Finding Accurate Frontiers: A Knowledge-Intensive Approach to Relational Learning

Michael Pazzani and Clifford Brunk
Information and Computer Science
University of California
Irvine, CA 92717
pazzani@ics.uci.edu
brunk@ics.uci.edu

Research supported by Air Force Office of Scientific Research Grant, F49620-92-J-0430

Outline

A. Using existing knowledge to improve the accuracy of learning

B. Background
   1. Inductive Learning from relational data (FOIL)
   2. Combining Inductive and Explanation-Based Learning

C. Problems with predefined levels of generality for analytic learning

D. Frontiers: Dynamically selecting the generality of analytic learning

E. Experimental Evaluation

F. Conclusion: *Determining the generality of entailments to discriminate positive from negative training examples leads to more learning rules that are more accurate on unseen data.*

Knowledge-based Systems

Two commonly used approaches to creating rule-based systems:

1. **Knowledge Engineering–** manually encoding expert knowledge
   - Time and Labor intensive to construct very accurate rules
   - Time and Labor intensive to maintain rule-base

2. **Inductive Learning–** creating rules encoding regularities in training examples
   - Requires many examples to learn accurate rules
   - Rules may not be understandable to human experts

Using existing knowledge to improve the accuracy of learning

Given:

A set of classification rules

A set of classified training examples

Produce:

A set of classification rules consistent with the training examples

Objective:

Learn rules at least as accurate as existing rules

Learn rules at least as accurate as those produced by induction

- Existing rules may be incomplete an/or incorrect
- Existing rules may need updating due to changes in environment

- Inductive learning can be focused to find regularities among examples that are not correctly classified by existing rules

First-Order Inductive Learner (Quinlan, 90)

Finding the smallest horn clause theory is NP-complete

```
no_payment_due(?P) :- enlisted(?P ?Org) & armed_forces(?Org).
no_payment_due(?P) :- disabled(?P).
```

```
Learn-clauses(Pos, Neg):
Until Pos is empty
    Let Clause = learn-clause(Pos, Neg)
    remove examples covered by Clause from Pos

Learn-clause(Pos,Neg):
    Initialize Body to True
    Until Neg is empty
        Let Literal = Best-Literal(Pos,Neg)
        Remove examples not covered by Clause from Pos and Neg
```

$$p_1\left(\log_2\left(\frac{p_1}{p_1+n_1}\right)-\log_2\left(\frac{p_0}{p_0+n_0}\right)\right)$$

LOAN	
Name	**Not Due**
Barbara-Nelson	True
Edgar-Sheppard	True
Lisa-Ford	True
Michael-Obrein	True
Michael-Dixon	False
Karen-Davis	False
David-Tyson	False
Michael-Adams	False

ENLIST	
Name	**Org.**
Lisa-Ford	Air-Force
Michael-Obrein	Navy
David-Tyson	Peace-Corp

ENROLLED		
Name	**School**	**Units**
Barbara-Nelson	UCLA	12
Edgar-Sheppard	UCI	14
Lisa-Ford	UCLA	3
Karen-Davis	CMU	6
David-Tyson	MIT	4

ARMED FORCES	
Org.	**Service**
Air-Force	True
Navy	True
Army	True
Marines	True
Peace-Corps	False

```
no_payment_due(?N) :-
        enrolled(?N ?S ?U)
```

LOAN	
Name	**Not Due**
Barbara-Nelson	True
Edgar-Sheppard	True
Lisa-Ford	True
Michael-Obrein	True
Michael-Dixon	False
Karen-Davis	False
David-Tyson	False
Michael-Adams	False

ENLIST	
Name	**Org.**
Lisa-Ford	Air-Force
Michael-Obrein	Navy
David-Tyson	Peace-Corp

ENROLLED		
Name	**School**	**Units**
Barbara-Nelson	UCLA	12
Edgar-Sheppard	UCI	14
Lisa-Ford	UCLA	3
Karen-Davis	CMU	6
David-Tyson	MIT	4

ARMED FORCES	
Org.	**Service**
Air-Force	True
Navy	True
Army	True
Marines	True
Peace-Corps	False

```
no_payment_due(?N) :-
        enrolled(?N ?S ?U) & ?U>11
```

LOAN	
Name	**Not Due**
Barbara-Nelson	True
Edgar-Sheppard	True
Lisa-Ford	True
Michael-Obrein	True
Michael-Dixon	False
Karen-Davis	False
David-Tyson	False
Michael-Adams	False

ENLIST	
Name	**Org.**
Lisa-Ford	Air-Force
Michael-Obrein	Navy
David-Tyson	Peace-Corp

ENROLLED		
Name	**School**	**Units**
Barbara-Nelson	UCLA	12
Edgar-Sheppard	UCI	14
Lisa-Ford	UCLA	3
Karen-Davis	CMU	6
David-Tyson	MIT	4

ARMED FORCES	
Org.	**Service**
Air-Force	True
Navy	True
Army	True
Marines	True
Peace-Corps	False

```
no_payment_due(?N):-enlist(?N ?O)
                    & armed-forces(?O)
```

