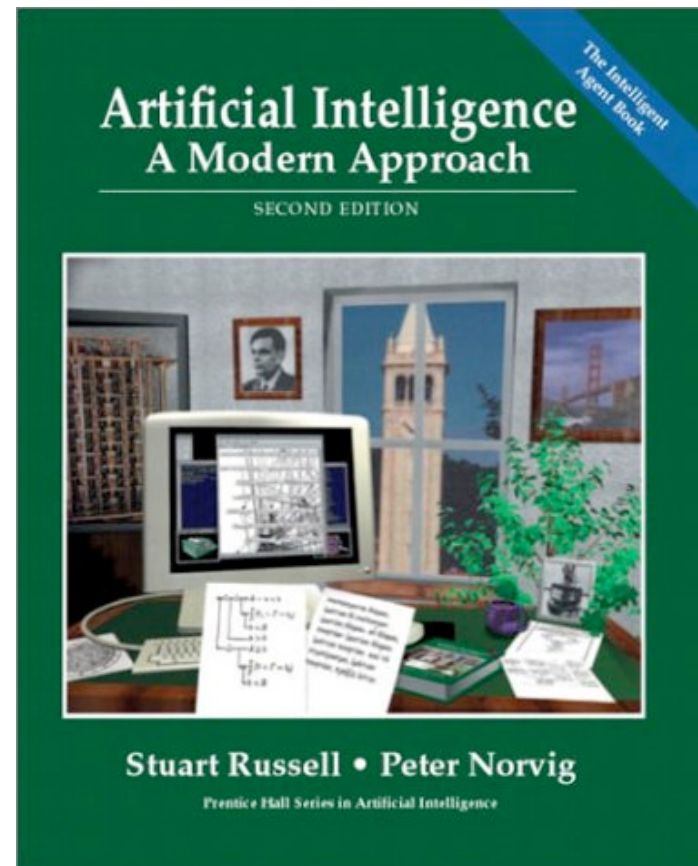


# Inductive Learning (continued)

## Chapter 18

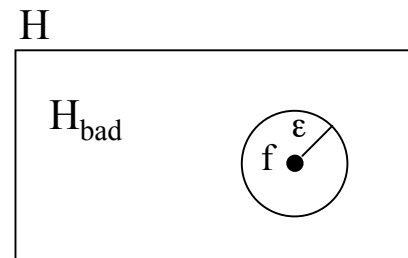
Slides by **Yun Peng**  
Some material adopted from notes  
by Chuck Dyer



# Computational learning theory

- Intersection of AI, statistics, and computational theory
- Probably approximately correct (PAC) learning:
  - Seriously wrong hypotheses can be found out almost certainly (with high probability) using a small number of examples
  - Any hypothesis that is consistent with a significantly large set of training examples is unlikely to be seriously wrong: it is **probably approximately correct**.
- How many examples are needed?
  - Sample complexity (# of examples to “guarantee” correctness) grows with the size of the model space
  - **Stationarity** assumption: Training set and test sets are drawn from the same distribution (i.e., the distribution does not change over time)

- Notations:
  - $X$ : set of all possible examples
  - $D$ : distribution from which examples are drawn
  - $H$ : set of all possible hypotheses
  - $N$ : the number of examples in the training set
  - $f$ : the true function to be learned
- **Approximately** correct:
  - $\text{error}(h) = P(h(x) \neq f(x) \mid x \text{ drawn from } D)$
  - Hypothesis  $h$  is approximately correct if  $\text{error}(h) \leq \epsilon$  where  $\epsilon$  is a given threshold
  - Approximately correct hypotheses lie inside the  $\epsilon$ -ball around  $f$



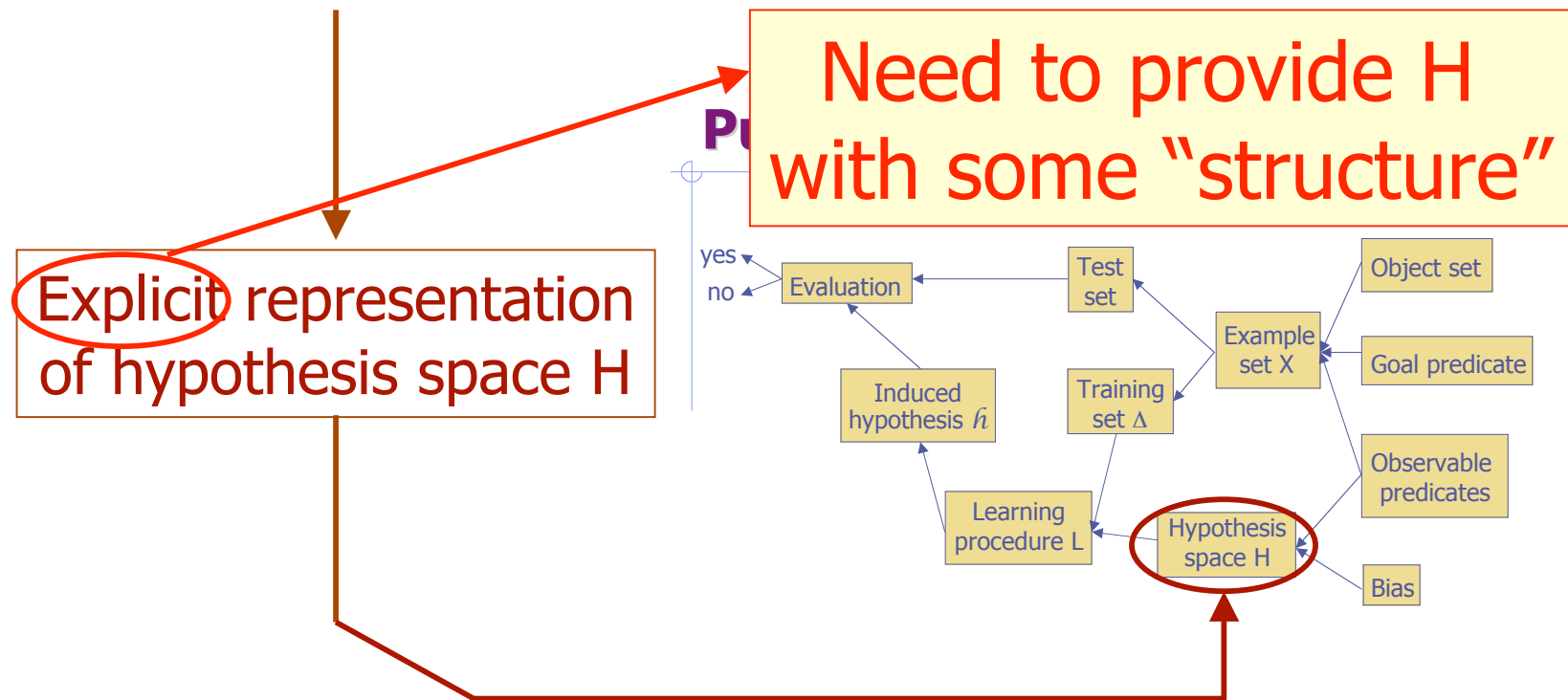
- **Probably Approximately correct** hypothesis  $h$ :
  - If the probability of  $\text{error}(h) \leq \epsilon$  is greater than or equal to a given threshold  $1 - \delta$
  - A loose upper bound on the number of examples needed to guarantee PAC:
    - We know that  $\text{error}(h_b) > \epsilon$
    - $P(h_b \text{ agrees with } N \text{ examples}) \leq (1 - \epsilon)^N$
    - $P(\mathbf{H}_{\text{bad}} \text{ contains cons. hyp.}) \leq |\mathbf{H}_{\text{bad}}| (1 - \epsilon)^N \leq |\mathbf{H}| (1 - \epsilon)^N < \delta$
    - with  $(1 - \epsilon)^N \leq e^{-\epsilon N}$
  - The more accurate you want (with smaller  $\epsilon$ ), and the more certain you want (with smaller  $\delta$ ), the more examples you need!
 
