Title
Why Some Modified Class Inclusion Tasks Are Easy for Young Children: a Process Model for Finding Referents of Labels in Arrays

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 3(0)

Author
Abrahamsen, Adele A.

Publication Date
1981

Peer reviewed

Why some modified class inclusion tasks are easy for young children: A process model for finding referents of labels in arrays

Adele A. Abrahamsen
New School for Social Research

Inhelder & Piaget's standard class inclusion problem involves the presentation of an array, e.g., five toy horses and three toy dogs, and the posing of a question concerning the larger subordinate class (A_1) and the superordinate class (A): "Are there more horses or more animals?" Children younger than about eight years usually say "More horses."

Where do they go wrong? Wilkinson (1976) has suggested the use of a strategy that forbids double-counting. He outlines a SCAN procedure that young children apply only once (avoiding double-counting), and older children apply twice.

Despite the intuitive appeal of this model, there is evidence that children have difficulties correctly assigning referents in certain related tasks using nonincluded sets, for which double-counting is not at issue (McGarrigle, Grieve & Hughes, 1978; Isen, Riley, Tucker & Trabasso, 1975). Conversely, Markman (1973) showed that children perform well on included sets if the more inclusive set is a collection, e.g., "family," rather than a class. Finally, Trabasso, Isen, Dolecki, McLanahan, Riley & Tucker (1978) reviewed a number of modified class inclusion tasks for which performance is modestly or dramatically better than for the standard task.

Though Trabasso et al. interpreted the findings in terms of eight component processes in class inclusion, they emphasized that the standard task encourages children to identify the smaller subordinate class as the referent of the superordinate term, whereas many of the modified tasks make the superordinate reference unambiguous. In this paper I expand on that insight by outlining two simple process models for finding the referents of labels in arrays, and showing that these models can account for younger and older children's performance on standard tasks and on those modified tasks which elicit dramatic improvements.

Process A (for younger children)

I propose that young children proceed as follows for class inclusion and related problems:

1. They exhaustively partition the array of objects into mutually exclusive sets. I assume that they tend to maximize similarity within sets and minimize similarity between sets, hence favoring small sets, but I do not model this procedure.
2. They seek referents for labels by conducting a self-terminating search of the sets.
3. They permit the referent sets of two labels to intersect (i.e., double-counting), but they cannot be identical; a search that would otherwise be terminated will continue to avoid this.

Figures 1 and 2 show modified flow charts of the two most important procedures of Process A. These are incomplete, but are detailed enough to address the gross empirical data, and are similar in format to Wilkinson's (1976) flow charts to facilitate comparison.

It is assumed that the child has constructed two lists: a list S* of the sets in the array and a list L* of verbal labels. Any relevant noun phrase in the class inclusion question or in the immediate verbal context is included in L*. The procedure LINK acts on these lists, linking labels to sets, and keeping track of the linkages by constructing a third list of labels and their referents: (L,REF)*. LINK processes every label in turn, getting its definition and calling the procedure SETSEARCH to find a matching set description. Once a set is found to have a matching description (by the procedure MATCH, not detailed here), the search terminates unless the result is a reference already "claimed" by another label, as determined by the procedure COMPARE. Whenever a set is "claimed" by a label, it is marked as used (#) and moved to the end of the list S*. After all labels have been processed, LINK uses LASTSEARCH to try once again to make a linkage for any remaining unclaimed sets. This is not detailed here, but for each such set a self-terminating search of the labels is performed.

Process B (for older children)

Process B conducts an exhaustive search of the sets for each label, and permits two labels to have identical referent sets. Intermediate processes incorporating only one of these two changes are of course possible. The changes primarily affect SETSEARCH (see Figure 3), but the exhaustive search feature makes the check for leftover sets in LINK unnecessary. Process B obtains the correct answer to standard class inclusion problems, unless the labels are defined too narrowly to match atypical sets.

Evidence

Figures 3 and 4 trace the highlights of Processes A and B, respectively, as applied to several versions of class inclusion tasks. The standard task is followed by tasks in which both subordinate labels are made salient (Ahr & Youniss, 1970; Winer, 1974) and in which several subordinate classes appear in the array (McLanahan, 1976); these versions are easy for young children. Last, three conditions involving atypical classes are shown (Carson & Abrahamson, 1976); these are difficult even for some older children.

With minor modification, Process A can also handle the findings of Markman (1973). The definition of a collection specifies its constituents, e.g., a set of parents and a set of children are the constituents of a family. A precedure CONSTITUENT could be inserted into LINK; it would call SETSEARCH separately for each constituent.

Processes A and B account well for the gross data. They could be tested more stringently by testing their processing sequences against detailed records.

References

Ahr, P. & Youniss, J. Reasons for failure on the class-inclusion problem. Child Development, 1970, 41, 131-143.

Carson, M. T. & Abrahamson, A. Some members are more equal than others: The effect of semantic typicality on class-inclusion performance. Child Development, 1976, 47, 1186-1190.

Isen, A. M., Riley, C. A., Tucker, T. & Trabasso, T. The facilitation of class inclusion by use of multiple comparisons and two-class perceptual displays. Paper presented at the meeting of SRCD, Denver, Colorado, April 1975.

Markman, E. The facilitation of part-whole comparisons by use of the collective noun "family." Child Development, 1973, 44, 837-840.

McGarrigle, J., Grieve, R. & Hughes, M. Interpreting inclusion: A contribution to the study of the child's cognitive and linguistic development. Journal of Experimental Child Psychology, 1978, 26, 528-550.

McLanahan, A. G. The class-inclusion problem: An information processing interpretation. Senior Honors Thesis, Princeton University, 1976.