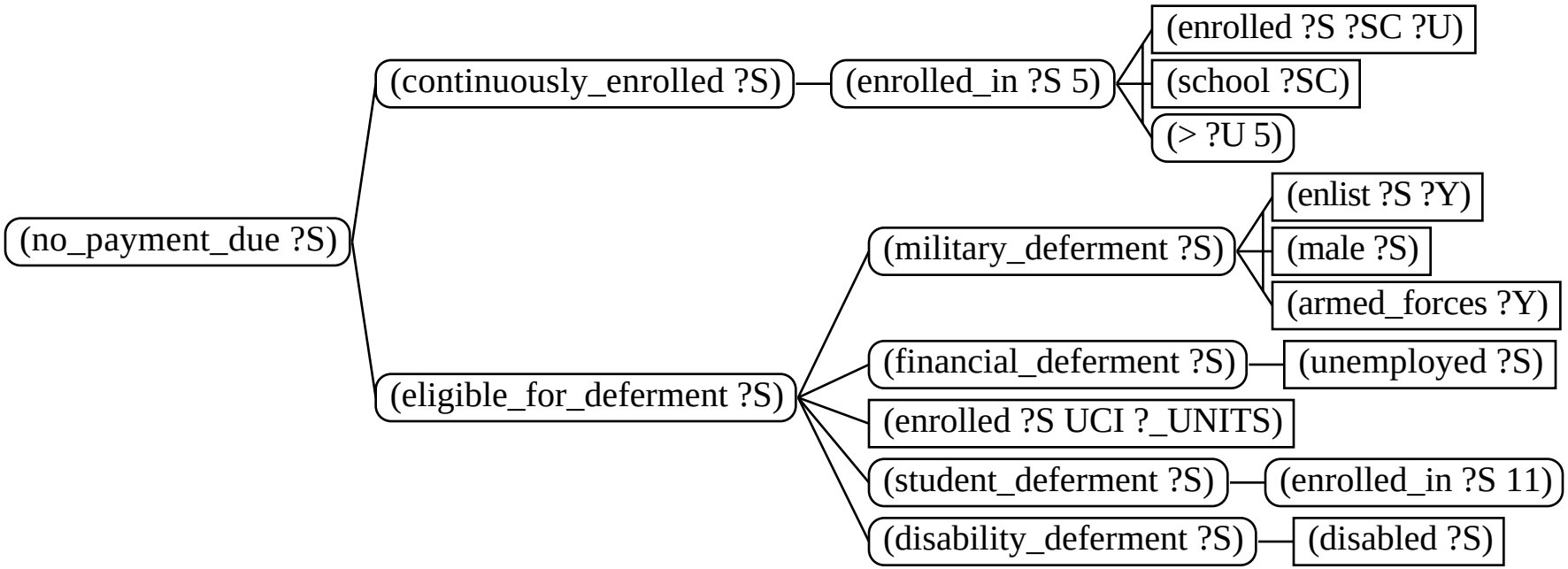
First-Order Combined Learner (Pazzani & Kibler, 92)

Two ways of adding literals

1. Inductive (as in FOIL)
2. Operationalization guided by information-gain

Whichever has the highest information-gain is used

Head :- $Conjunction_{Inductive}$ & $Conjunction_{Operationalize}$



An information-based approach to operationalization

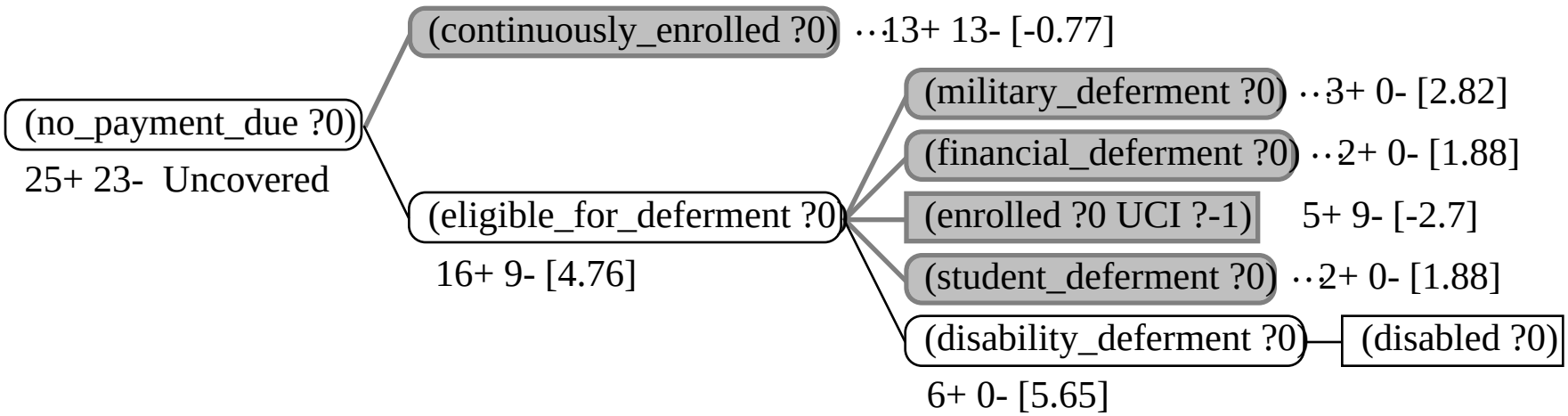
- **EBL** (Mitchell et al, 1986)

First proof of a single example

```
no_payment_due(john), disabled(john)
```

```
no_payment_due(?P) :- disabled(?P).
```

- FOCL- Proof that best discriminates training data



Problems with a static definition of operationality– 1
Overspecialization of correct general concepts

The learned concept may not include some combinations of operational predicates although there is no evidence that these specializations are incorrect.

a :- b,d

b:-f,g,h
b:- i,j

d:- m,n,o
d :- p,q
d:- r,s,t

a:-f,g,h,m,n,o
a:-f,g,h,p,q
a:-f,g,h,r,s,t

a:- i,j,m,n,o
a:- i,j,p,q
a:- i,j,r,s,t

Problems with a static definition of operationality– 2
Concepts learned may be too specialized

