Reliable and Interpretable Artificial Intelligence
Exercise 2

Dana Drachsler-Cohen

ETH Zurich

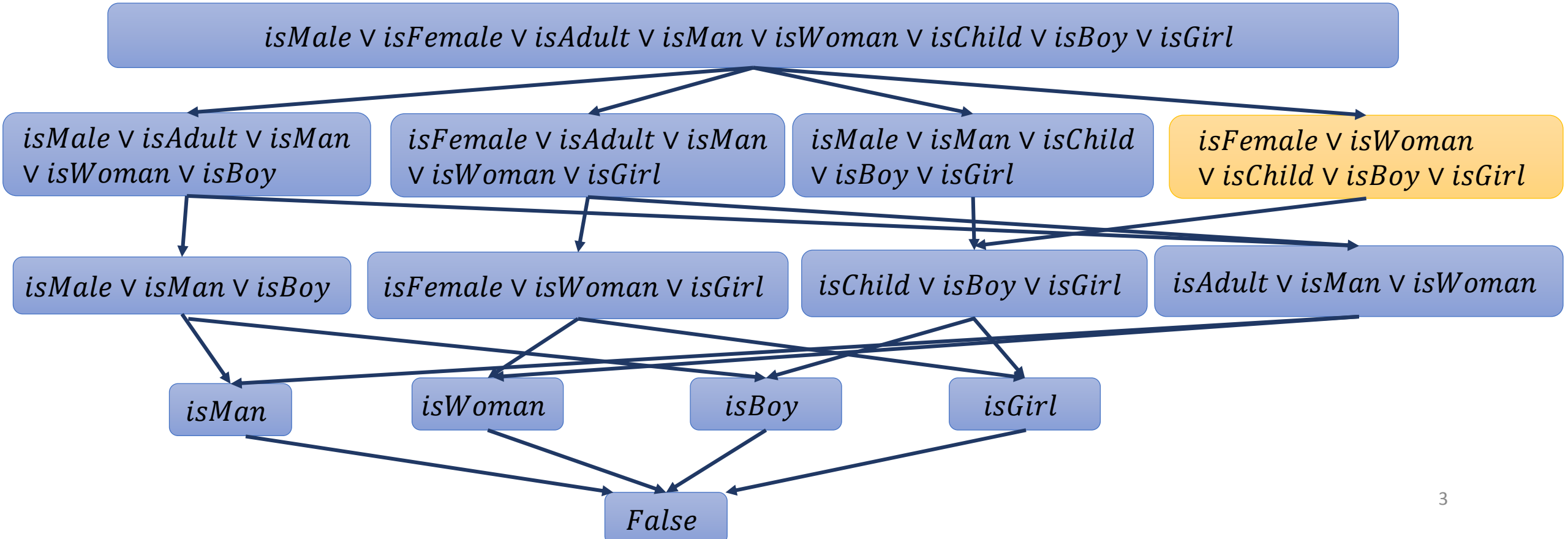
Fall 2017

Exact Programming by Example

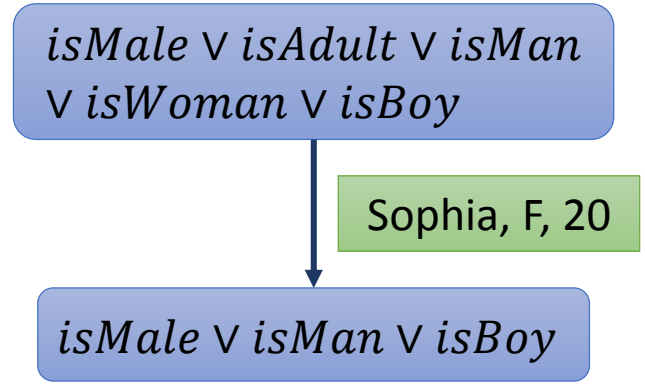
- Programming by examples is the task of synthesizing a program from a set of input-output examples
  - The hope is that the examples are representative enough for the heuristic to synthesize the correct program
- Exact programming by examples is the two-step task:
  1. Learning a mathematical formula capturing the user intent on all inputs
  2. Synthesizing a program meeting the learned specification
- The learning step is phrased in the setting of exact learning
  - Reason about the number of queries compared to the optimum/lower bound

Learning Disjunctive Specifications

- We defined a hypothesis space in which nodes are representative of the equivalence classes and edges link implied nodes



The D-SPEX Algorithm



- We defined the notion of a witness, based on which showed D-SPEX

D-SPEX(φ,T): // initially, call with D-SPEX($\vee_{q \in Q} q, \emptyset$)

$Q \leftarrow Q(\varphi)$;

$Flag \leftarrow 1$

For $\varphi' \in children(\varphi)$:

If $(\forall R \in T. Q(\varphi') \not\subset R)$

e $\leftarrow$witness(φ, φ')

If $(mem(e) = 0)$

$Q = Q \cap Q(\varphi'); Flag \leftarrow 0$

Else $T \leftarrow T \cup \{Q(\varphi')\}$

If $(Flag = 1)$ return $\vee_{q \in Q} q$

D-SPEX($\vee_{q \in Q} q, T$)

Complexity Analysis and Lower Bound

Theorem: If the children and witnesses of any $\varphi \in N$ can be found in time t, then D-SPEX learns ψ in time $t \cdot |Q|$ and $|Q| \cdot \max_{n \in N}(children(n))$ membership queries.