$$N \geq \frac{1}{\epsilon} (\ln \frac{1}{\delta} + \ln |\mathbf{H}|)$$
- Theoretical results apply to fairly simple learning models e.g., decision list learning. A decision list is a logical expression of a restricted form.

# Version spaces

- READING: Russell & Norvig, 19.1; Mitchell, *Machine Learning*, Chapter 2 (through section 2.5 required; 2.6-2.8 optional)
- Hypotheses are represented by a set of logical sentences.
- Incremental construction of hypothesis.
- Prior “domain” knowledge can be included/used.
- Enables using the full power of logical inference.

# Predicate-Learning Methods

- Decision tree
- Version space



# Version Spaces

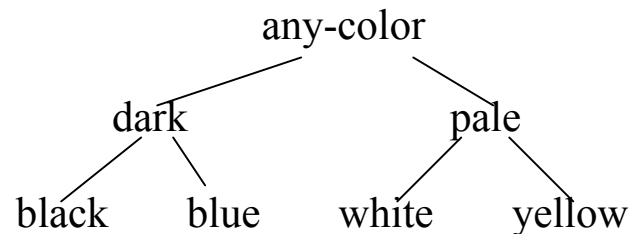
- The “version space” is the set of all hypotheses that are consistent with the training instances processed so far.
- An algorithm:
  - $V := H$  ;; the version space  $V$  is ALL hypotheses  $H$
  - For each example  $e$ :
    - Eliminate any member of  $V$  that disagrees with  $e$
    - If  $V$  is empty, FAIL
  - Return  $V$  as the set of consistent hypotheses

# Version Spaces: The Problem

- PROBLEM:  $V$  is huge!!
- Suppose you have  $N$  attributes, each with  $k$  possible values
- Suppose you allow a hypothesis to be any disjunction of instances
- There are  $k^N$  possible instances  $\rightarrow |H| = 2^{k^N}$
- If  $N=5$  and  $k=2$ ,  $|H| = 2^{32}!!$

# Version Spaces: The Tricks

- First Trick: Don't allow arbitrary disjunctions
  - Organize the feature values into a hierarchy of allowed disjunctions, e.g.



- Now there are only 7 “abstract values” instead of 16 disjunctive combinations (e.g., “black of white” isn't allowed)
- Second Trick: Define a partial ordering on H (“general to specific”) and only keep track of the upper bound and lower bound of the version space
- RESULT: An incremental, efficient algorithm!

# Rewarded Card Example

$(r=1) \vee \dots \vee (r=10) \vee (r=J) \vee (r=Q) \vee (r=K) \Leftrightarrow \text{ANY-RANK}(r)$

$(r=1) \vee \dots \vee (r=10) \Leftrightarrow \text{NUM}(r)$

$(r=J) \vee (r=Q) \vee (r=K) \Leftrightarrow \text{FACE}(r)$

$(s=\spadesuit) \vee (s=\clubsuit) \vee (s=\diamondsuit) \vee (s=\heartsuit) \Leftrightarrow \text{ANY-SUIT}(s)$

$(s=\spadesuit) \vee (s=\clubsuit) \Leftrightarrow \text{BLACK}(s)$

$(s=\diamondsuit) \vee (s=\heartsuit) \Leftrightarrow \text{RED}(s)$

A hypothesis is any sentence of the form:

$$R(r) \wedge S(s) \Leftrightarrow \text{IN-CLASS}([r,s])$$

where:

- $R(r)$  is  $\text{ANY-RANK}(r)$ ,  $\text{NUM}(r)$ ,  $\text{FACE}(r)$ , or  $(r=j)$
- $S(s)$  is  $\text{ANY-SUIT}(s)$ ,  $\text{BLACK}(s)$ ,  $\text{RED}(s)$ , or  $(s=k)$

# Simplified Representation

For simplicity, we represent a concept by  $rs$ , with:

- $r \in \{a, n, f, 1, \dots, 10, j, q, k\}$
- $s \in \{a, b, r, \clubsuit, \spadesuit, \diamondsuit, \heartsuit\}$

For example:

- $n\spadesuit$  represents:  
 $\text{NUM}(r) \wedge (s=\spadesuit) \Leftrightarrow \text{IN-CLASS}([r,s])$
- $aa$  represents:  
 $\text{ANY-RANK}(r) \wedge \text{ANY-SUIT}(s) \Leftrightarrow \text{IN-CLASS}([r,s])$

# Extension of a Hypothesis

The **extension** of a hypothesis  $h$  is the set of objects that satisfies  $h$

Examples:

- The extension of  $f♠$  is:  $\{j♠, q♠, k♠\}$
- The extension of  $aa$  is the set of all cards

