mathbf{A} \Rightarrow \mathbf{A}$ [Ax2 with $Q = \mathbf{A}$]

5 $\mathbf{A} \Rightarrow \mathbf{A}$ [MP 3 4]

Properties of Calculi (Theoretical Logic)

- Correctness: (provable implies valid)
  - $\mathbf{A} \vdash \mathbf{B}$ implies $\mathbf{A} \models \mathbf{B}$ (equivalent: $\vdash \mathbf{A}$ implies $\models \mathbf{B}$)
- Completeness: (valid implies provable)
  - $\mathbf{A} \models \mathbf{B}$ implies $\mathbf{A} \vdash \mathbf{B}$ (equivalent: $\models \mathbf{A}$ implies $\vdash \mathbf{B}$)
- Goal: $\vdash \mathbf{A}$ iff $\models \mathbf{A}$ (provability and validity coincide)
  - To TRUTH through PROOF (CALCULEMUS [Leibniz ~1680])





What is Natural Language Understanding?

- Generally: Studying of natural languages and development of systems that can use/generate these.
- Here: Understanding natural language

  0) Speech processing: acoustic signal $\longrightarrow$ word net

  1) syntactic processing: word sequence $\longrightarrow$ phrase structure

  2) semantics construction: phrase structure $\longrightarrow$ (quasi-)logical form

  3) semantic-pragmatic analysis:
     (quasi-)logical form $\longrightarrow$ knowledge representation

  4) problem solving (application-specific using the generated knowledge)

- In this talk: logic/deductive modeling of 2) and 3).

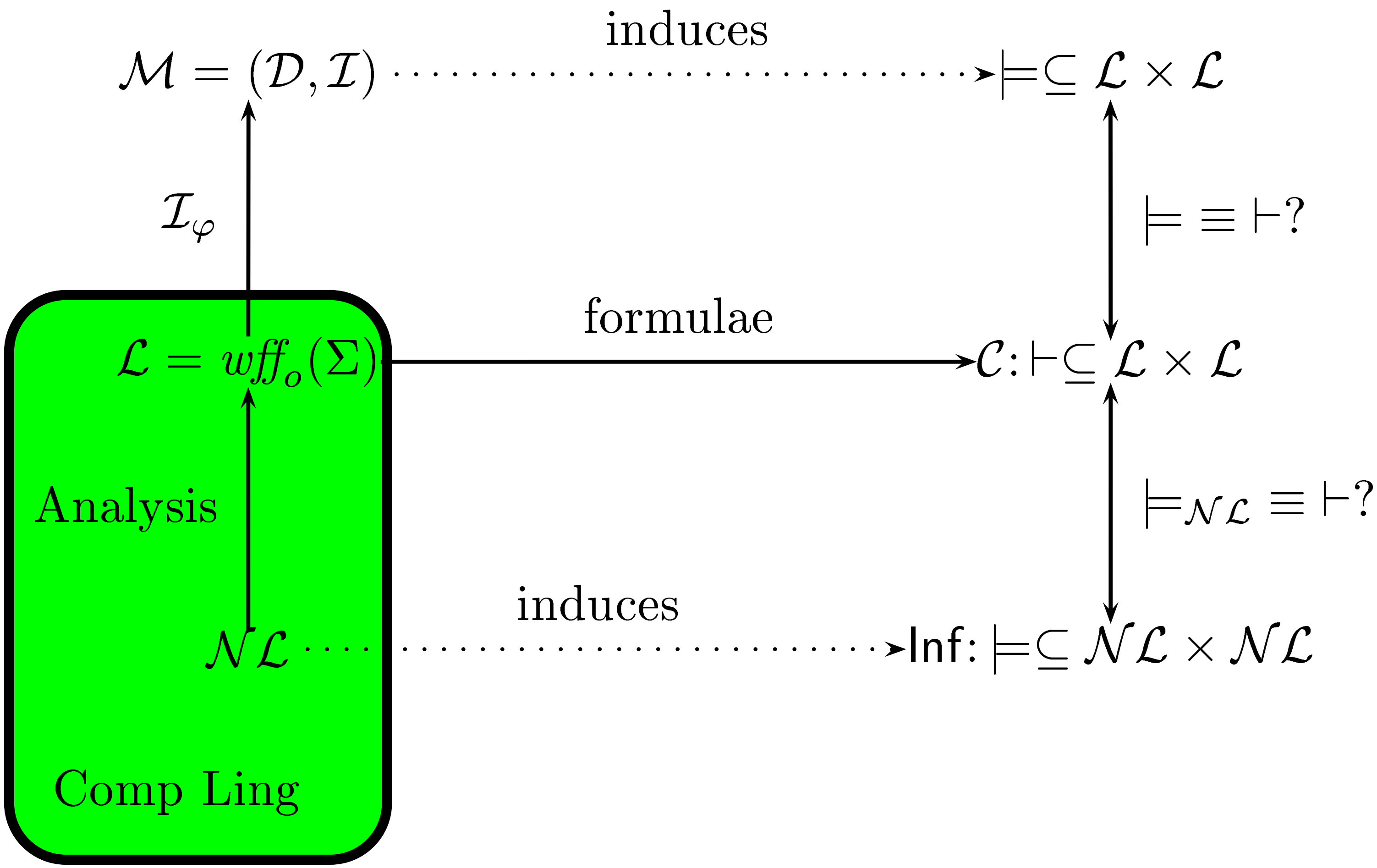
What is the State of the Art?