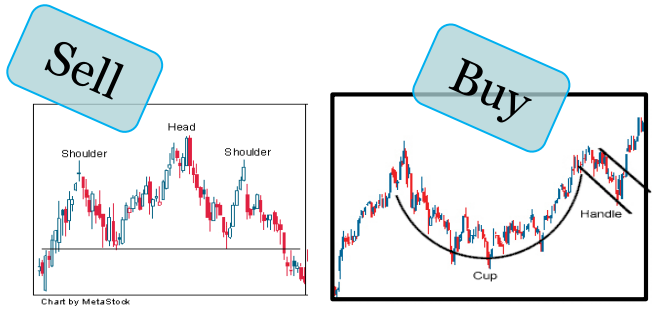
Theorem: any learning algorithm that learns Q_V must pose at least $\max(\log(|N|), \max_{n \in N}|Children(n)|)$.

In particular, D-SPEX poses at most $|Q| \cdot OPT(Q_V)$.

Today: An End-to-End Exact PBE

Technical Analysis: Predict price direction using current prices

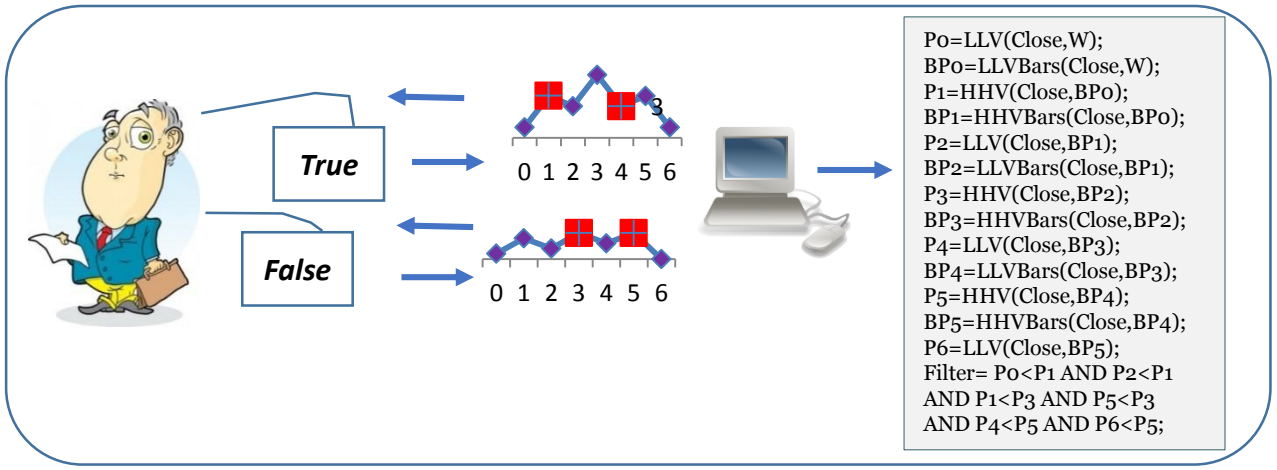
Patterns: Special forms that signal whether to buy or sell



Goal: Synthesize a program (a model) detecting a pattern

```
P0=LLV(Close,W);
BP0=LLVBars(Close,W);
P1=HHV(Close,BP0);
BP1=HHVBars(Close,BP0);
P2=LLV(Close,BP1);
BP2=LLVBars(Close,BP1);
P3=HHV(Close,BP2);
BP3=HHVBars(Close,BP2);
P4=LLV(Close,BP3);
BP4=LLVBars(Close,BP3);
P5=HHV(Close,BP4);
BP5=HHVBars(Close,BP4);
P6=LLV(Close,BP5);
Filter= P0<P1 AND P2<P1 AND P1<P3 AND
P5<P3 AND P4<P5 AND P6<P5;
```

Idea: *Learn* the *exact* pattern from charts



Challenges:
Which questions to ask?
How to reduce their number?

Solution:
Algorithm that will ask at most $|F| \cdot OPT(F_V)$ membership queries where F is the set of possible features, $OPT(F_V)$ is the minimum worst case number of membership queries for F_V