Incorrect concepts results in replication of induction

a :- b,d

b:-f,g,h
b:- i,j

d:- m,n,o
d :- p,q
d:- r,s,t

a:- f,h,m,n,o,**g**
a:- f,h,p,q,**g**
a:- f,h,r,s,t,**g**

a:- i,j,m,n,o
a:- i,j,p,q
a:- i,j,r,s,t

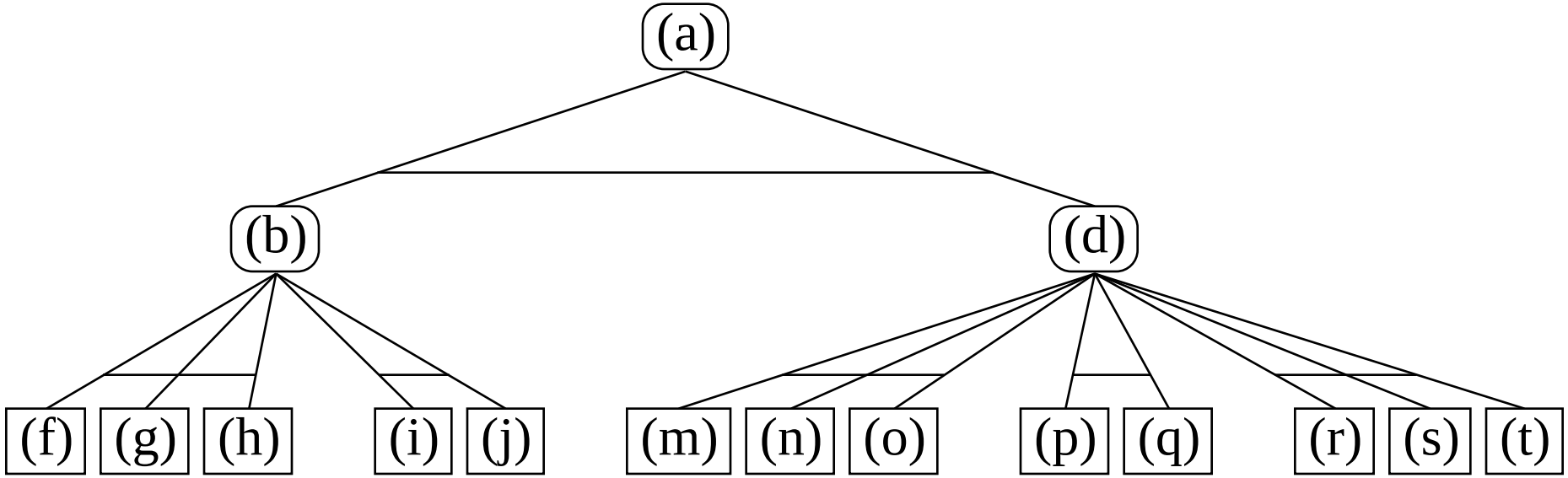
a:- f,h,d,**g**

a:- i,j,d.

• For FOCL to recover from this error induction must induce **g** 3 times.
• Induction is less likely to find **g** 3 times from 3 partitions of a data set than one on the union of the data sets

Frontiers

1. Non-operational predicates (e.g., `b`)
2. A disjunction of two or more clauses that define a non-operational predicate (e.g., `(m o) (p q)`)
3. Not all literals from a conjunction (`n`)



Frontiers

There are $2^{m^d n^d}$ frontiers where
m is the number of conjunctions per clause,
n the number of clauses per rule
d the depth of the proof tree.
(2^{12} in student loan, 2^{25} in KRK chess, $2^{2,046,395}$ in NynexMax)

Cohen (1991) Find all proofs of all examples, find a cover of examples
• ANA-EBL Retain *k* nodes of proof trees (and all remaining leaves)
- $O(n^k)$ where *n* is the size of a proof tree
- Restricted to small values of *k* (2)

Speed-up learning: Assumes domain theory is correct and tries to improve performance of queries
• Braverman & Russell (88), Hirsh (88), Keller(88), Segre (88)

A greedy approach to finding frontiers

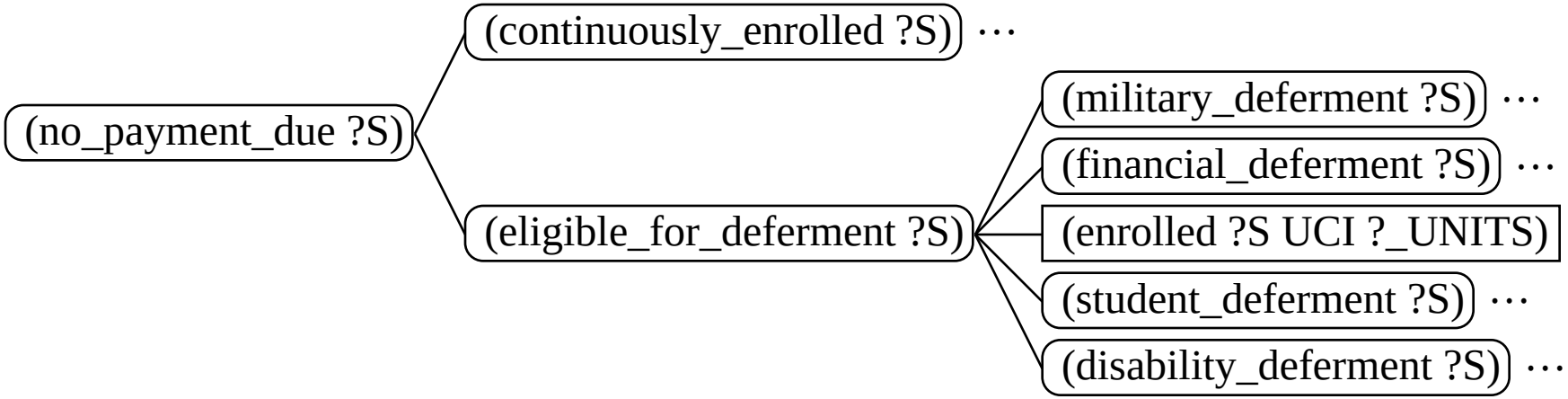
- Hill-climbing search with transformation operators.

```
Initialize current-frontier to target-concept
Until no operator increases information gain
   Apply operators to derive new frontiers
   Set current-frontier to derived frontier with max gain
```

- Rule specialization
- Specialization by removing a disjunct
- Generalization by adding a disjunct
- Generalization by literal deletion