# More General/Specific Relation

- Let  $h_1$  and  $h_2$  be two hypotheses in  $H$
- $h_1$  is **more general** than  $h_2$  iff the extension of  $h_1$  is a proper superset of the extension of  $h_2$

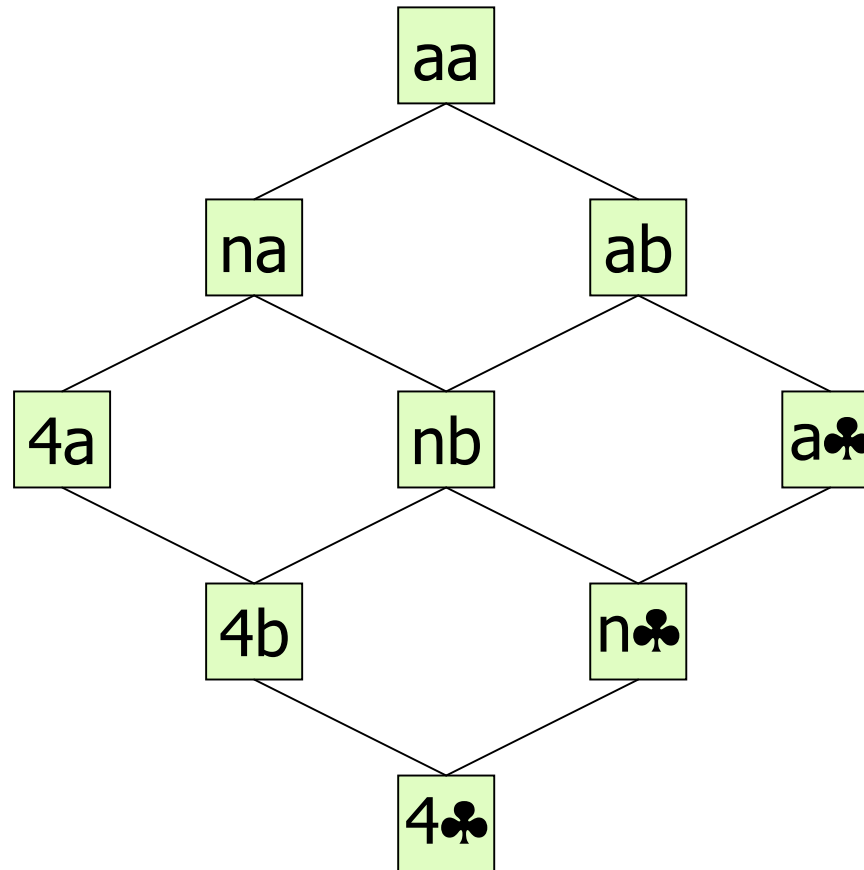
## Examples:

- aa is more general than f♦
- f♥ is more general than q♥
- fr and nr are not comparable

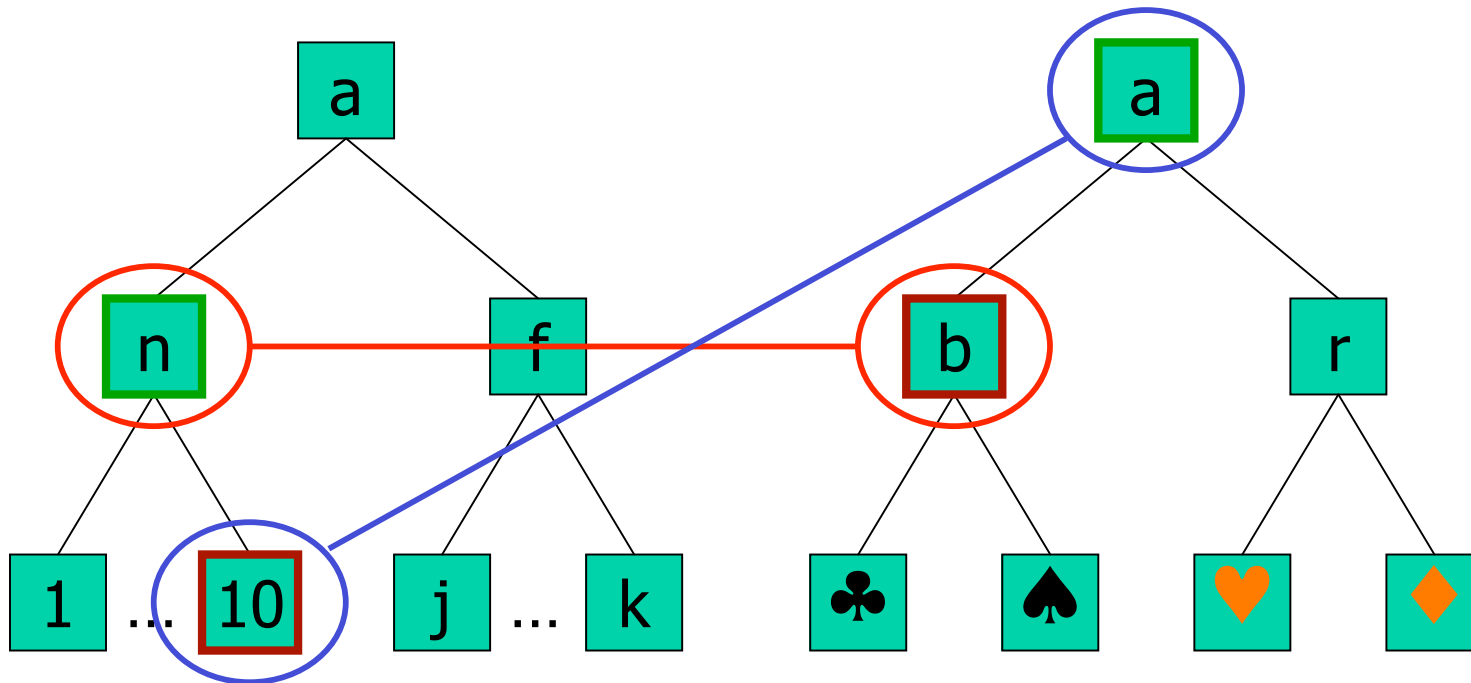
# More General/Specific Relation

- Let  $h_1$  and  $h_2$  be two hypotheses in H
- $h_1$  is **more general** than  $h_2$  iff the extension of  $h_1$  is a proper superset of the extension of  $h_2$
- The inverse of the “more general” relation is the “**more specific**” relation
- The “more general” relation defines a **partial ordering** on the hypotheses in H