Time-Series Patterns from Charts

Goal: an exact PBE that learns patterns in time-series charts

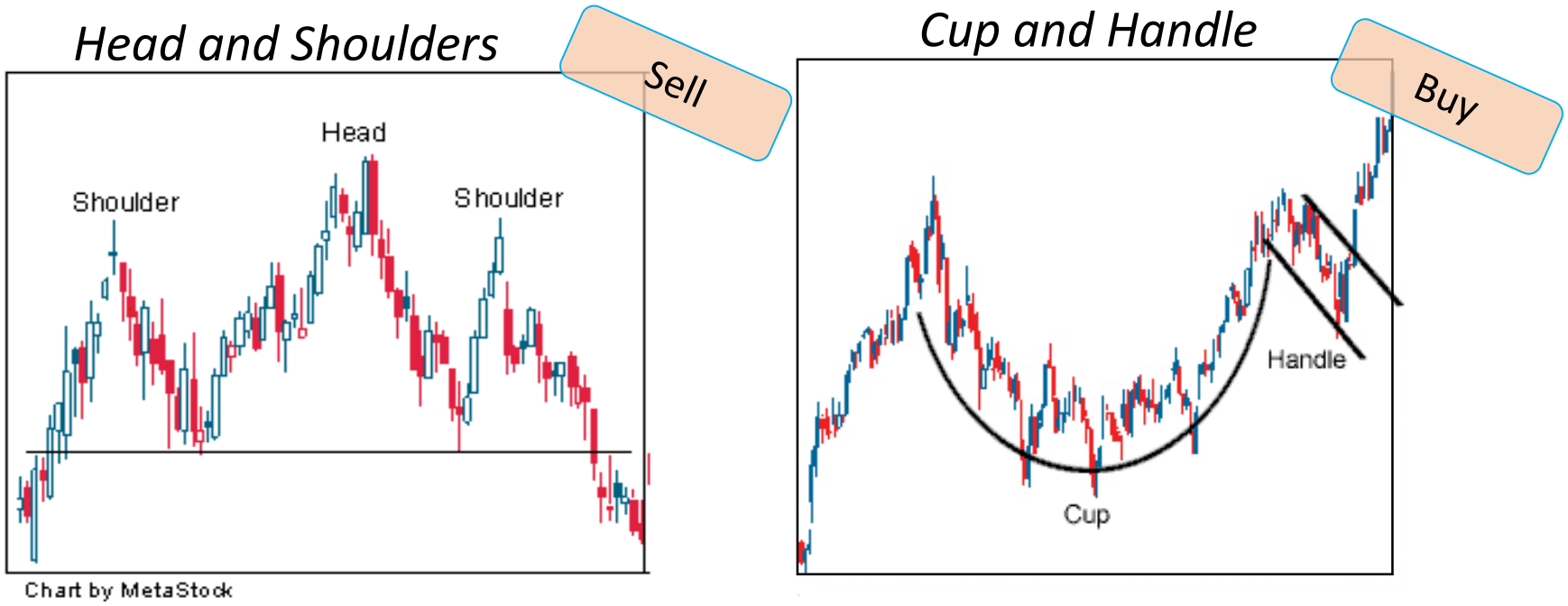
Time-series charts are used in many domains including financial analysis, medicine, and seismology.

Time-Series Patterns from Charts

Experts use these charts to predict important events (e.g., trend changes in a stock price) indicated by special patterns

Time-Series Patterns from Charts

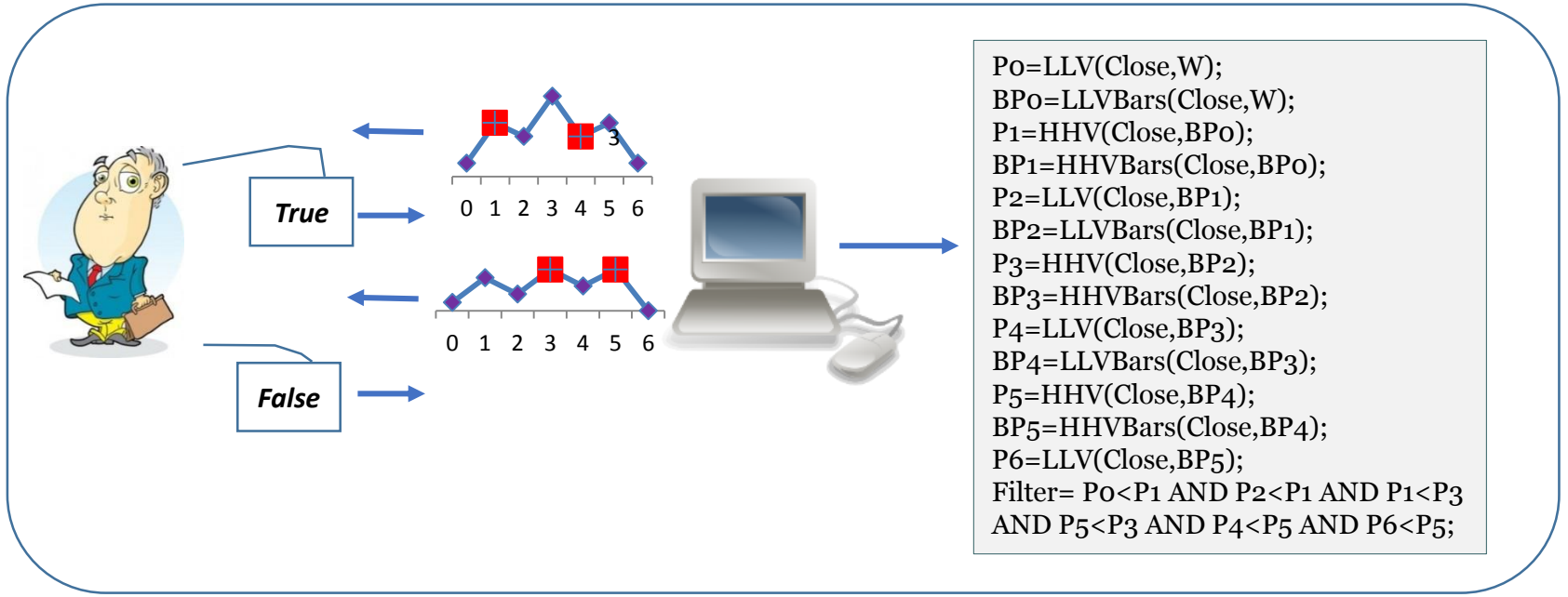
There is a lot of study on common patterns and there are many softwares that enable these experts to write a program that alerts upon detecting their customized pattern



Time-Series Patterns from Charts

Unfortunately, writing programs is a complex task for these experts, who are not programmers

