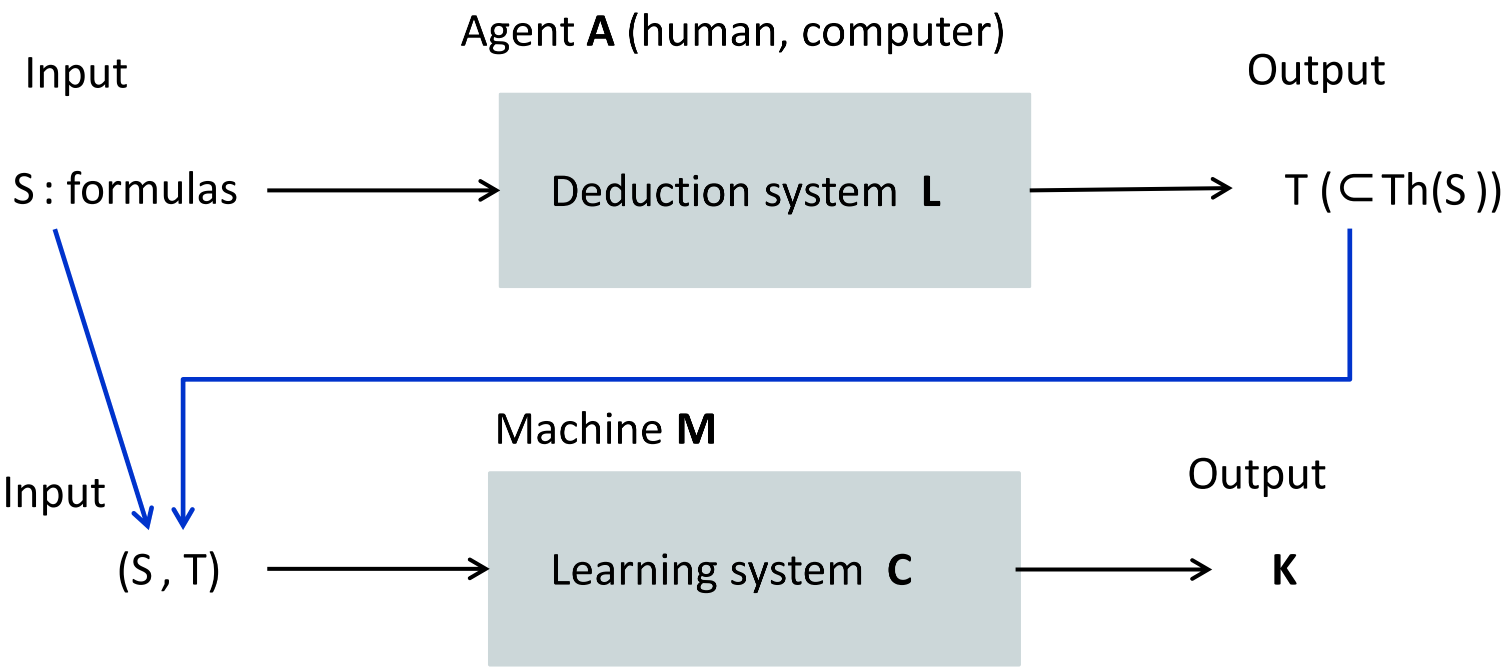
# Can Machines Learn Logics?

**Chiaki Sakama** (Wakayama University)

**Katsumi Inoue** (National Institute of Informatics)

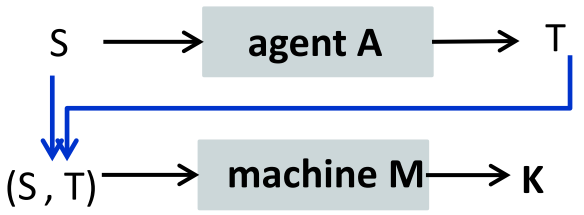
# Learning Logics




- Given input (S, T), a machine **M** produces an axiomatic system **K**.
- **K** is sound (resp. complete) wrt **L** if **K** ⊆ **L** (resp. **L** ⊆ **K**).