# Example: Subset of Partial Order



# Construction of Ordering Relation

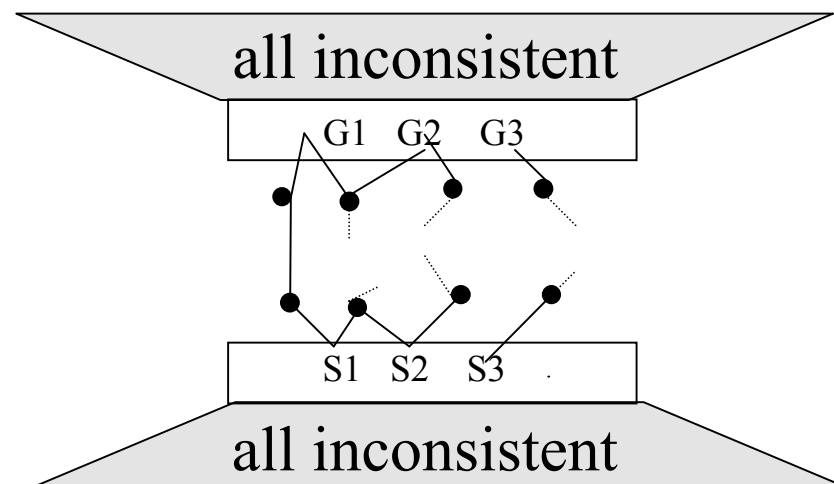


# G-Boundary / S-Boundary of $V$

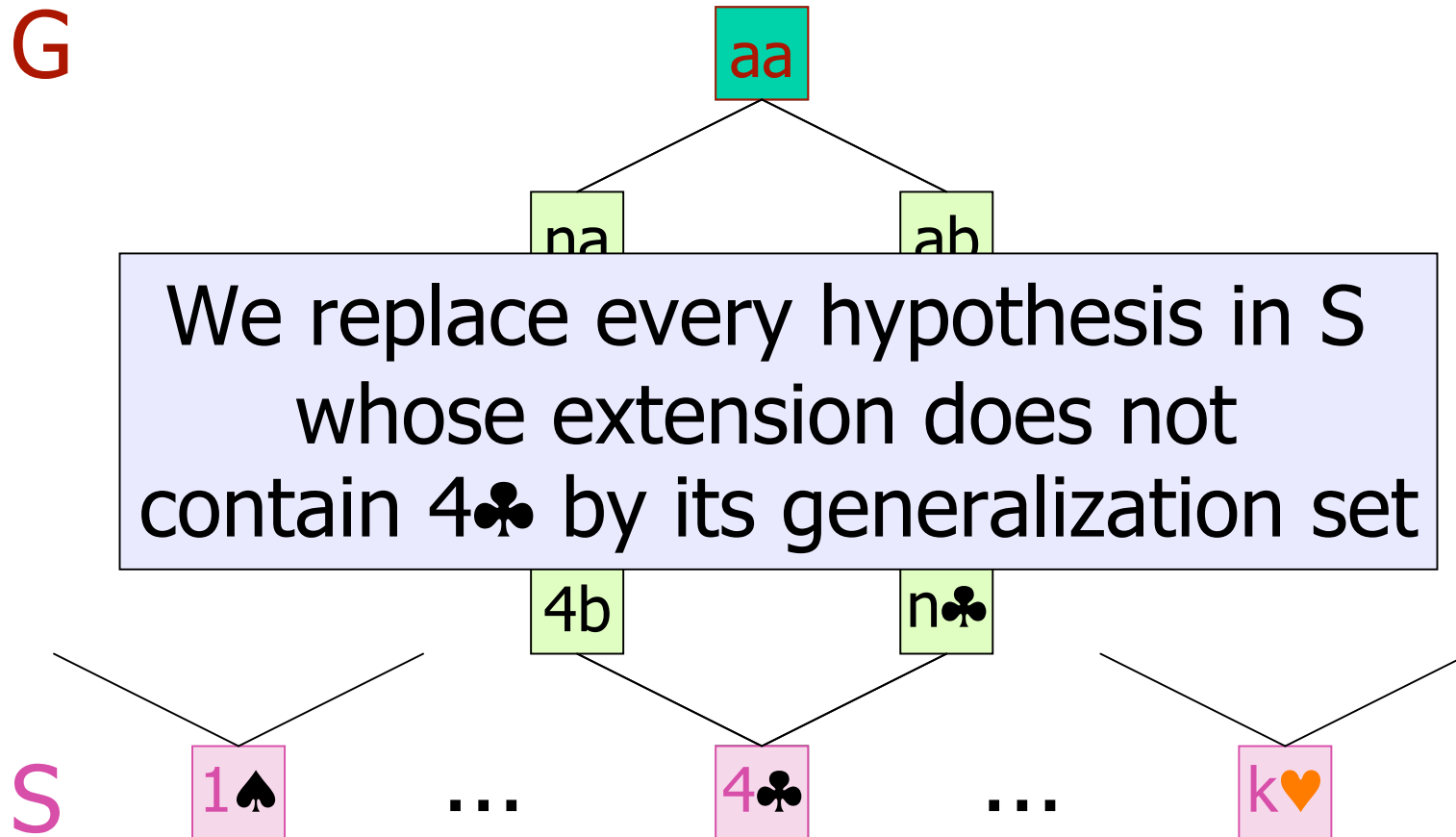
- A hypothesis in  $V$  is **most general** iff no hypothesis in  $V$  is more general
- **G-boundary**  $G$  of  $V$ : Set of most general hypotheses in  $V$