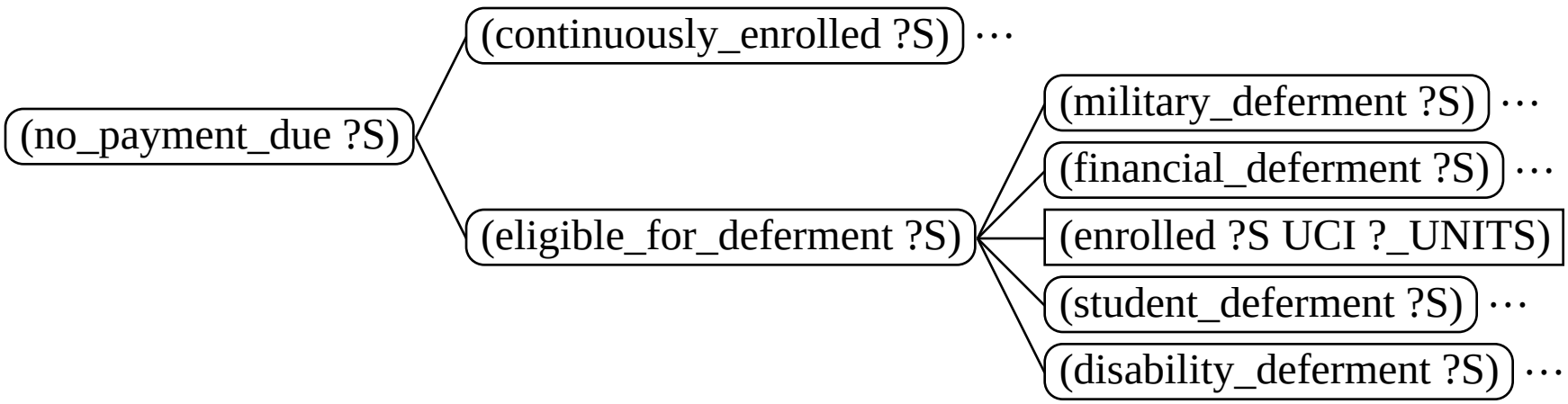
Rule specialization

If there is a frontier containing a literal p, and there are exactly *n* rules of the form $p \leftarrow \beta_1, \ldots, p \leftarrow \beta_i, \ldots, p \leftarrow \beta_n$, then *n* frontiers formed by replacing p with β_i are evaluated



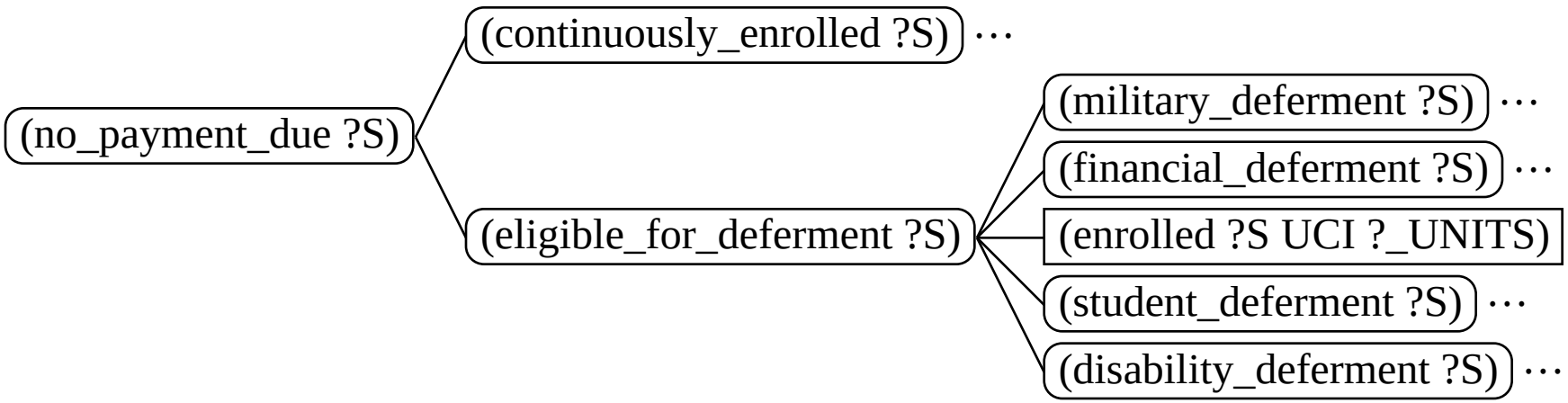
Specialization by removing a disjunct -1

If there is a frontier containing a literal p, and there are *n* rules of the form $p \leftarrow \beta_1, \ldots, p \leftarrow \beta_i, \ldots, p \leftarrow \beta_n$, then *n* frontiers formed by replacing p with $\beta_1 \ \ldots \ \beta_{i-1} \ \beta_{i+1} \ \ldots \ \beta_n$ are evaluated (provided $n > 2$).



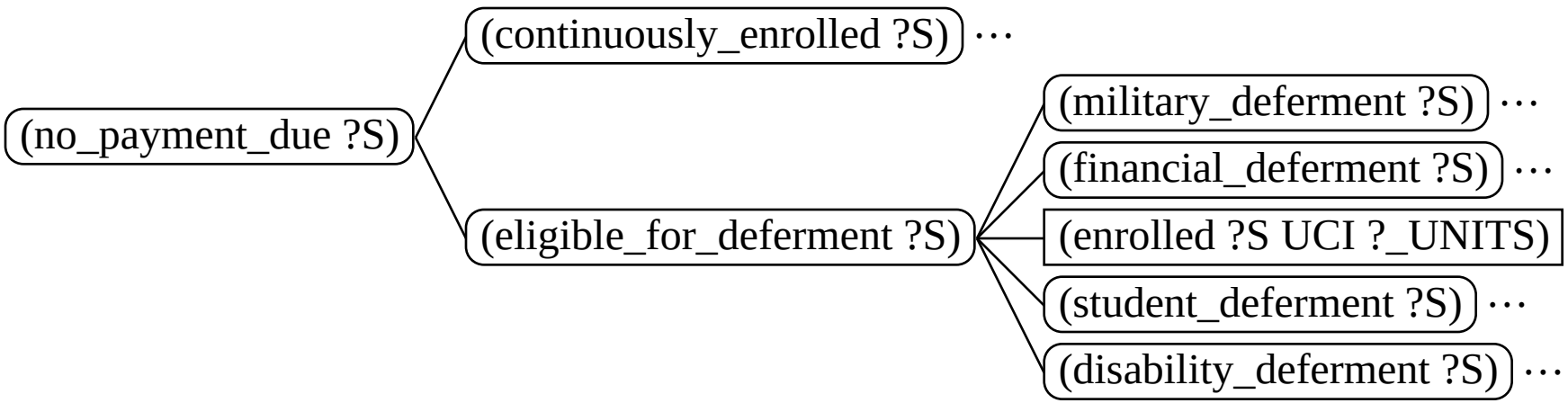
Specialization by removing a disjunct -2

If there is a frontier containing a disjunction $\beta_1 \ldots \beta_{i-1}\ \beta_i\ \beta_{i+1} \ldots \beta_m$, then *m* frontiers replacing this disjunction with $\beta_1 \ldots \beta_{i-1}\ \beta_{i+1} \ldots \beta_m$ are evaluated (provided $m > 2$).



