

- *Martin's Law.* You can't learn anything unless you almost know it already.

# **Classification of Learning**

by Carbonell, Michalski, and Mitchell

- Rote learning and direct implanting of knowledge
- Learning from instruction
- Learning by analogy
- Learning from examples
  - Source is teacher
  - Source is learner
  - Source is external environment
- Learning from Observation and discovery
  - Passive observation
  - Active experimentation

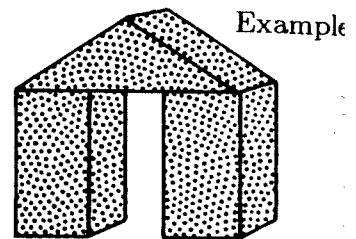
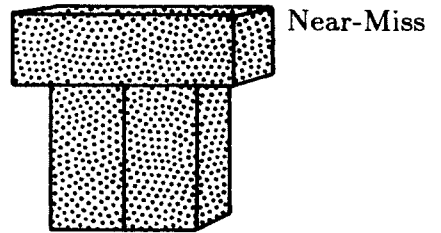
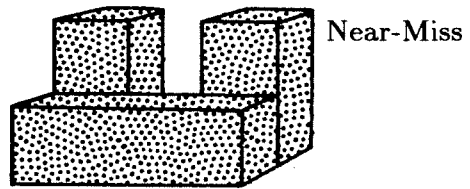
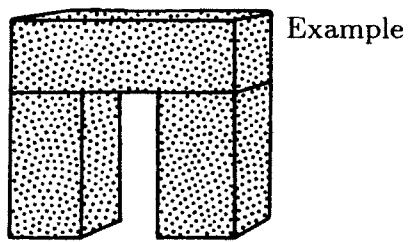


Figure 11-1. A sequence of examples and near misses for learning about arches.

