1. First recall that if a variable appears only in the body of a rule, then it is assumed to be existentially quantified. For instance, in

\[ \text{grandparent}(X,Y) :\rightarrow \text{parent}(X,Z), \text{parent}(Z,Y). \]

\( Z \) is existentially quantified, and scoped by the two universally quantified variables \( X \) and \( Y \). This rule represents the implication

\[ \forall x \forall y ((\exists z \text{parent}(x,z) \land \text{parent}(z,y)) \Rightarrow \text{grandparent}(x,y)) \]

Note that the existential quantifiers here are in the body of the rule whereas the patterns we need to find are ones where they are in the head of the rule. The ILP systems we learnt in class cannot mine either of the given patterns. If you thought \( \text{enrolledin}(X,\text{cs6604},'A'). \) was a valid representation for the first sentence, you would be incorrect since \( X \) here is universally quantified.

There is a system called PCL that has been developed with such patterns in mind (see S.-H. Nienhuys-Chung, W. Van Laer, J. Ramon, and L. De Raedt, Generalizing Refinement Operators to Learn Prenex Conjunctive Normal Forms, Proceedings of the 9th International Workshop on Inductive Logic Programming, Lecture Notes in Artificial Intelligence, Vol. 1634, pages 245–256, Springer Verlag, 1999). If you answered WARMR as a possible system choice, you will receive no credit because WARMR uses query extensions to make implications between two existentially quantified sentences, and only those implications where the head subsumes the body. The given patterns do not fit these constraints.

2. If your rules are of the form:

\[ \text{willgraduate}(X) :\rightarrow \text{hastaken}(X,\text{CS3414}), \text{hastaken}(X,\text{CS4604}), \text{hastaken}(X,\text{CS5604}). \]

\[ \text{willgraduate}(X) :\rightarrow \text{hastaken}(X,\text{CS3414}), \text{hastaken}(X,\text{CS4604}), \text{hastaken}(X,\text{CS5614}). \]

\[ \text{willgraduate}(X) :\rightarrow \text{hastaken}(X,\text{CS3414}), \text{hastaken}(X,\text{CS4604}), \text{hastaken}(X,\text{CS5804}). \]

... 

\[ \text{willgraduate}(X) :\rightarrow \text{hastaken}(X,\text{CS5614}), \text{hastaken}(X,\text{CS5804}), \text{hastaken}(X,\text{CS6604}). \]

then LINUS (and its descendants) can learn these rules. If, however, you formulated the rule as:

\[ \text{willgraduate}(X) :\rightarrow \text{hastaken}(X,A), \text{hastaken}(X,B), \text{hastaken}(X,C), \]

\[ A<>B, A<>C, B<>C. \]

then only MINUS (this is the name we christened to the last algorithm in the [Lavrac and Flach, 2001] paper) can mine this pattern. In this context, the entire body of the rule has to be abstracted in the feature construction phase into a single predicate (e.g., \( f1(X) \)), and made available for propositionalization.

3. Propositionalization is a possible strategy for recursive rule learning only if restrictions are made to guarantee algorithmic convergence and we can curtail the combinatorial explosion of possibilities (e.g., limits on depth of recursion). Given such restrictions many propositionalization software (e.g., DINUS) can be used for recursive rule learning.
4. Both attribute-value learning and ILP are suitable for inferring FDs from data. For an ILP system to be used for mining FDs, we require the important properties of multiple predicate learning (since there is no target predicate), simultaneous covering (since there is no notion of sequentiality and we are performing descriptive induction) and, preferably, the ability to create existentially quantified statements. In addition, algorithms that learn from interpretations are especially suitable since the task involves several independent assessments of database coverage. TERTIUS and WARMR are examples of ILP systems that can be used for mining FDs.


5. Kowalski normal form is more general than Horn clauses but has the same representational power as first-order logic. In fact, it can be used as a canonical form of sentences for theorem proving and reasoning. There are many cases where we would need this normal form, the best example we found was:

\[
\text{International}(x) \land \text{VTGradStudent}(x) \Rightarrow \text{Indian}(x) \lor \text{Chinese}(x)
\]

A descriptive learner such as TERTIUS is capable of learning sentences in this form (already covered in class). You will receive full credit if you made the above observations and included details on (i) the name of the software, (ii) the algorithm used to search within the space of sentences, and (iii) a critical survey of some deficiencies (e.g., the use of a heuristic for driving the search).