# G-Boundary / S-Boundary of V

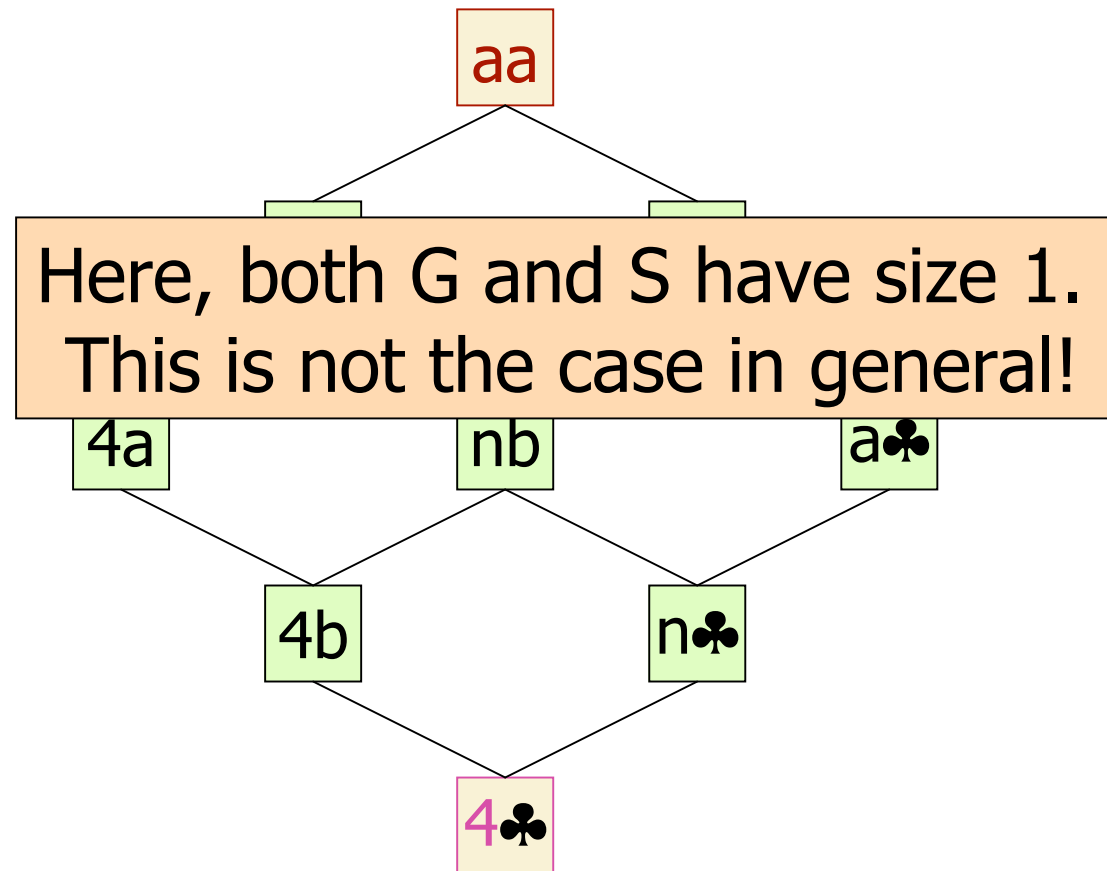
- A hypothesis in V is **most general** iff no hypothesis in V is more general
- **G-boundary** G of V: Set of most general hypotheses in V
- A hypothesis in V is **most specific** iff no hypothesis in V is more general
- **S-boundary** S of V: Set of most specific hypotheses in V