Goal: learn the specifications from chart examples, then synthesize a program



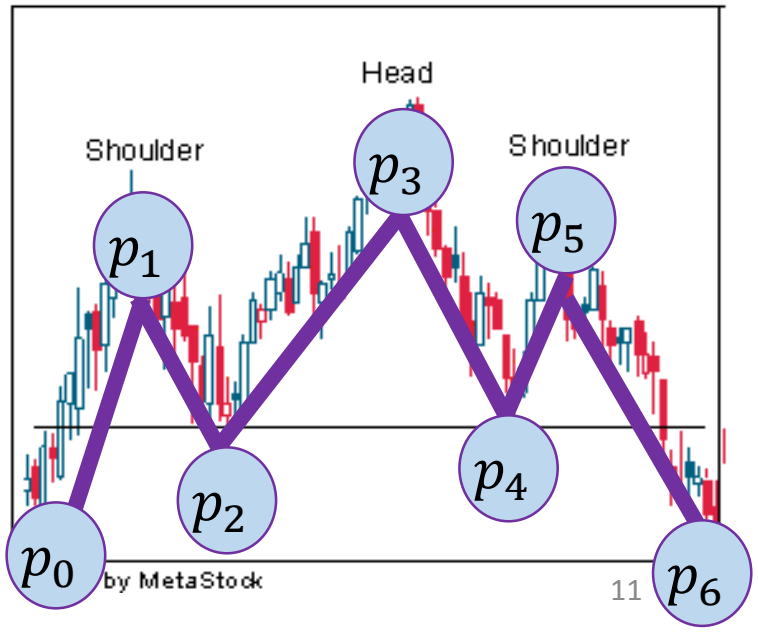
Time-series Patterns

A chart is a function over time

p_i is the price at time point i

A pattern is a conjunction over $Q = \{p_i < p_j | i \neq j\}$

$$\varphi_{HS} = (p_0 < p_2) \wedge (p_2 < p_1) \wedge (p_1 < p_3) \wedge (p_2 < p_4) \wedge (p_4 < p_5) \wedge (p_6 < p_5) \wedge (p_5 < p_3) \wedge (p_6 < p_0)$$

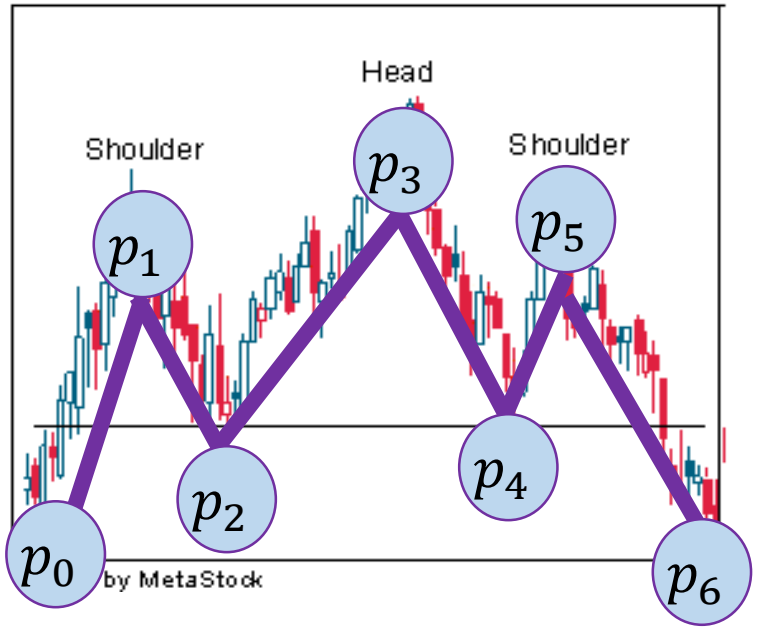


Exact PBE for Time-series Patterns

- We will assume a slightly different setting
- The learning begins from *the user* who provides an initial chart example e
- Then, the set of predicates is

$Q_e = \{p_i < p_j | e \vDash p_i < p_j\}$

- Target formula is in $Q_{e,\wedge} = \{\bigwedge_{q \in \mathrm{P}} q \,| P \subseteq Q_e\}$
- Note that $Q_{e,\wedge}$ cannot contain cyclic constraints
  - $p_i < p_j \in Q_{e,\wedge} \Rightarrow p_j < p_i \notin Q_{e,\wedge}$



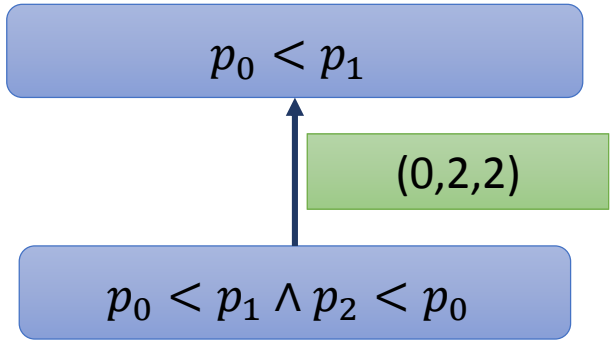
HW: Exact PBE for Time-series Patterns

1. Define C-SPEX, a variation of D-SPEX for learning conjunctions
   - Hint: What is the hypothesis space? How do the lemma change?
2. Let e be a chart example and $Q_e = \{p_i < p_j \mid e \vDash p_i < p_j\}$
   a) Define how to compute the children of a node. What is the time complexity?
   b) Define how to find the witnesses. What is the time complexity?
      - Hint: topological sorting
   c) Determine how many membership queries C-SPEX can present.
      - Hint: we have a much better bound…