Coverage → vs. Analysis ↓	wide	narrow
Deep	Not there yet	Rule-based we are here
Shallow	Statistical Methods where the action is	no-one wants this

Modeling Natural Language Semantics

- Problem: Find formal (logic) system for the meaning of natural language
- History of ideas
  - Propositional logic [ancient Greeks like Aristotle]
    ***Every human is mortal*
  - First-Order Predicate logic [Frege $\leq$ 1900]
    ***I believe, that my audience already knows this.*
  - Modal logic [Lewis18, Kripke65]
    ***A man sleeps. He snores.* $(\exists X.\mathtt{man}(X)) \wedge \mathtt{snores}(X)$
  - Various dynamic approaches (e.g. DRT, DPL)
    ***Most men wear black*
  - Higher-order Logic, e.g. generalized quantifiers

Natural Language Semantics?



Three Principal Modes of Inference

- Deduction, we have seen already: $$\frac{rains \Rightarrow wet\_street \quad rains}{wet\_street}D$$
- Abduction ($\hat{=}$ explanation), $$\frac{rains \Rightarrow wet\_street \quad wet\_street}{rains}A$$
- Induction ($\hat{=}$ learning), $$\frac{wet\_street \quad rains}{rains \Rightarrow wet\_street}I$$
- Experience in Automated Reasoning:
  - they can be done with largely the same methods.

Learning Language in Logic (Why is this interesting)

- Logic (and related formalisms) allow to integrate world knowledge
  - explicity (gives more understanding than statistical methods)
  - transparently (symbolic methods are monotonic)
  - systematically (we can prove theorems about our systems)
- Signal + World knowledge makes more powerful model
  - Does not preclude the use of statistical methods to guide inference
- Problems with logic-based approaches
  - Where does the world knowledge come from? (Ontology problem)
  - How to guide search induced by logical calculi(combinatorial explosion)
- One possible answer: use learning techniques. (rest of the talk)

Logic Programming

- Idea: Use logic as a programming language
- Example: $\left\{ \begin{array}{rl} \text{Program:} & \text{Leibniz is human} \\ & \text{Sokrates is is human} \\ & \text{Sokrates is a Greek} \\ & \text{Every human is fallible} \\ \text{Query:} & \text{Are there fallible Greeks} \\ \text{Answer:} & \text{Yes, Sokrates} \end{array} \right.$
- How to achieve this? Restrict the logic calculus sufficiently that it can be used as computational procedure.
- [Kowalski]: Computation = Logic + Control

PROLOG (Horn clauses)

- Definition: Each clause contains at most one positive literal
  - $B_1 \vee \ldots \vee B_n \vee \neg A$ (`A :-B1,...,Bn.`)
  - Rule clause: `fallible(X) :- human(X)`
  - Fact clause: `human(sokrates)`
  - Program: set of rule and fact clauses
  - Query: `:- fallible(X),greek(X)`
- Theorem: *SLD Resolution is sound and complete on horn clauses*
  - Each instance of the query that is entailed by the program is subsumed by a positive answer.