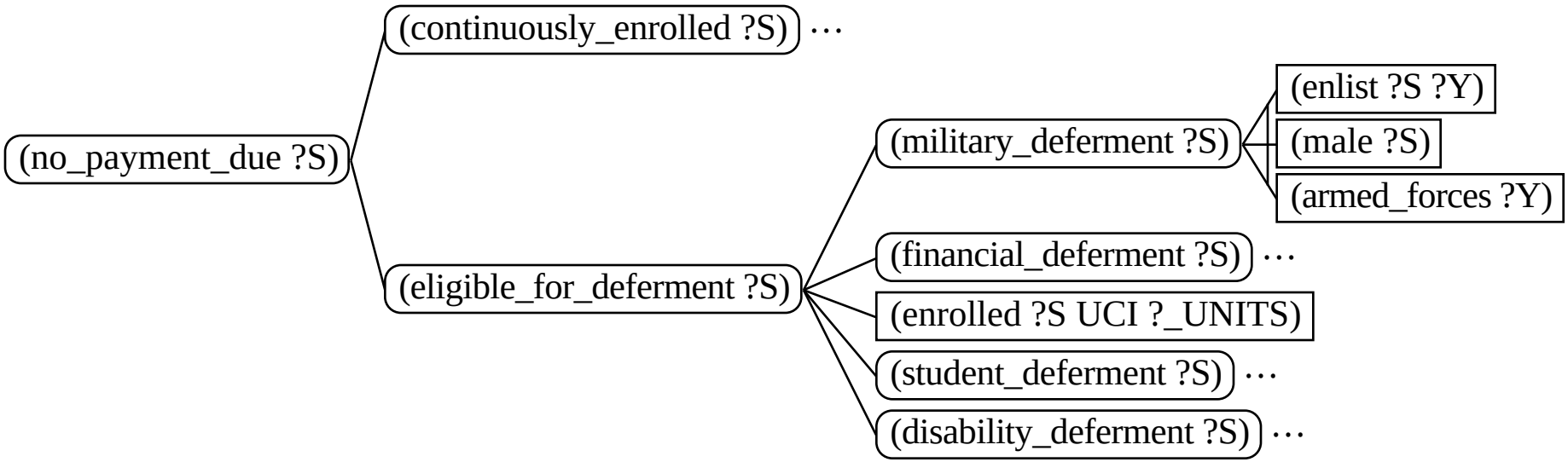
Generalization by adding a disjunct

If there is a frontier containing a (possibly trivial) disjunction of conjunction of literals $\beta_1 \ldots \beta_{i-1}\ \beta_{i+1} \ldots \beta_m$ and there are rules of the form $p \leftarrow \beta_1, \ldots, p \leftarrow \beta_{i-1}, p \leftarrow \beta_i, p \leftarrow \beta_{i+1}, \ldots, p \leftarrow \beta_n$ and $m < n-1$, then $n-m$ frontiers replacing the disjunction $\beta_1 \ldots \beta_{i-1}\ \beta_{i+1} \ldots \beta_m$ with $\beta_1 \ldots \beta_{i-1}\ \beta_i\ \beta_{i+1} \ldots \beta_m$ are evaluated.