The Hypothesis Space

- $N = \{\bigwedge_{q \in A} q \mid \forall q' \in Q_e \setminus A : (\bigwedge_{q \in A} q \wedge q') \not\equiv \bigwedge_{q \in A} q\}$
- $E = \{(\varphi_1, \varphi_2) \mid \varphi_1 \vDash \varphi_2 \wedge \forall \varphi_3 (\varphi_1 \vDash \varphi_3 \wedge \varphi_3 \neq \varphi_2 \rightarrow \varphi_3 \nvDash \varphi_2)\}$
- Lemma 1: If φ_1 is a descendant of φ_2, then $Q(\varphi_1) \subset Q(\varphi_2)$
- Lemma 2: $Q(\gcd(\varphi_1, \varphi_2)) = Q(\varphi_1) \cap Q(\varphi_2)$
  - In particular, $\wedge \left(Q(\varphi_1) \cap Q(\varphi_2)\right) \in N$

Witnesses



- A **witness** for $\varphi_1, \varphi_2 \in N$ is an example $e \in D$ such that $\varphi_1(e) \neq \varphi_2(e)$
- Lemma 3: Let φ_1 be a child of φ_2 and e a witness for them. Then:
  1. $\varphi_1(e) = 1, \varphi_2(e) = 0$
  2. For every $q \in Q(\varphi_1), q(e) = 1$
  3. For every $q \in Q(\varphi_2) \setminus Q(\varphi_1), q(e) = 0$
- Lemma 4: Let $\{\varphi_1, \dots, \varphi_k\}$ be the children of φ. If e_i is a witness for φ_i and φ, then e_i is not a witness for φ, φ_j for any $j \neq i$.
- Lemma 5: Let φ_i be a child of φ, e_i a witness, φ_j a descendant of φ
  1. If $\varphi_j(e_i) = 1$, φ_j is a descendant of φ_i or equal to φ_i
  2. If $\varphi_j(e_i) = 0$, φ_j is *not* a descendant of φ_i nor equal to φ_i

The C-SPEX Algorithm

C-SPEX(φ,T): // initially, call with C-SPEX($\wedge_{q \in Q}\, q, \emptyset$)

$Q \leftarrow Q(\varphi)$;

$Flag \leftarrow 1$

For $\varphi' \in children(\varphi)$:

If $(\forall R \in T. Q(\varphi') \not\subset R)$

e ←witness(φ, φ')

If $(mem(e) = 1)$

$Q = Q \cap Q(\varphi'); Flag \leftarrow 0$

Else $T \leftarrow T \cup \{Q(\varphi')\}$

If $(Flag = 1)$ return $\wedge_{q \in Q}\, q$

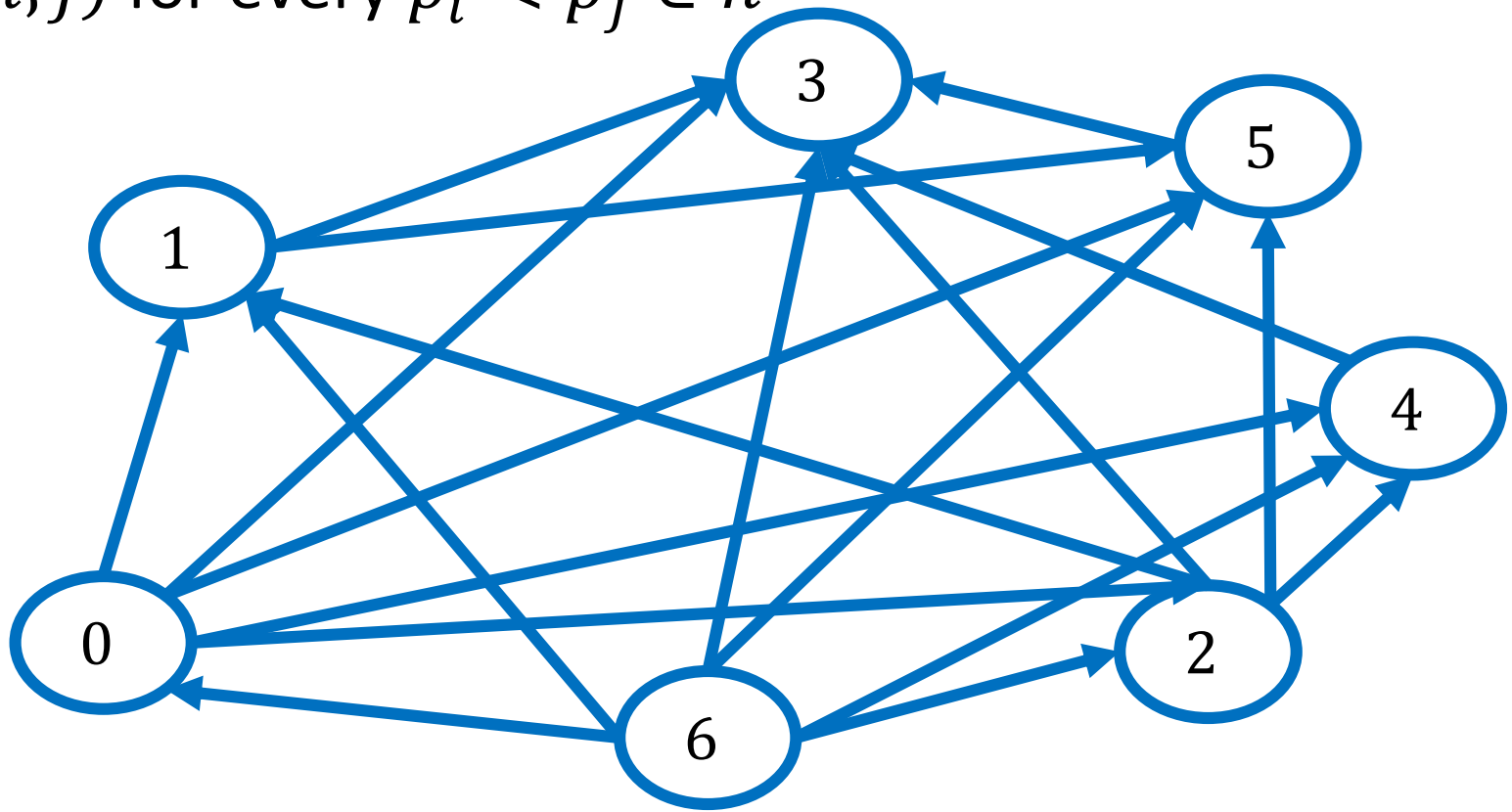
C-SPEX($\wedge_{q \in Q}\, q, T$)