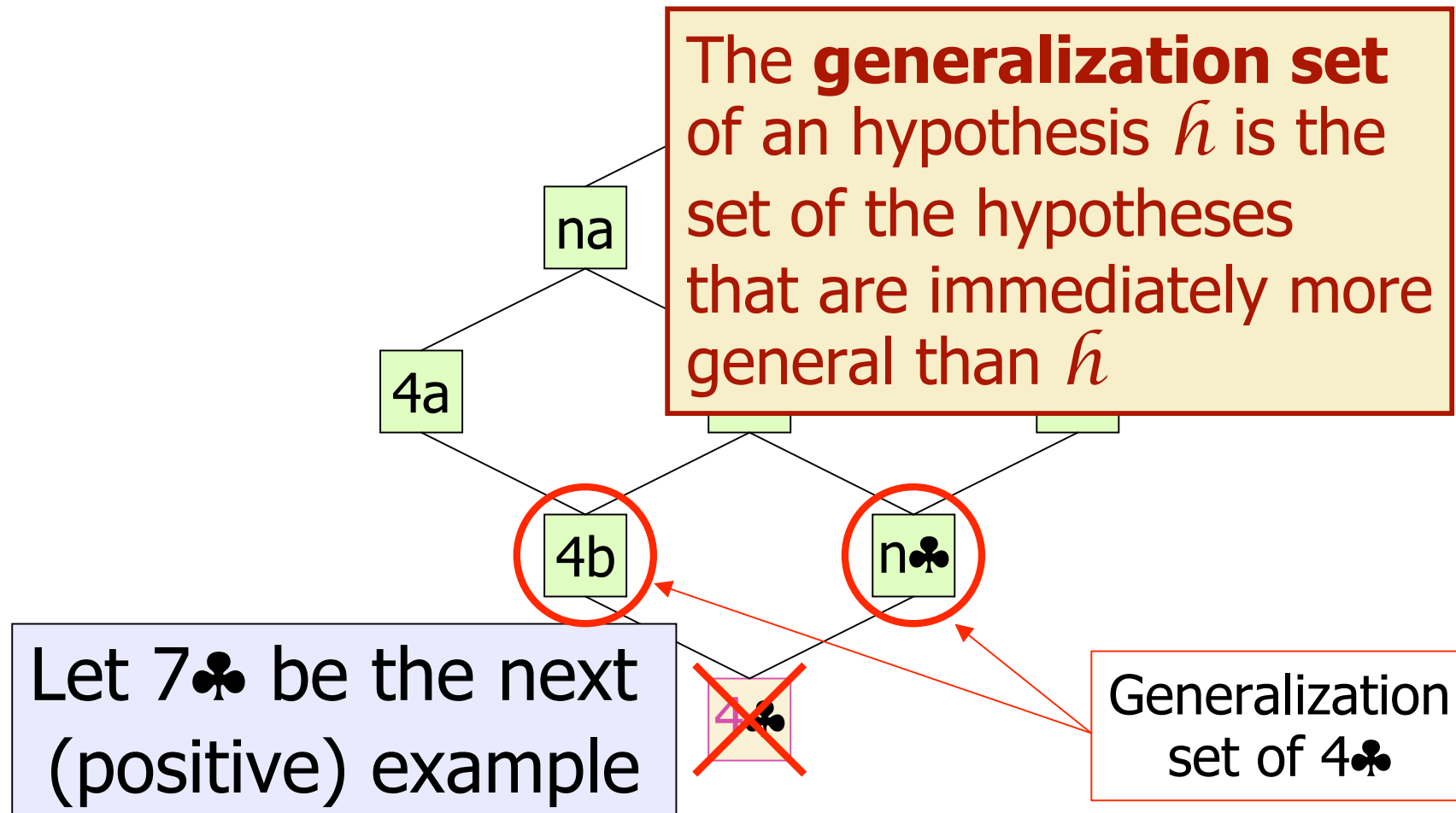
# Example: G-/S-Boundaries of V



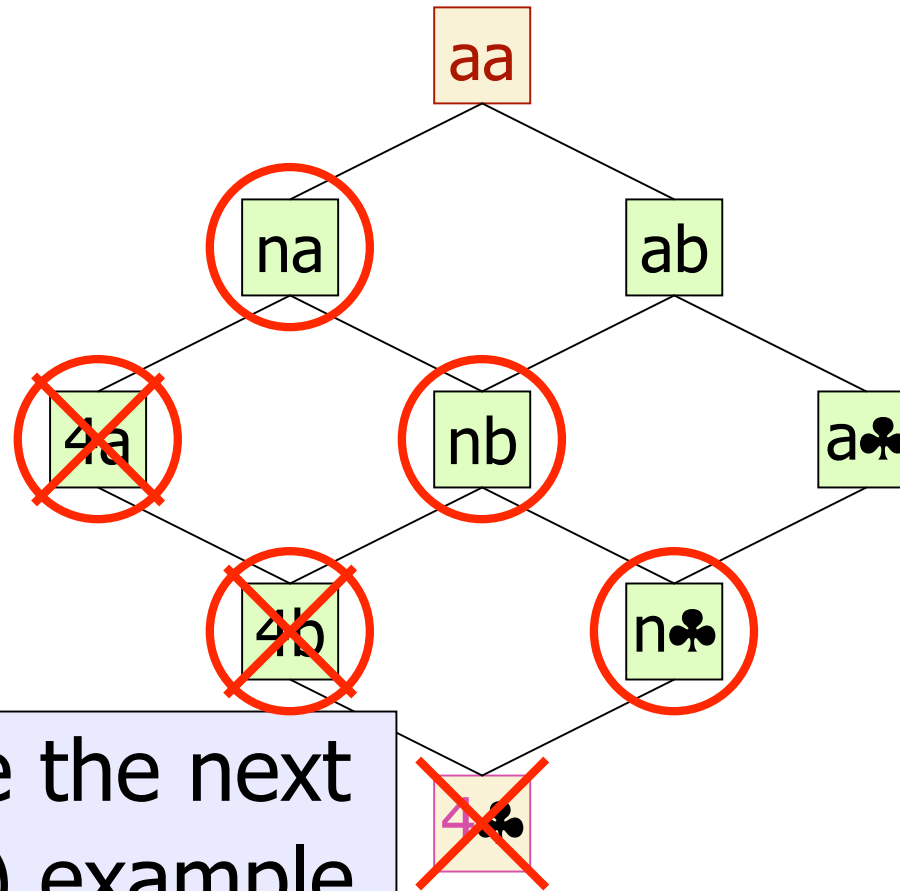
# Example: G-/S-Boundaries of V



# Example: G-/S-Boundaries of V

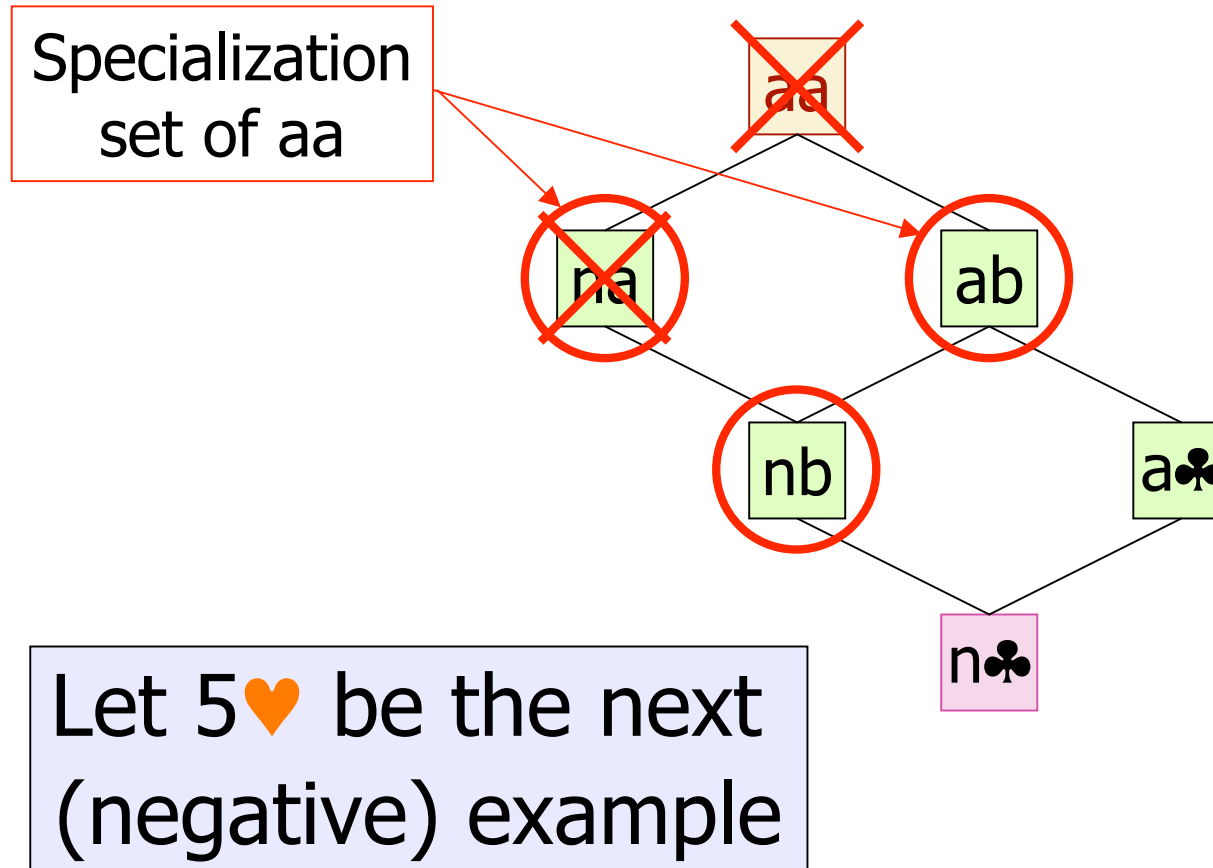


# Example: G-/S-Boundaries of V



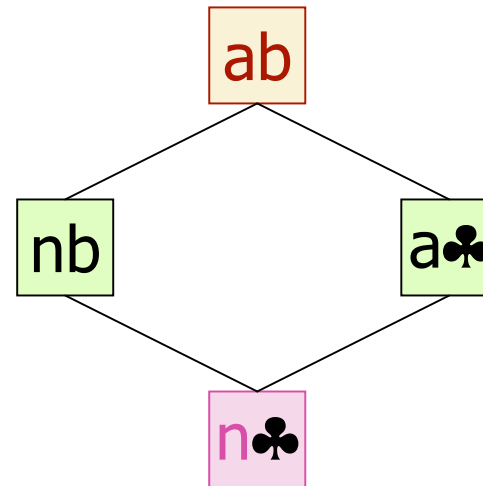
Let  $7♣$  be the next  
(positive) example

# Example: G-/S-Boundaries of V



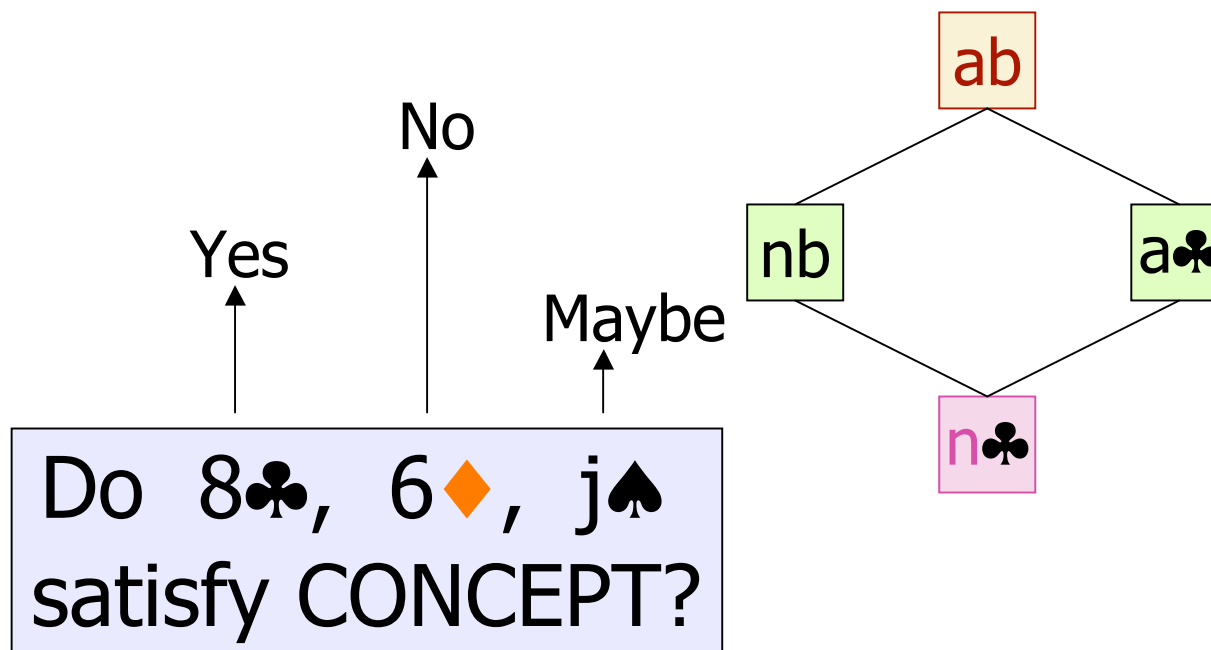
# Example: G-/S-Boundaries of V

G and S, and all hypotheses in between form exactly the version space