# Remarks

- An agent **A** plays the role of a teacher who provides training examples representing premises along with entailed consequences.
- The output **K** is refined by incrementally providing examples.
- An agent **A** could be a system of arbitrary logic, e.g. nonmonotonic logic, modal logic, fuzzy logic, as far as it has a formal system of inference.

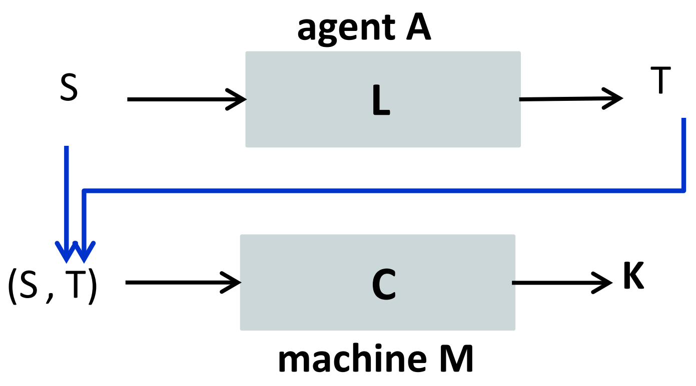


# Remarks

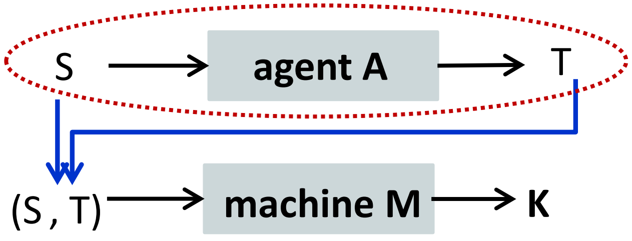
- Alternatively, we can consider a framework in which a teacher agent **A** is absent.
- In this case, given input-output pairs (S ,T) as data, the problem is whether a machine **M** can find an unknown logic (or axiomatic system) that produces a consequence T from a premise S.

# Challenging Problems

- Can we develop a sound and complete algorithm **C** for learning a classical or non-classical logic **L**?
- Is there any difference between learning axioms and learning inference rules?
- Does a machine **M** discover a new axiomatic system **K** such that **K** |- F iff **L**|- F for any formula F?



# A simple case study: Learning deduction rules




- *S*: a set of atomic formulas containing $hold(F)$ where *F* is a formula in propositional logic.

- An agent **A** with an inference system **L** performs the inference:

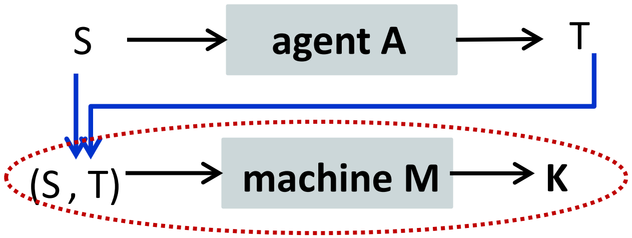
  from $hold(p)$ and $hold(p \supset q)$ infer $hold(q)$

  where *p* and *q* are propositional variables.

- In this case, given a finite set *S* of atoms as an input, **A** outputs the set:

$$T = S \cup \{\, hold(q) \mid hold(p) \in S \text{ and } hold(p \supset q) \in S \,\}$$

# A simple case study: Learning deduction rules




- Given each pair (S, T) as an input, consider a machine **M** that constructs a rule:

$$A \leftarrow \bigwedge_{B_j \in S} B_j \qquad \text{where} \quad A \in T \setminus S$$

- For example, given the set

$S = \{\, hold(p), hold(r), hold(p \supset q), hold(p \supset r), hold(r \supset s) \,\}$,