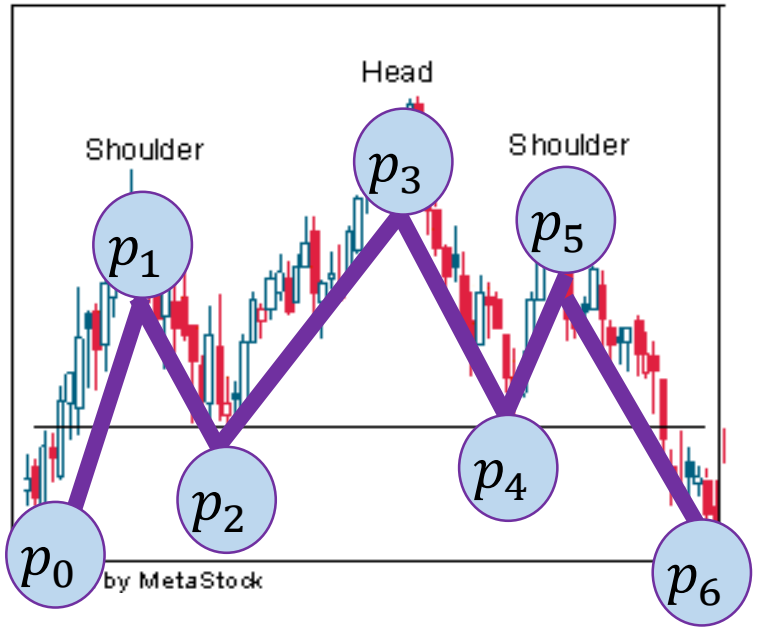
HW: Exact PBE for Time-series Patterns

1. Define C-SPEX, a variation of D-SPEX for learning conjunctions
   - Hint: What is the hypothesis space? How do the lemma change?
2. Let e be a chart example and $Q_e = \{p_i < p_j \mid e \vDash p_i < p_j\}$
   a) Define how to compute the children of a node. What is the time complexity?
      - Hint: represent the constraints in a graph whose nodes are p_i
   b) Define how to find the witnesses. What is the time complexity?
      - Hint: topological sorting
   c) Determine how many membership queries C-SPEX can present.
      - Hint: we have a much better bound…

Representing Nodes as Constraint Graphs

Given a node n, we define a graph G_n capturing n's constraints: nodes are points, there is an edge (i, j) for every $p_i < p_j \in n$





$(p_0 < p_2) \wedge (p_2 < p_1) \wedge$
$(p_1 < p_3) \wedge (p_2 < p_4)$
$\wedge (p_4 < p_5) \wedge (p_6 < p_5)$
$\wedge (p_5 < p_3) \wedge (p_6 < p_0)$

Computing the Node's Children

The children of n are not equivalent to n

That is, they have (at least) one less constraint… That is, one less edge in the graph

Can there be two fewer constraints?

What kind of constraints can be removed to obtain a child?

Lemma: Let n be a node. For every (i, j) in the graph G_n, such that (i, j) are reachable only through this edge, $n \setminus \{p_i < p_j\}$ is a child of n

Computing a Node's Children

Lemma: Let n be a node. For every (i, j) in the graph G_n, such that (i, j) are reachable only through this edge, $n \setminus \{p_i < p_j\}$ is a child of n