PROLOG: Our Example

- Program:

```
human(sokrates) :-.
human(leibniz) :-.
greek(sokrates) :-.
fallible(X) :- human(X).
```

- Query: `:- fallible(X),greek(X).`
- Answer substitution: `[sokrates/X]`

Why Only Horn Clauses?

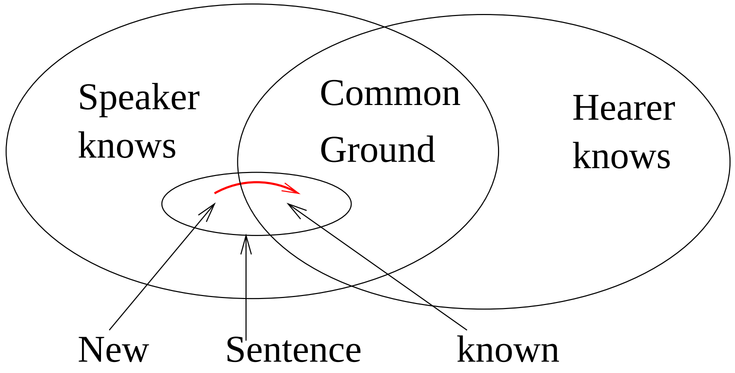
- General clauses of the form `A1,...,An :- B1,...,Bn.`
- e.g. `greek(sokrates),greek(perikles):-.`
  - Question: Are there fallible Greeks?
  - Indefinite answer: Yes, Perikles or Sokrates
  - Warning: how about Sokrates and Perikles?
- e.g. `greek(sokrates),roman(sokrates):-.`
  - Query: Are there fallible Greeks?
  - Answer: Yes, Sokrates, if he is not a roman
  - Is this abduction?????

Abduction = Inference to the best explanation

- Given: a logic $\mathcal{L}$:
- $\mathbf{A}$ is the best explanation for $\mathbf{B}$ in theory $\mathcal{T}$ ($\mathcal{T} \models_{\mathcal{L}} \mathbf{A} \rightsquigarrow \mathbf{B}$)
- call $\mathbf{A}$ explanans $\mathbf{B}$ explanandum
  - $\mathcal{T} \models_{\mathcal{L}} \mathbf{A} \Rightarrow \mathbf{B}$ (Explanation)
  - $\mathcal{T} \cup \{\mathbf{A}\}$ is consistent wrt. $\models_{\mathcal{L}}$ (non-trivial)
  - $\mathcal{T} \models_{\mathcal{L}} \mathbf{A} \Rightarrow \mathbf{C}$ for all $\mathbf{C}$ with $\mathcal{T} \models_{\mathcal{L}} \mathbf{C} \Rightarrow \mathbf{B}$ (best????)
- Abduction is non-monotonic reasoning
  - Explananses may have to be retracted in the light of further information

Application: Interpretation of NL [Stickel89,Hobbs93]

Parts of the information are known, others are new.