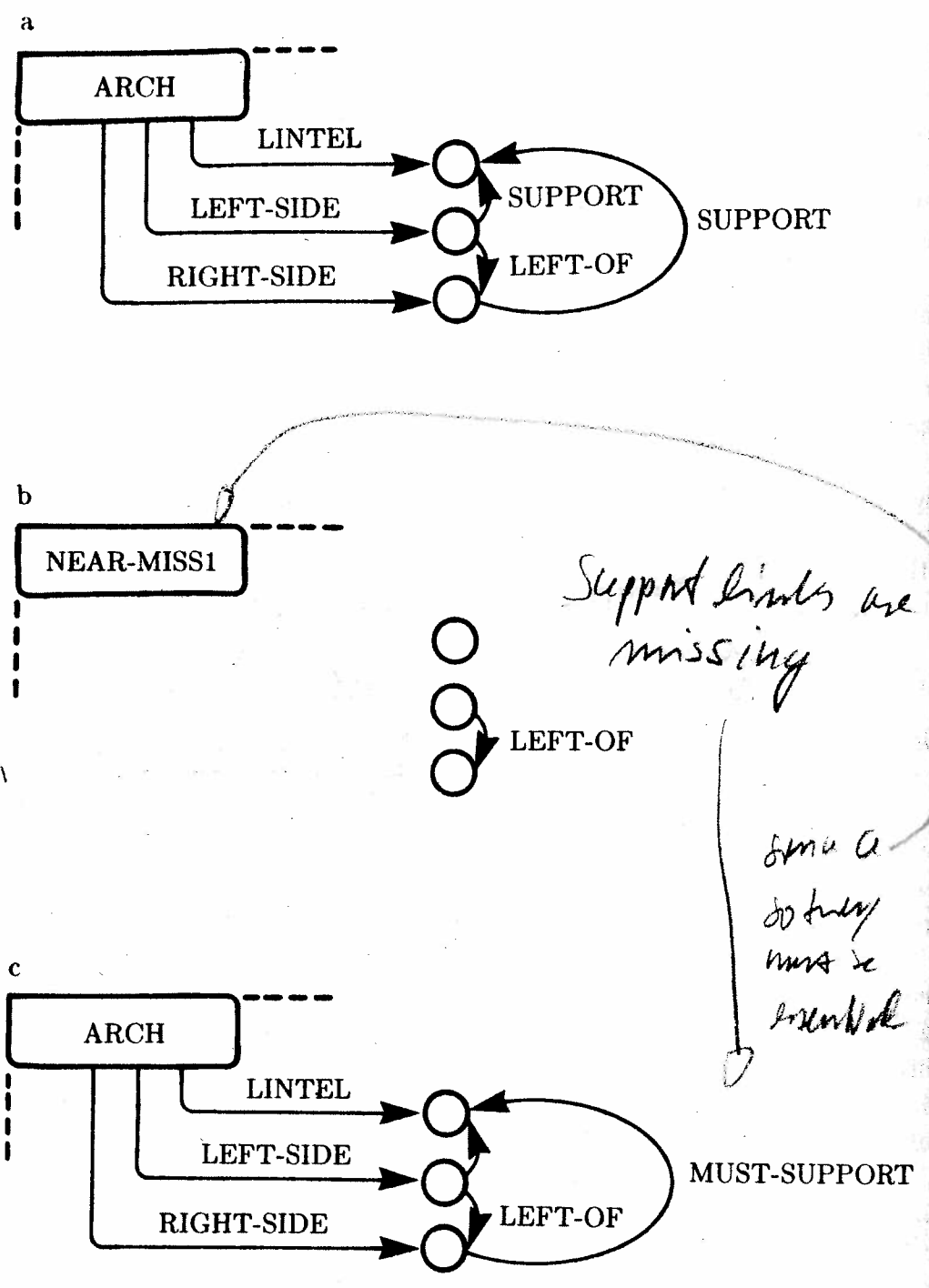


Figure 11-2. The require-link generalization rule. Compared with the ARCH frame in a, the near-miss frame in b lacks SUPPORT links. The conclusion is that SUPPORT links are essential, so the SUPPORT links in the ARCH frame are altered, indicating that they are required in all arches, as shown in c. The LEFT-OF link is shown to