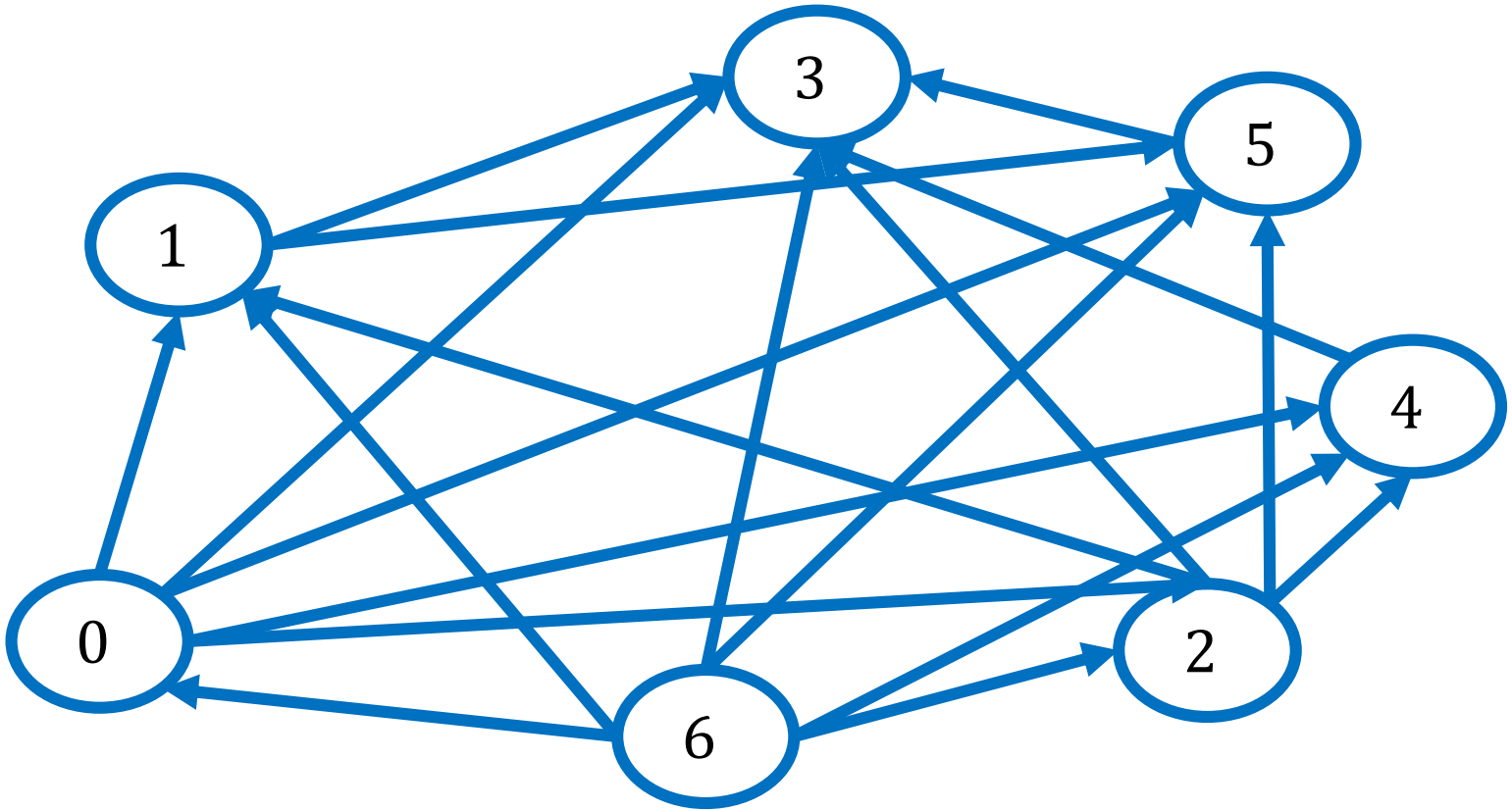
- To find the children, check for every edge (i, j):
  - Is j reachable from i in the graph that does not contain (i, j)
    - E.g., BFS
- Time complexity $O(E \cdot (E + N))$

HW: Exact PBE for Time-series Patterns

1. Define C-SPEX, a variation of D-SPEX for learning conjunctions
   - Hint: What is the hypothesis space? How do the lemma change?
2. Let e be a chart example and $Q_e = \{p_i < p_j \mid e \vDash p_i < p_j\}$
   a) Define how to compute the children of a node. What is the time complexity?
      - Hint: represent the constraints in a graph whose nodes are p_i
   b) Define how to find the witnesses. What is the time complexity?
      - Hint: topological sorting
   c) Determine how many membership queries C-SPEX can present.
      - Hint: we have a much better bound…

Finding a Witness

- When learning conjunctive formulas, a witness is an example satisfying the child but not the parent



Finding a Witness

Let n be a node and $n_{i,j}$ be a child of n.

In the graph $G_{n_{i,j}}$, there is no edge between i, j

Match i, j to a single node, and get a DAG

Use topological sorting to find an assignment to the points

$\Rightarrow$This assignment satisfies $n_{i,j}$ but not n

HW: Exact PBE for Time-series Patterns

1. Define C-SPEX, a variation of D-SPEX for learning conjunctions
   - Hint: What is the hypothesis space? How do the lemma change?
2. Let e be a chart example and $Q_e = \{p_i < p_j \mid e \vDash p_i < p_j\}$
   a) Define how to compute the children of a node. What is the time complexity?
      - Hint: represent the constraints in a graph whose nodes are p_i
   b) Define how to find the witnesses. What is the time complexity?
      - Hint: topological sorting
   c) Determine how many membership queries C-SPEX can present.
      - Hint: we have a much better bound…

#Queries

How many times can a constraint be asked about?

If the classification is positive?