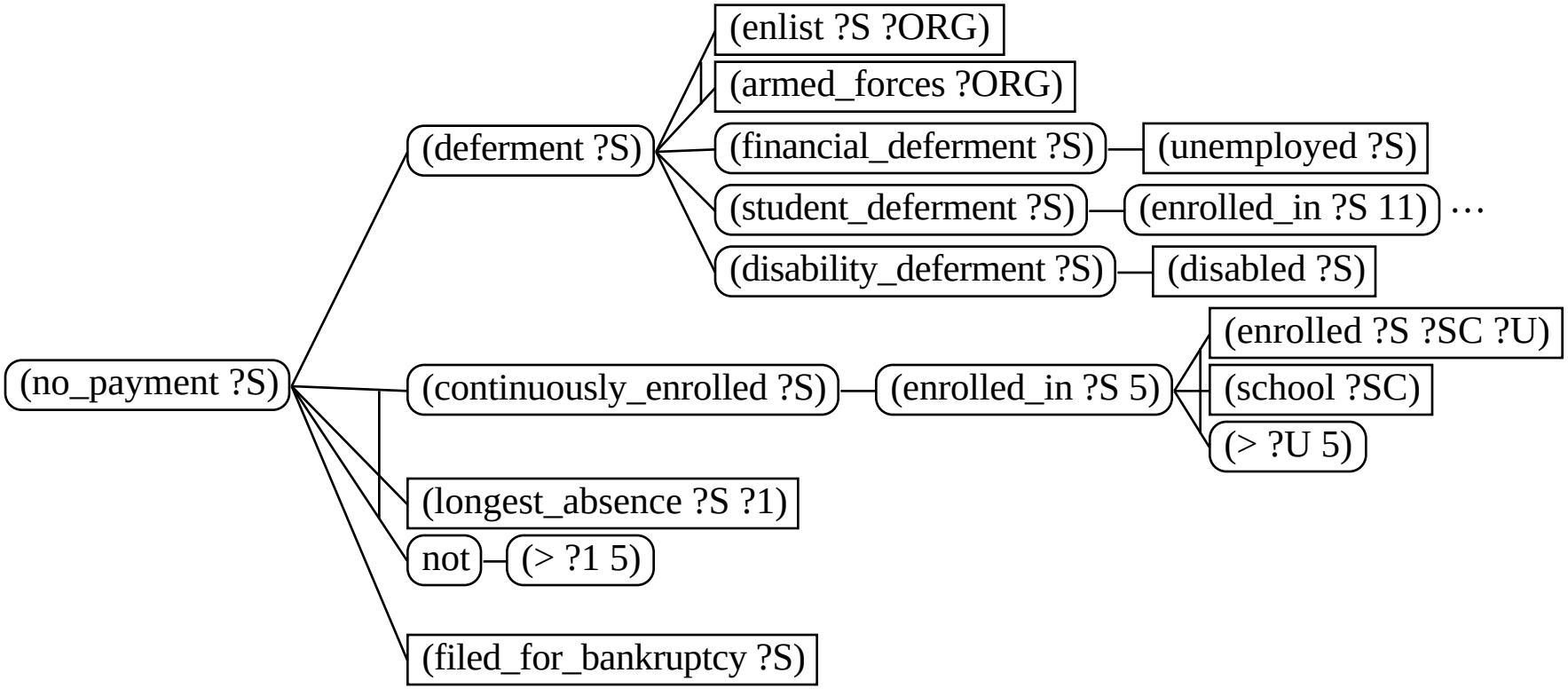


Generalization by literal deletion

If there is a frontier containing a conjunction of literals $p_1 \ldots p_{i-1}\ p_i\ p_{i+1} \ldots p_n$, then *n* frontiers replacing this conjunction with $p_1 \ldots p_{i-1}\ p_{i+1} \ldots p_n$ are evaluated.