emphasize the need for evidence that is sufficient to establish the correct correspondence between the parts of the arch and the parts of the near miss. Many links have been omitted from the drawing for clarity.

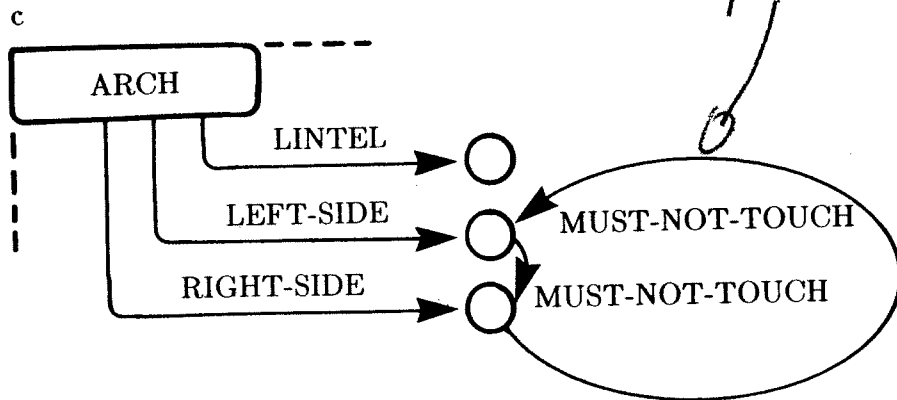
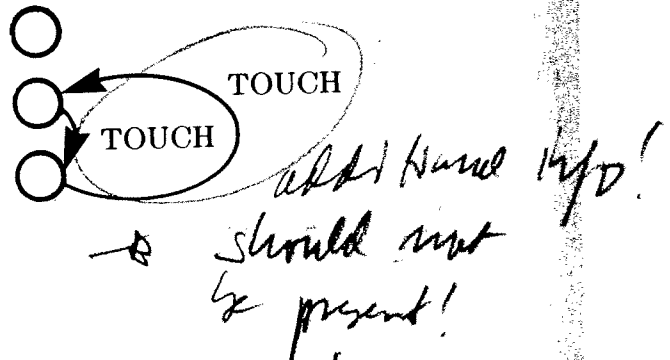
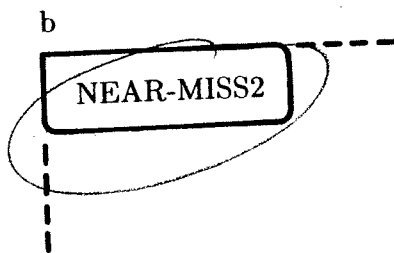
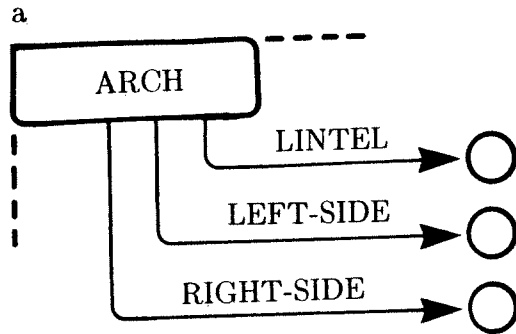