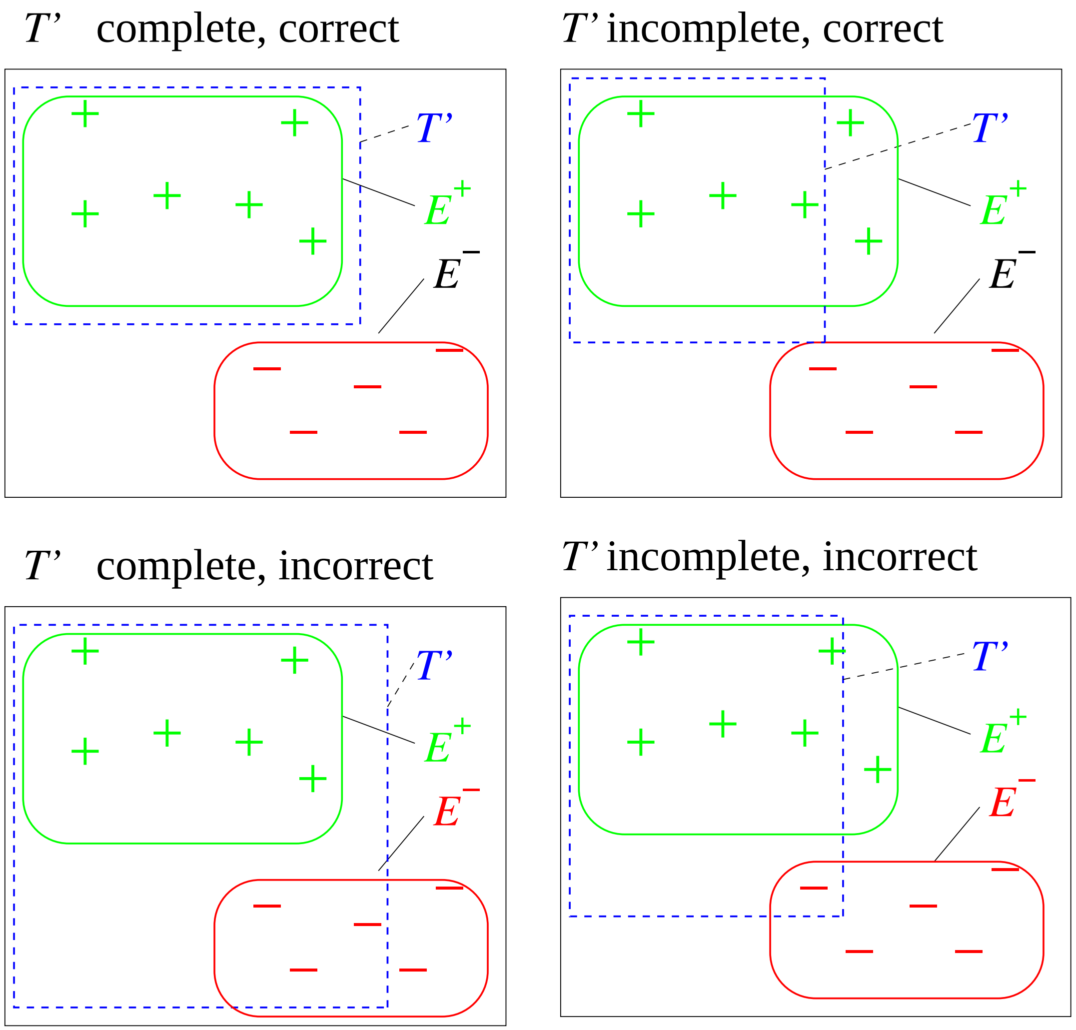


- Example: ... The car is red. ... (Logical form: $\mathcal{F} := car(x) \wedge red(x)$)
- Explain the logical form (explanandum) from the common ground $\mathcal{T}$ and extend $\mathcal{T}$ by the explanations.
  - If $car(a) \in \mathcal{T}$, then $\mathcal{T} \models_{PL1} red(a) \leadsto \mathcal{F}$
  - If $red(b) \in \mathcal{T}$, then $\mathcal{T} \models_{PL1} car(b) \leadsto \mathcal{F}$
  - else $\mathcal{T} \models_{PL1} red(c), car(c) \leadsto \mathcal{F}$
- Acquire only the new knowledge given in the sentence