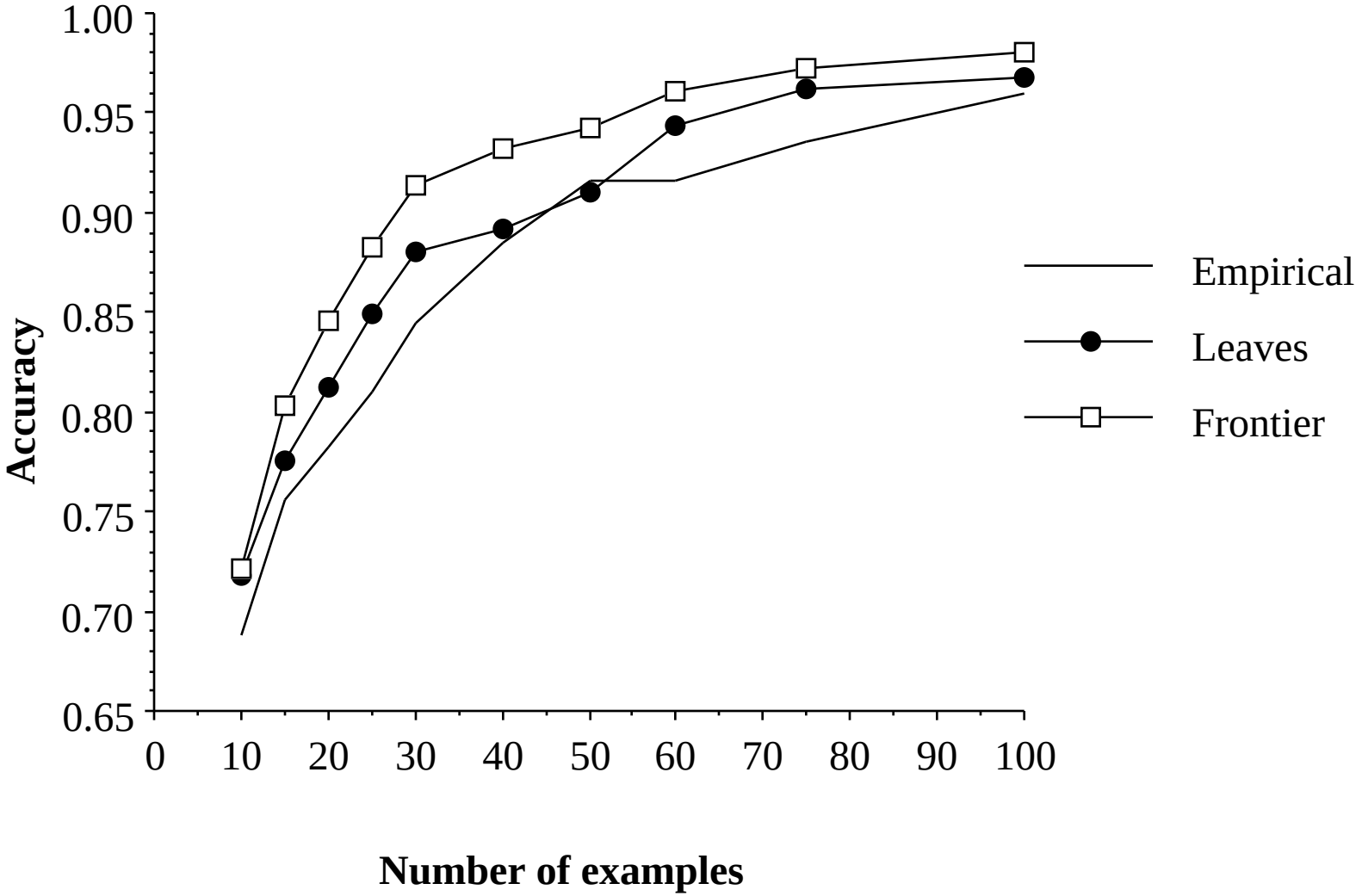


A Sample Learned Concept

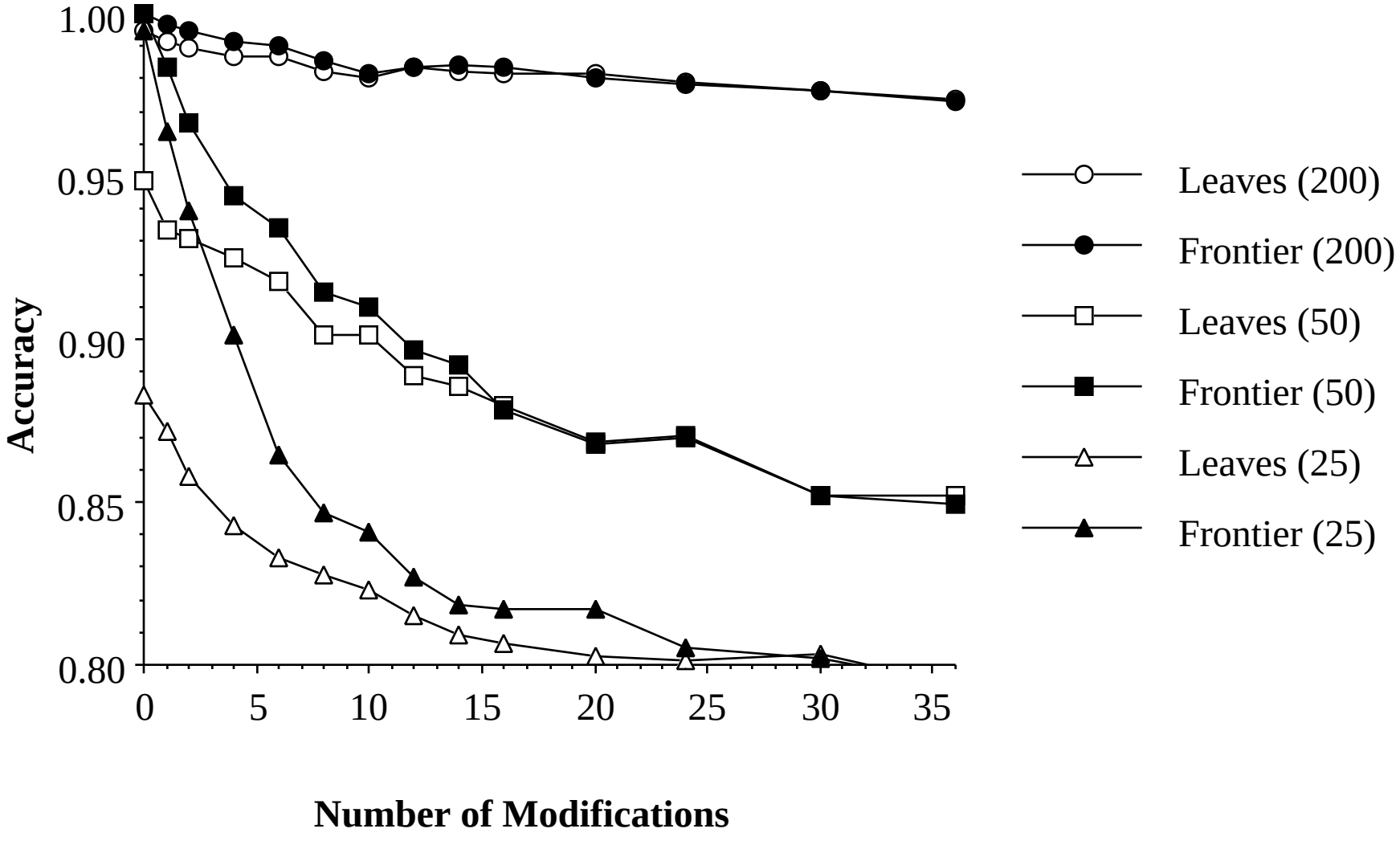


Frontier vs. Operationalization on

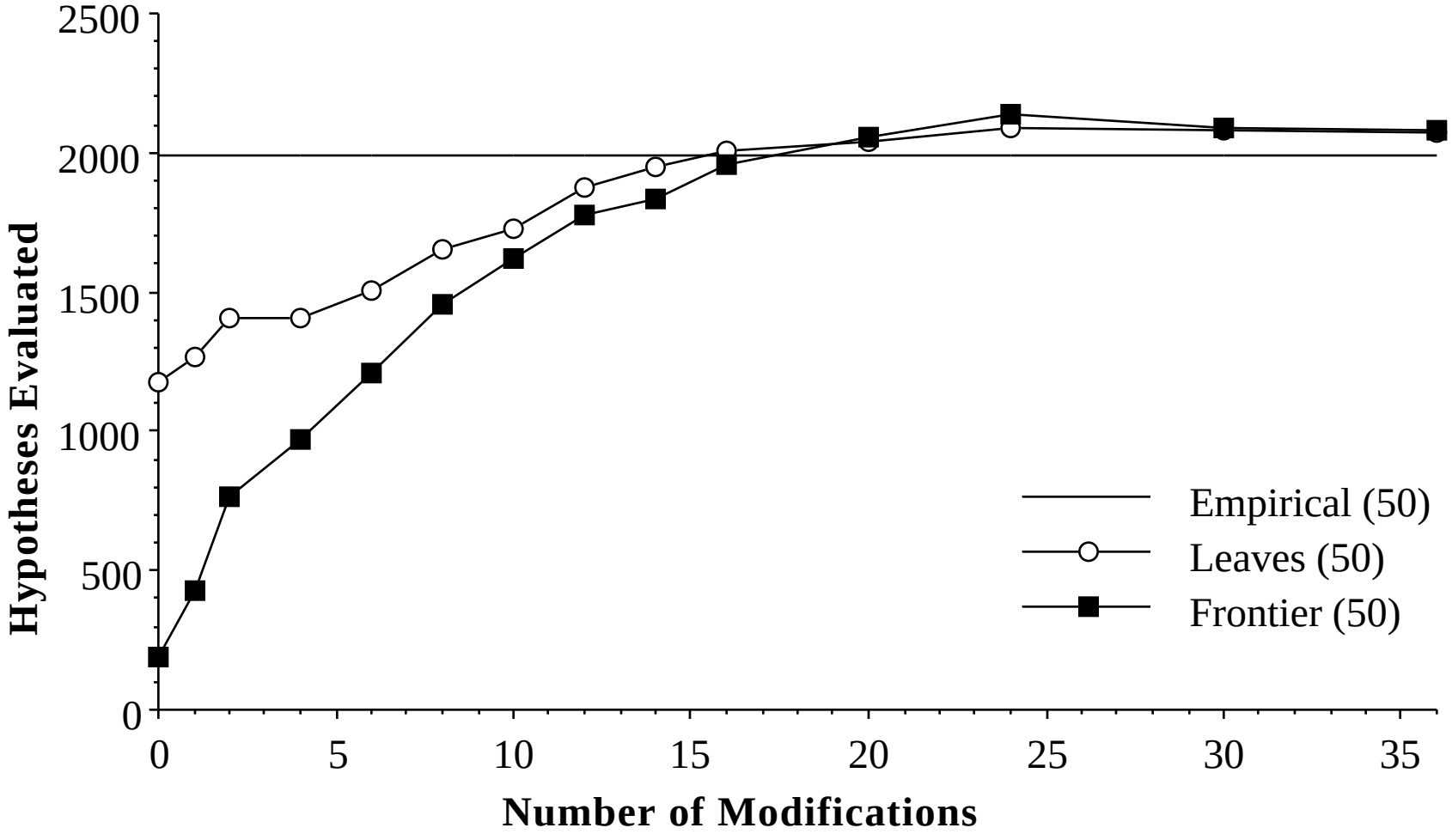
`no_payment_due`



Frontier vs. Operationalization on `KRK-illegal`



Amount of work performed on `KRK-illegal` (50 examples)