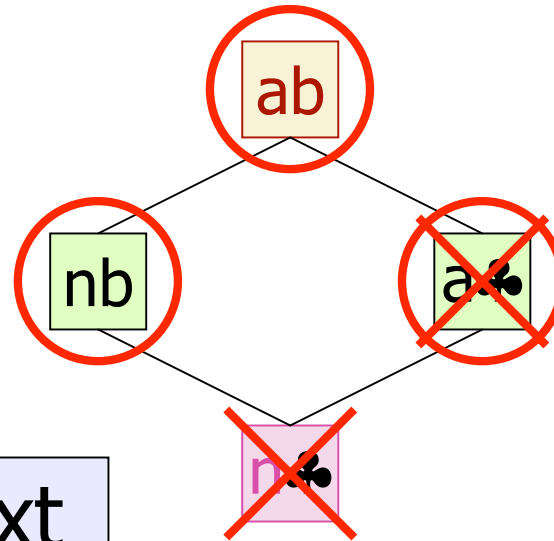


# Example: G-/S-Boundaries of V

At this stage ...

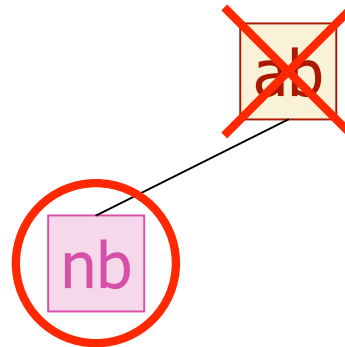


# Example: G-/S-Boundaries of V



Let  $2♠$  be the next  
(positive) example

# Example: G-/S-Boundaries of V



Let  $j_{\spadesuit}$  be the next  
(negative) example

# Example: G-/S-Boundaries of V

+ 4♣ 7♣ 2♠  
- 5♥ j♠

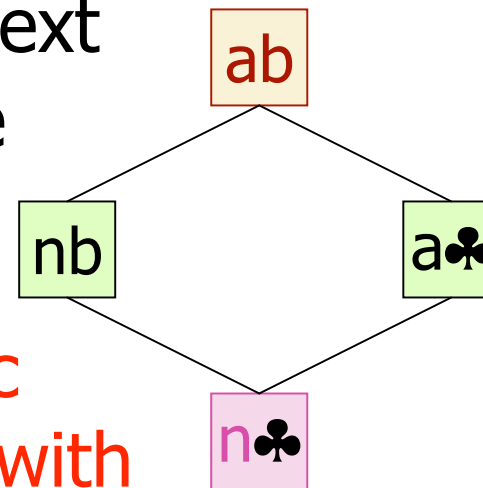
nb

$\text{NUM}(r) \wedge \text{BLACK}(s) \Leftrightarrow \text{IN-CLASS}([r,s])$

# Example: G-/S-Boundaries of V

Let us return to the version space ...