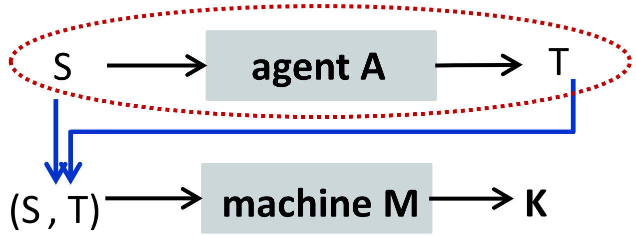
two atoms $hold(q)$ and $hold(s)$ are in $T \setminus S$. Then the following two rules are constructed

$$hold(q) \leftarrow hold(p) \wedge hold(r) \wedge hold(p \supset q) \wedge hold(q \supset r) \wedge hold(r \supset s)$$

$$hold(s) \leftarrow hold(p) \wedge hold(r) \wedge hold(p \supset q) \wedge hold(q \supset r) \wedge hold(r \supset s)$$

- The condition contains atoms which do not contribute to deriving the atom in the consequence.

# A simple case study: Learning deduction rules




- For each pair (S, T) from **A** such that $T \setminus S \neq \phi$, assume that the following rule R is constructed.

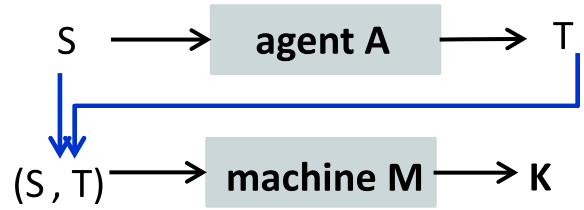
$$A \leftarrow \bigwedge_{B_j \in S} B_j \quad \text{where} \quad A \in T \setminus S$$

- Then select a subset $S_i$ of S and give it as an input to **A**. If its output $T_i$ still contains the atom A, replace R with

$$A \leftarrow \bigwedge_{B_j \in S_i} B_j \quad \text{where} \quad A \in T_i \setminus S_i$$

- By continuing this process, find a minimal set $S_i$ satisfying $A \in T_i$ . Such $S_i$ contains atoms that are necessary and sufficient for deriving atoms in $T_i \setminus S_i$ .

# A simple case study: Learning deduction rules




- In the above example, there is the unique minimal set

  $S_1$={ hold(p), hold(p ⊃ q)}

 that satisfies hold(q) ∈ $T_1$, and there are two minimal sets that contain hold(s) in their output:

  $S_2$= { hold(r), hold(r ⊃ s) }

  $S_3$= { hold(p), hold(p ⊃ r), hold(r ⊃ s) }

- Then the following 3 rules are obtained by replacing S with $S_i$

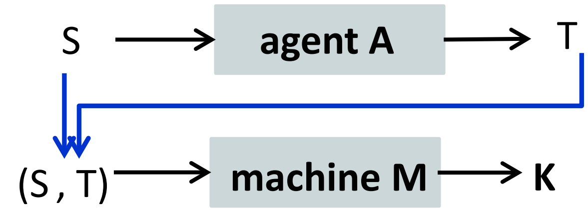
  hold(q) ← hold(p) ∧ hold(p ⊃ q)  :**Modus Ponens**

  hold(s) ← hold(r) ∧ hold(r ⊃ s)

  hold(s) ← hold(p) ∧ hold(p ⊃ r) ∧ hold(r ⊃ s)

  :**Multiple Modus Ponens**

# A simple case study: Learning deduction rules




- Using the technique, the following inference rules are obtained:
  - $hold(\neg p) \leftarrow hold(\neg q) \wedge hold(p \supset q)$ : **Modus Tollens**
  - $hold(p \supset r) \leftarrow hold(p \supset q) \wedge hold(q \supset r)$ : **Hypothetical Syllogism**
  - $hold(p) \leftarrow hold(p \vee q) \wedge hold(\neg q)$ : **Disjunctive Syllogism**
  - $hold(p) \leftarrow hold(q) \wedge hold(p \supset q)$ :
    **Fallacy of Affirming the Consequence** (for **abductive inference**)

- An interesting question is whether the same or a similar technique can be applied for learning non-logical systems (e.g. pragmatic rules of inference, conversational implicature, etc).