Figure 11-3. The forbid-link generalization rule. Compared with the ARCH frame in a, the near-miss frame in b adds touch links. The conclusion is that the touch links must not be present, so touch links are added to the ARCH frame, altered to indicate that they are forbidden in all arches, as shown in c. Many links have been omitted from the drawing for clarity.

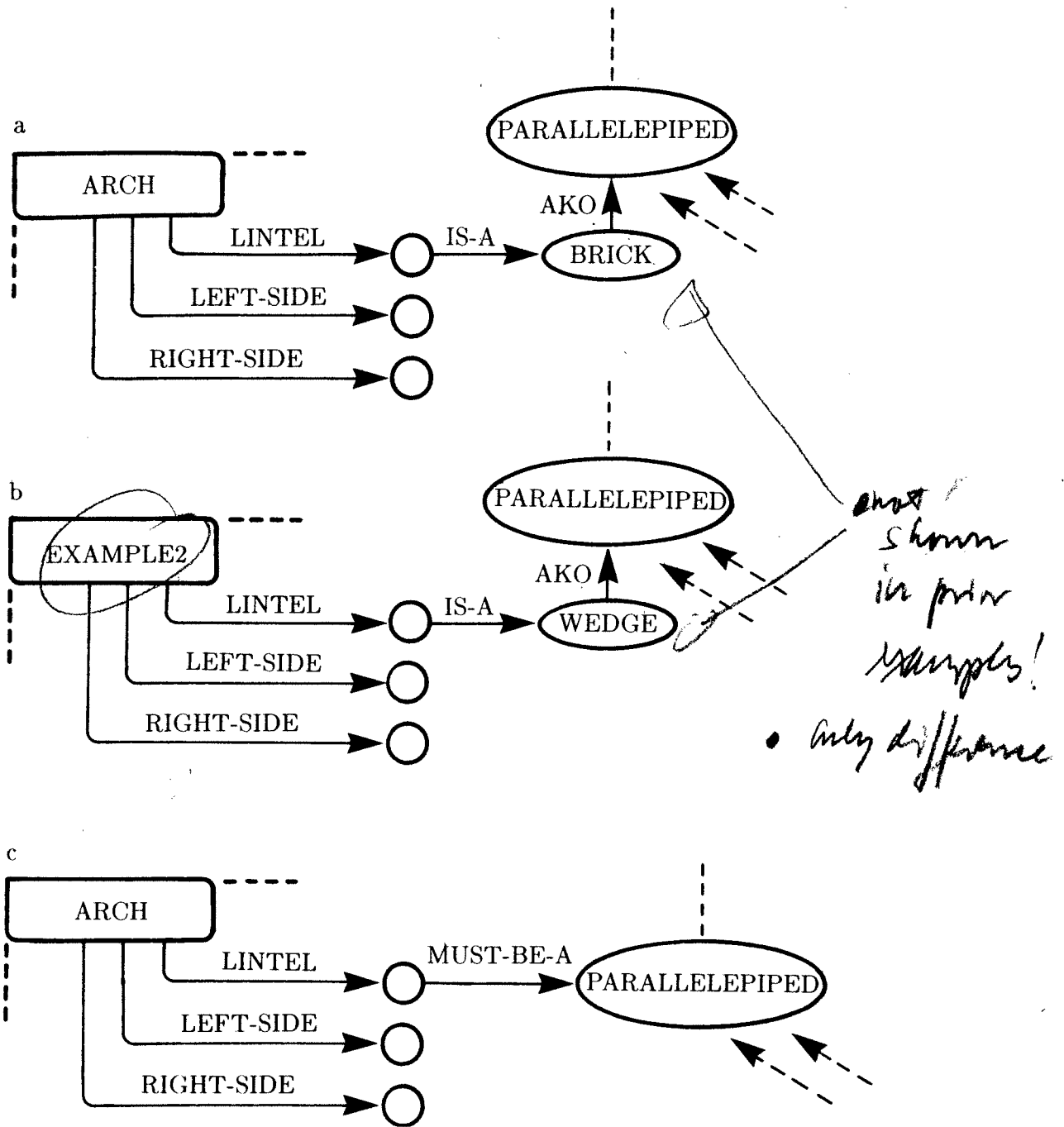


Figure 11-4. The climb-tree heuristic. The lintel in the ARCH frame in *a* is a brick, while the corresponding object in the example frame in *b* is a wedge. Evidently it does not matter. The IS-A link in the ARCH frame is changed to MUST-BE-A and redirected from BRICK to PARALLELEPIPED, as shown in *c*, which is the most specific common generalization of BRICK and WEDGE. Many links have been omitted from the drawing for clarity.

near miss 1

near miss 2

Example 2

if you can't  
use

because it  
does not  
matter!

✓  
makes  
sense

- The *require-link* heuristic is used when an evolving model has a link in a place where a near miss does not. The model link is converted to a MUST form.
- The *forbid-link* heuristic is used when a near miss has a link in a place where an evolving model does not. A MUST-NOT form is installed in the evolving model.
- The *climb-tree* heuristic is used when something in an evolving model corresponds to a different thing in an example. MUST-BE-A links are routed to the most specific shared class in the classification tree above the model thing and the example thing.
- The *enlarge-set* heuristic is used when something in an evolving model corresponds to a different thing in an example. MUST-BE-A links are routed to a new class composed of the union of the things' classes.
- The *drop-link* heuristic is used when the things that are different in an evolving model and in an example form an exhaustive set. The *drop-link* heuristic is also used when an evolving model has a link that is not in the example. The link is dropped from the model.
- The *close-interval* heuristic is used when a number or interval in an evolving model corresponds to a number in an example. If the model uses a number, the number is replaced by an interval spanning the model's number and the example's number. If the model uses an interval, the interval is enlarged to reach the example's number.

To learn using the W procedure:

- 1 Let the description of the first sample, which must be an example, be the initial description.
  - 2 For all subsequent samples:
    - 2a If the sample is a near miss, use procedure SPECIALIZE. *Important!*
    - 2b If the sample is an example, use procedure GENERALIZE.
- 

**SPECIALIZE:**

- 1 Match the evolving model to the sample to establish correspondences among parts.
- 2 Determine whether there is a single, most important difference between the evolving model and the near miss:
  - 2a If there is a single, most important difference, determine whether the evolving model or the near miss has a link that is not in the other:
    - 2aa If the evolving model has a link that is not in the near miss, use the require-link heuristic.
    - 2ab If the near miss has a link that is not in the model, use the forbid-link heuristic.
  - 2b Otherwise ignore the sample.

*How do we determine that?*

**GENERALIZE:**

- 1 Match the evolving model to the sample to establish correspondences among parts.
- 2 For each difference, determine the difference type:
  - 2a If the difference is that the link points to a different class in the evolving model from the class the link points to in the sample, determine if the classes are part of a classification tree:
    - 2aa If the classes are part of a classification tree, use the climb-tree heuristic.
    - 2ab If the classes form an exhaustive set, use the drop-link heuristic.
    - 2ac Otherwise, use the enlarge-set heuristic.
  - 2b If the difference is that a link is missing in either the evolving model or the example, use the drop-link heuristic. *J*
  - 2c If the difference is that different numbers, or an interval and a number outside the interval, are involved, use the close-interval heuristic. *✓*
  - 2d Otherwise ignore the difference.