NYNEX MAX

MAX- MAintenance eXpert used by NYNEX to determine location of malfunction for customer reported telephone troubles

Input- type of equipment, location of customer, various voltages and resistances

Output- location to which a repairman should be dispatched (central office, customer's equipment, cable facilities, customer's wiring)

MAX is used at over 65 sites

Revisions to the rule-base are necessary to meet local conditions in NYNEX Max or to reflect changes in operation since the development of the expert system.

Tested FOCL's ability to revise MAX

FOCL Test Procedure

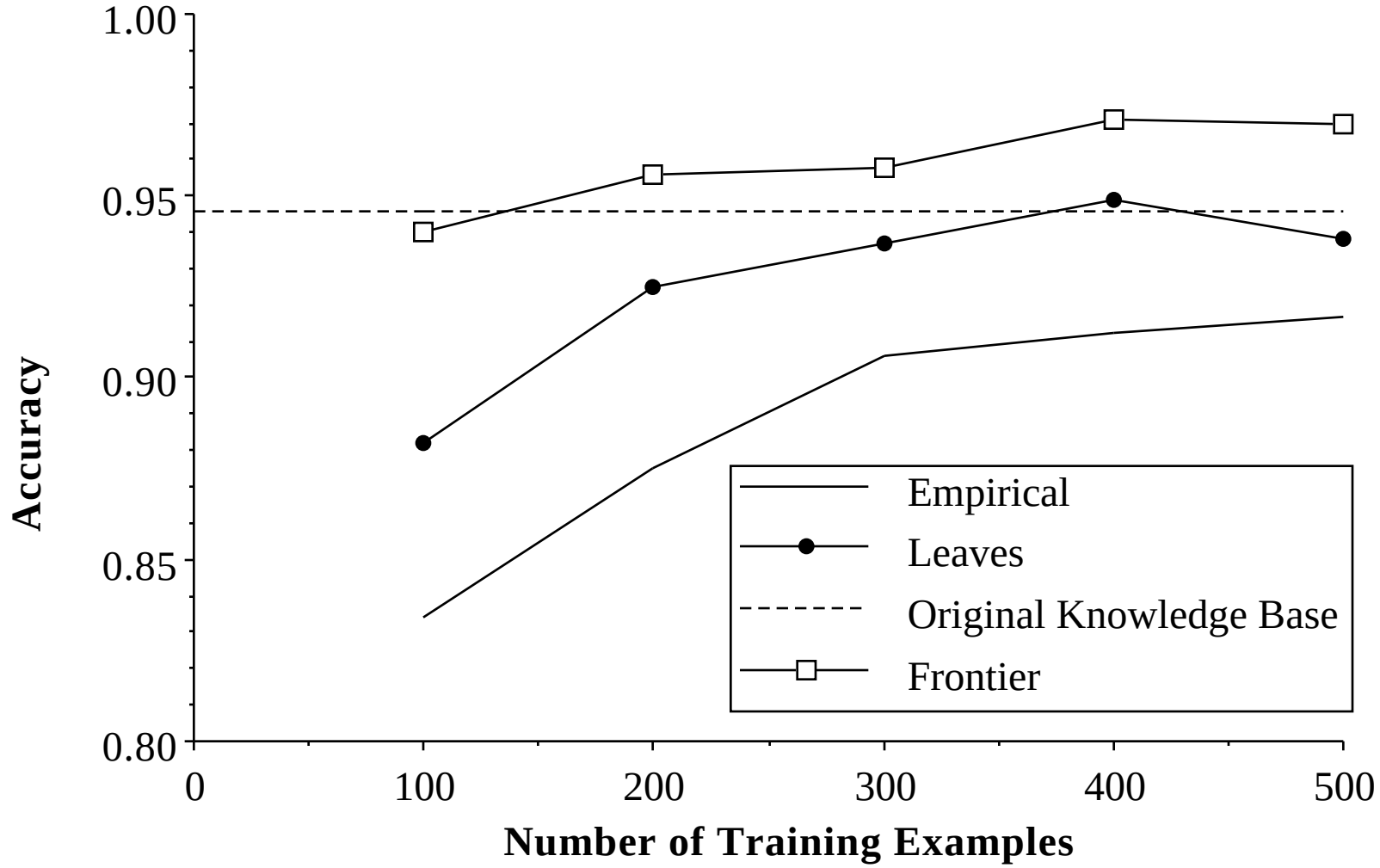
Experiment- Compare Induction to **FOCL** and **FOCL-**FRONTIER

Given:
   Max Data from Varick
   Knowledge base for East Bronx

Produce: Rules to classify data

Results (averaged over 20 trials)
Accuracy measured on independent test set (200 examples)

FOCL Results on Max Data



Current Work

1. Investigating the use of special purpose operators for deriving frontier
   - Replace one predicate by a “similar” one: `senior-citizen -> retired`
   - Reorder arguments of predicate: `between(A,C,B) -> between(A,C,B)`
   - Negate literals: `between(A,C,B) -> not(between(A,C,B))`
   - Changing a numeric threshold `X > 7 -> X > 9`

2. A tighter integration with induction.
   Adding inductive literals at any node in a frontier

3. Dealing with noise directly in the analytic learner

Conclusions

1. Frontiers bias an analytic learner to use abstract parts of domain knowledge (unless specialized parts are more accurate).
   - Frontier- Remove if it is worse
   - Operationalization: Retain if it is better

2. Procedure for deriving useful frontiers of a domain theory

3. Experimentation on commonly used problem shows an improvement in accuracy over operationalization.

4. Experimentation on customizing an operational knowledge-based system yielded promising results.