Induction (ILP): Inductive Logic Programming

- Infers concept descriptions from examples. (Machine learning)
- Given
  - A concept description language $\mathcal{L}$
    (concepts as relations $\mathcal{K} \subseteq \mathcal{U}^*$ on the universe $\mathcal{U}$)
  - sets $\mathcal{E}^+$ and $\mathcal{E}^-$ of positive and negative examples
  - A consistent concept description $\mathcal{T}$ (background theory), such that $\mathcal{T} \not\models e^+$ for at least one $e^+ \in \mathcal{E}^+$.
- Find concept description $\mathcal{T}' \supseteq \mathcal{T}$, such that
  $\mathcal{T}' \models \mathcal{E}^+$ (Completeness) $\mathcal{T}' \not\models \mathcal{E}^-$ (Correctness/Consistency)
- Non-monotonic rule of inference
  (concept descriptions can be revised by further examples)

Coverage Of Concept Descriptions



Induction (Success Criteria)

- Coverage
- Classification Accuracy %age of the correctly classified new Objects
- Transparency Is the concept description readable for humans?
- Statistic significance
  Are domain regularities discovered or is the coverage only by chance?
- Information content
  - Success relative to the problem difficulty
  - Uses the a priori-likelihood of the concept to be learned
  - e.g. in medical diagnosis:
    Classification in common diagnosis contains less information than diagnosis of a rare disease from few examples.

Induction (Example $\mathcal{L}$ = Horn logic)

- Background theory $\mathcal{T}$

  `parent(anna,maria).   parent(anna,tim).`

  `parent(tim,eva).   parent(paul,anna).`

  `female(anna).   female(maria).   female(eva).`

- positive examples $\mathcal{E}^+$ `daughter(maria,anna), daughter(eva,tim)`
- negative examples $\mathcal{E}^-$ `daughter(tim,anna), daughter(eva,anna)`
- Different levels of concept description:
  - Extensional: $\mathcal{K}$ set of facts

    (`daughter(maria,anna), daughter(eva,tim), daughter(anna,paul)`)
  - Intensional: $\mathcal{K}$ set of rules (`daughter(X,Y) :- female(X),parent(Y,X).`)
- Learn intensional concept descriptions, instead of huge sets of facts.

Induction as Search

- Induction searches in hypotheses space of concept descriptions $\mathcal{T}'$
- structuring the space of hypotheses by subsumption: $\mathcal{K} \leq \mathcal{K}'$, iff $\mathcal{K}' \models \mathcal{K}$
- search procedures
  - Bottom-Up (generalizing) procedures

    search for most specific generalizations of the positive examples
  - Top-Down (specializing) procedures

    search most general specializations of an initial concept description
- $\leq$ allows to prune large parts of the search space: Let $\mathbf{C} \leq_\theta \mathbf{D}$
  - during generalization of $\mathbf{D}$ to $\mathbf{C}$:

    If $\mathbf{D}$ is inconsistent ($\mathcal{T}, \mathbf{D} \models e^-$), then $\mathbf{C}$ is too; do not generalize further.
  - during specialization of $\mathbf{C}$ to $\mathbf{D}$:

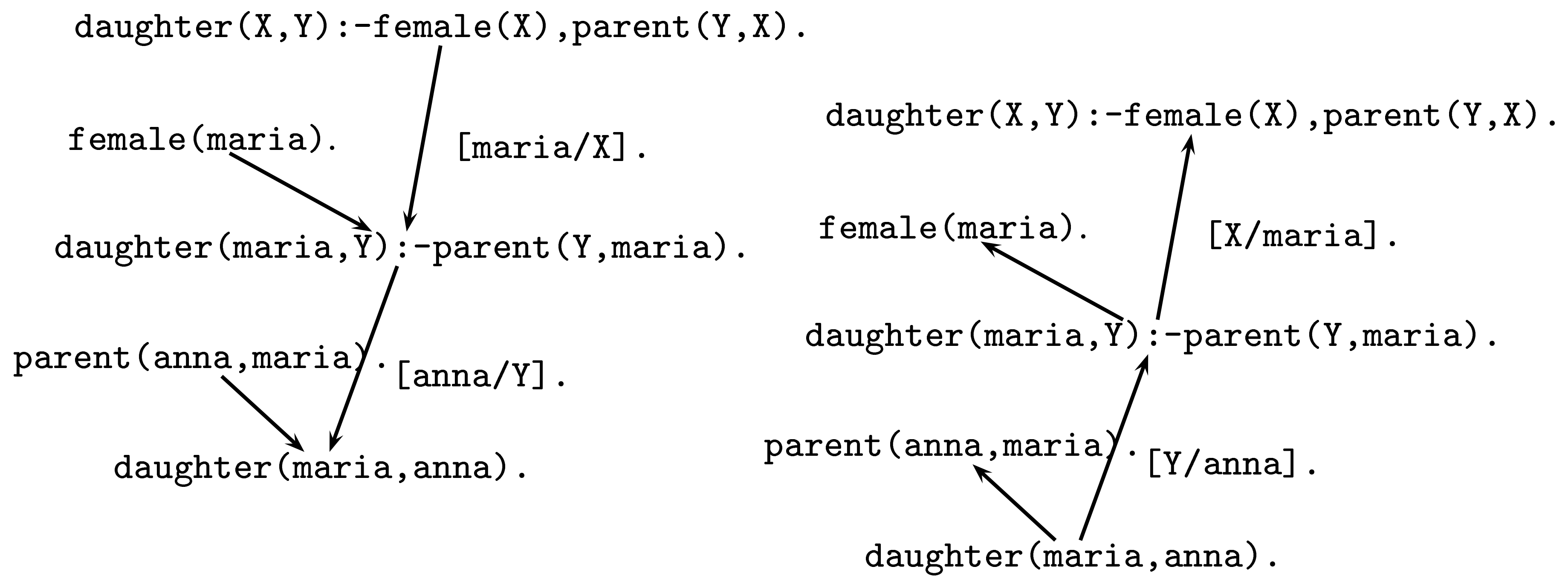
    If $\mathbf{C}$ is incomplete ($\mathcal{T}, \mathbf{C} \not\models e^+$), then $\mathbf{D}$ is too; do not specialize further.

Inverse Resolution (bottom-up, top-down, subsumption)

- e.g. Induct the concept `daughter`
  - Background theory $\mathcal{T}$: `parent(anna,maria).` `female(maria).`
  - positive examples $\mathcal{E}^+$: `daughter(maria,anna)`

---

resolution proof | Inverse resolution proof



ILP for simple Grammar Learning

- represent the rule $VP \longrightarrow VP\ \ MOD$ as `vp(S,E) :- vp(S,M),mod(M,E)`
  - For instance in Chart parsing: $_0$ *She*$_1$ *ran*$_2$ *quickly*$_3$
  - With rule we can infer `vp(1,3)`, `vp(1,2)`, `mod(2,3)`, `vp(1,3)`.
  - Idea: use ILP to learn rules from
    * training example: `vp(1,3)`
    * background theory: `np(0,1) word(she,0,1) vp(1,2) word(ran,1,2) mod(2,3) word(quickly,2,3)`
  - can be done with the methods described above
  - Problem: over-generalizes, e.g. `vp(S,E):-`, `vp(S,E):-vp(S,M)`
  - Use Closed-World-Assumption or negative examples.

Scaling Up without Dumbing Down [Mooney]

- Scaling up: Corpus-based Methods for Shallow Problems (Dumbing Down)
  - Language Technology: e.g. information retrieval,... (classification)
  - Artificial Intelligence: Language Understanding, (modeling)
- Mooney's Conclusion: need ILP to scale up deep methods
  - learn extensive sets of world knowledge.
  - learn search guidance for inference procedures.
- Experiments for ILP in semantic analysis: Chill, Rapier, ... show that ILP-based methods can compete with statistical ones (in some areas)
- My own research: (see me if you are interested)
  - now: build logic/inference-based models for NLU (model-generation)
  - soon?: tune them with corpus-based methods (need help)