$p_i < p_j$ is not in the target formula, and is thus removed

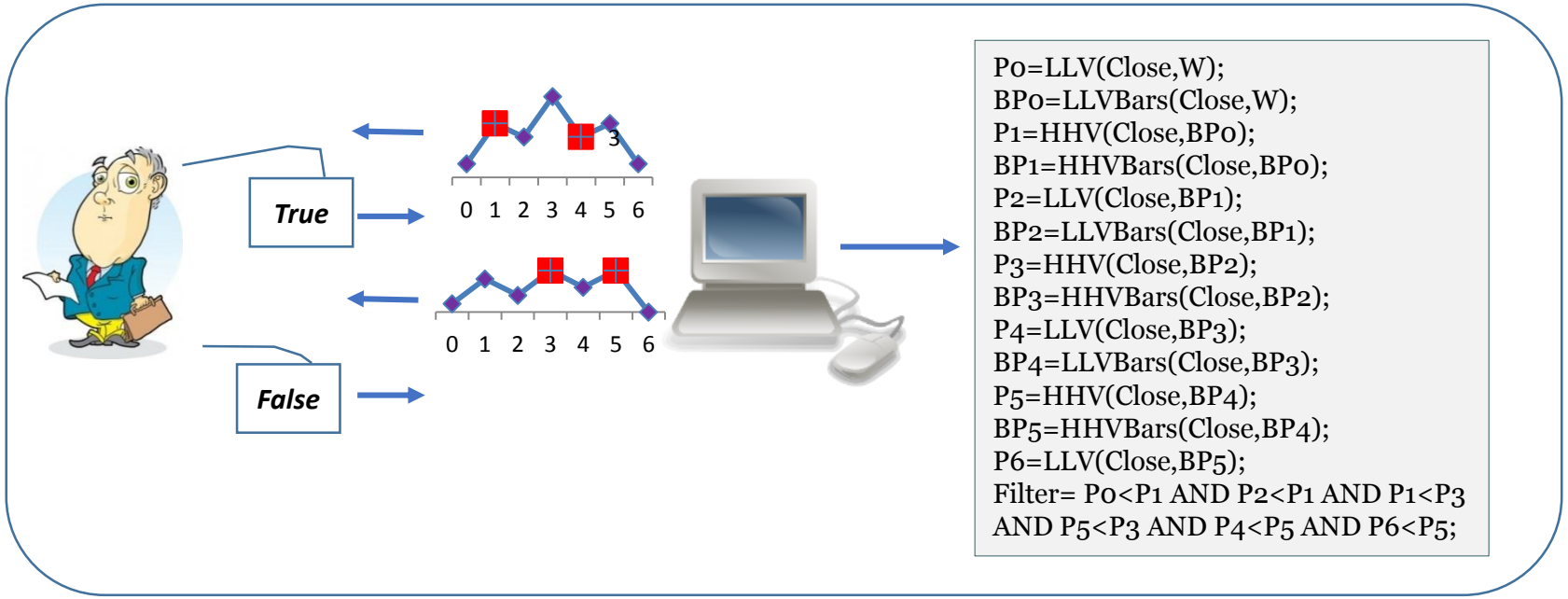
If the classification is negative?

$p_i < p_j$ is in the target formula, and thus C-SPEX proceeds to its descendants, all include this constraint

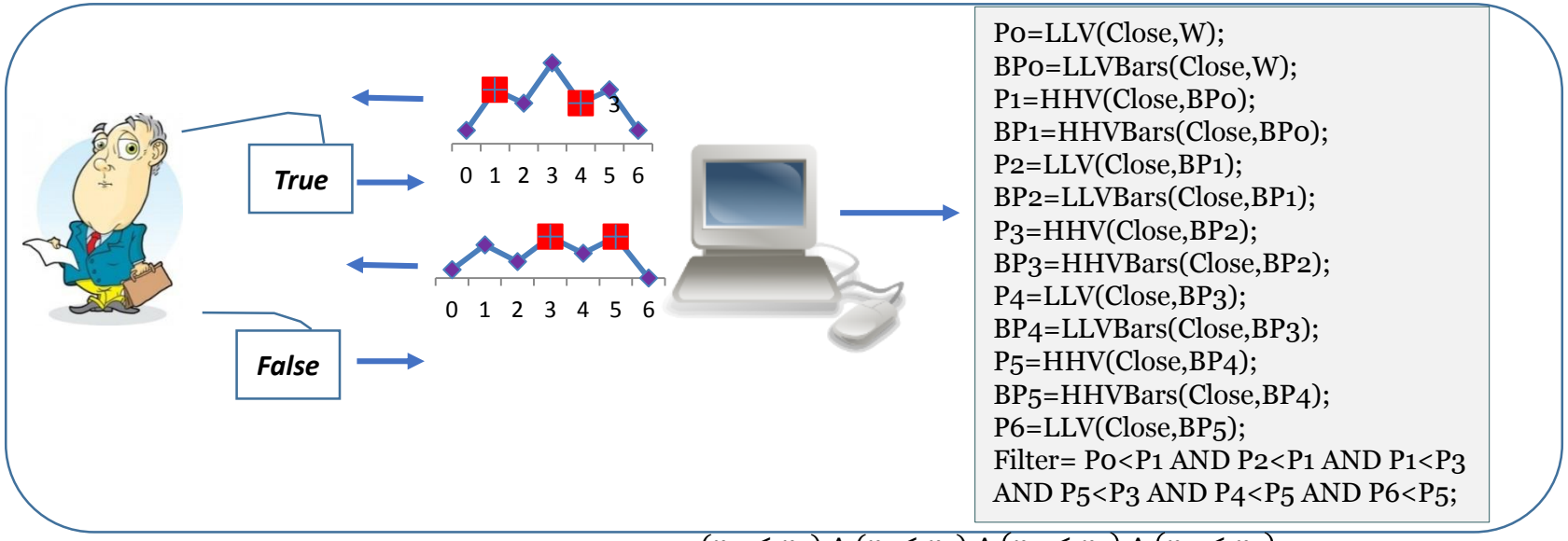
Lemma: Given an initial example e, the class $Q_{e,\wedge}$ is learnable in polynomial time and with at most $|Q_{e,\wedge}|$ membership queries

An End-to-End Exact PBE

- How can we obtain the end-to-end synthesizer?

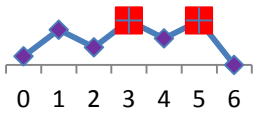


An End-to-End Exact PBE



From the specification infer the minimum & maximum points, and generate the code

The witnesses are assignments $(p_0, \ldots, p_n)$ —display as chart



$(p_0 < p_2) \wedge (p_2 < p_1) \wedge (p_1 < p_3) \wedge (p_2 < p_4)$
$\wedge (p_4 < p_5) \wedge (p_6 < p_5) \wedge (p_5 < p_3) \wedge (p_6 < p_0)$