... and let  $8\clubsuit$  be the next (negative) example

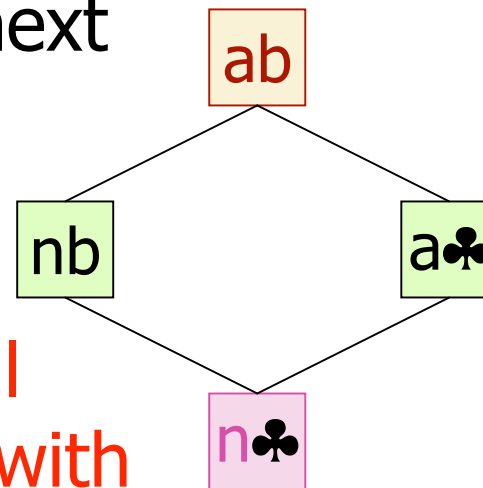


The only most specific hypothesis disagrees with this example, so no hypothesis in H agrees with all examples

# Example: G-/S-Boundaries of V

Let us return to the version space ...

... and let  $j \heartsuit$  be the next (positive) example



The only most general hypothesis disagrees with this example, so no hypothesis in H agrees with all examples

## Version Space Update

1.  $x \leftarrow$  new example
2. If  $x$  is positive then  
 $(G,S) \leftarrow$  POSITIVE-UPDATE( $G,S,x$ )
3. Else  
 $(G,S) \leftarrow$  NEGATIVE-UPDATE( $G,S,x$ )
4. If  $G$  or  $S$  is empty then return failure

## POSITIVE-UPDATE( $G, S, x$ )

1. Eliminate all hypotheses in  $G$  that do not agree with  $x$

## POSITIVE-UPDATE( $G, S, x$ )

1. Eliminate all hypotheses in  $G$  that do not agree with  $x$
2. Minimally generalize all hypotheses in  $S$  until they are consistent with  $x$

Using the generalization sets of the hypotheses

## POSITIVE-UPDATE( $G, S, x$ )

1. Eliminate all hypotheses in  $G$  that do not agree with  $x$
2. Minimally generalize all hypotheses in  $S$  until they are consistent with  $x$
3. **Remove** from  $S$  every hypothesis that is neither more specific than nor equal to a hypothesis in  $G$

This step was not needed in the card example

## POSITIVE-UPDATE( $G, S, x$ )

1. Eliminate all hypotheses in  $G$  that do not agree with  $x$
2. Minimally generalize all hypotheses in  $S$  until they are consistent with  $x$
3. Remove from  $S$  every hypothesis that is neither more specific than nor equal to a hypothesis in  $G$
4. Remove from  $S$  every hypothesis that is more general than another hypothesis in  $S$
5. Return  $(G, S)$

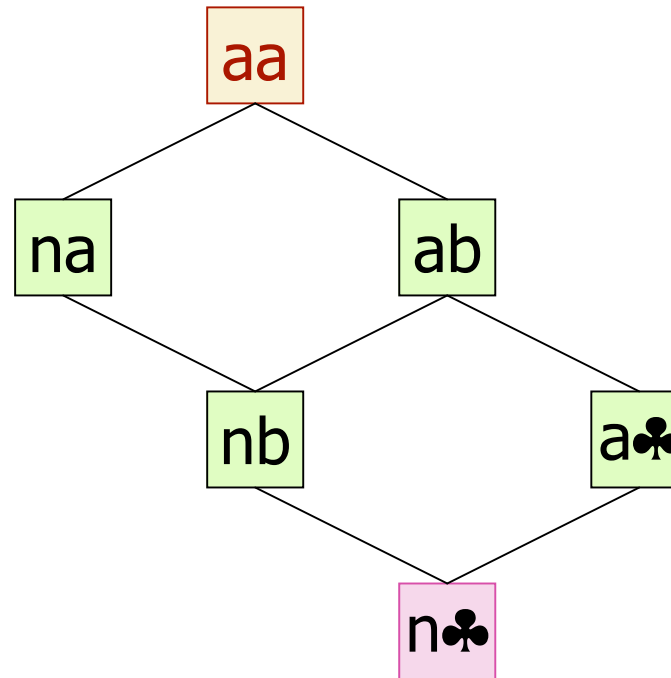
## NEGATIVE-UPDATE( $G, S, x$ )

1. Eliminate all hypotheses in  $S$  that do agree with  $x$
2. Minimally specialize all hypotheses in  $G$  until they are consistent with (exclude)  $x$
3. Remove from  $G$  every hypothesis that is neither more general than nor equal to a hypothesis in  $S$
4. Remove from  $G$  every hypothesis that is more specific than another hypothesis in  $G$
5. Return  $(G, S)$

# Example-Selection Strategy

- Suppose that at each step the learning procedure has the possibility to select the object (card) of the next example
- Let it pick the object such that, whether the example is positive or not, it will eliminate one-half of the remaining hypotheses
- Then a single hypothesis will be isolated in  $O(\log |H|)$  steps

# Example



- 9♣?
- j♥?
- j♣?

# Example-Selection Strategy

- Suppose that at each step the learning procedure has the possibility to select the object (card) of the next example
- Let it pick the object such that, whether the example is positive or not, it will eliminate one-half of the remaining hypotheses
- Then a single hypothesis will be isolated in  $O(\log |H|)$  steps
- But picking the object that eliminates half the version space may be expensive

# Noise

- If some examples are **misclassified**, the version space may collapse
- **Possible solution:**  
Maintain several G- and S-boundaries, e.g., consistent with all examples, all examples but one, etc...

# VSL vs DTL

- Decision tree learning (DTL) is more efficient if all examples are given in advance; else, it may produce successive hypotheses, each poorly related to the previous one
- Version space learning (VSL) is incremental
- DTL can produce simplified hypotheses that do not agree with all examples
- DTL has been more widely used in practice