Trabasso, T., Isen, A. M., Dolecki, P., McLanahan, A. G., Riley, C. A. & Tucker, T. How do children solve class-inclusion problems? In R. S. Siegler (Ed.), Children's Thinking: What Develops? Hillsdale, N. J.: Erlbaum, 1978.

Wilkinson, A. Counting strategies and semantic analysis as applied to class inclusion. Cognitive Psychology, 1976, 8, 64-85.

Winer, G. A. An analysis of verbal facilitation of class inclusion reasoning. Child Development, 1974, 45, 224-227.

```
        ( LINK(L*,S*,(L,REF)*) )
Find the next label L in L*
Make the reference of L empty: REF   NIL
Get the definition of L: DEF
REF = SETSEARCH(S*,REF,DEF,(L,EXT)*)
Form the pair (L,REF)
Move (L,REF) to (L,REF)*
  End of L*?— Yes—  All S in S* used?— Yes— Exit
     No                      No
                       Delete every #S from S
    LINK               LASTSEARCH
```

Figure 1. Modified flow chart of the LINK procedure used by both Process A and Process B (boxed section needed only for Process A).

```
      ( SETSEARCH(S*,REF,DEF,(L,EXT)*) )
Find the next set S in S*
Get the description of S: DES
M   MATCH(DEF,DES)
  M = NIL? ——— Yes— End of S*?— Yes— ?
    No                    No
Copy S into REF          SETSEARCH
Mark S as used: #S
Move S to end of S*
V   COMPARE(REF,(L,REF)*)
  V   CLAIMED?— Yes— End of S*?— Yes— ?
       No                 No
      Exit               SETSEARCH

      ( SETSEARCH(S*,REF,DEF,(L,EXT)*) )
Find the next set S in S*
Get the description of S: DES
M = MATCH(DEF,DES)
  M   NIL? ——— Yes— End of S*?—Yes— Exit
    No                    No
Copy S into REF          SETSEARCH
  End of S*?—Yes—Exit
      No
  SETSEARCH
```

Figure 2. Modified flow chart of the SETSEARCH procedure used by Processes A (top) and B (bottom).

Standard task; Carson & Abrahamsen (1976) typical-typical condition. 5 horses & 3 dogs.
L*: "Horses, Animals" S*: HORSES, DOGS
L1: "Horses"/HORSES-Match-OK/ S*: DOGS, #HORSES
L2: "Animals"/DOGS-Match-OK/ S*: #HORSES, #DOGS
Result: 5 "horses" & 3 "animals" so "More horses"

Ahr & Youniss (1970); Winer (1974). 5 horses 3 dogs.
L*: "Horses, Dogs, Animals" S*: HORSES, DOGS
L1: "Horses"/HORSES-Match-OK/ S*: DOGS, #HORSES
L2: "Dogs"/DOGS-Match-OK/ S*: #HORSES, #DOGS
L3: "Animals"/HORSES-Match-But claimed, so try more /DOGS-Match-Ref of HORSES & DOGS is OK/
Result: 5 "horses" & 8 "animals" so "More animals"

McLanahan (1976). 4 horses, 2 dogs, 2 cats, 2 pigs.
L*: "Horses, Animals" S*: HORSES, DOGS, CATS, PIGS
L1: "Horses"/HORSES-Match-OK/
S*: DOGS, CATS, PIGS, #HORSES
L2: "Animals"/DOGS-Match-OK/
S*: CATS, PIGS, #HORSES, #DOGS
Have reached end of L*, but some sets unused.
Delete used sets to get S*:CATS,PIGS. Do LASTSEARCH.
S1: CATS/"Horses"-Mismatch/"Animals"-Match-OK/
S2: PIGS/"Horses"-Mismatch/"Animals"-Match-OK/
Result: 4 "horses" & 6 "animals" so "More animals"

Carson & Abrahamson (1976) atypical-atypical condition. 5 bees & 3 flies.
L*: "Bees, Animals" S*: BEES, FLIES
L1: "Bees"/BEES-Match-OK/ S*: FLIES, #BEES
L2: "Animals"/FLIES-Mismatch/#BEES-Mismatch/
Have reached end of L*, but one set unused.
Delete used set to get S*: FLIES. Do LASTSEARCH.
S1: FLIES/"Bees"-Mismatch/"Animals"-Mismatch/
Result: 5 "bees" & 0 "animals" so "More bees"

Carson & Abrahamson (1976) atypical-typical condition. 5 bees & 3 dogs.
L*: "Bees, Animals" S*: BEES, DOGS
L1: "Bees"/BEES-Match-OK/ S*: DOGS, #BEES
L2: "Animals"/DOGS-Match-OK/ S*: #BEES, #DOGS
Result: 5 "bees" & 3 "animals" so "More bees"

Carson & Abrahamson (1976) typical-atypical condition. 5 horses & 3 flies.
L*: "Horses, Animals" S*: HORSES, FLIES
L1: "Horses"/HORSES-Match-OK/ S*: FLIES, #HORSES
L2: "Animals"/FLIES-Mismatch/#HORSES-Match-But claimed, and no more sets to try. What to do?
Result: varies. Some compare the 5 HORSES to the 3 FLIES and answer "More horses"

Figure 3. Process A as a model of younger children's performance in several class inclusion studies: Highlights of possible sequences.

Standard task; Carson & Abrahamson (1976) typical-typical condition. 5 horses & 3 dogs.

L*: "Horses, Animals" S*: HORSES, DOGS
L1: "Horses"/HORSES-Match/DOGS-Mismatch/
L2: "Animals"/HORSES-Match/DOGS-Match/
Result: 5 "horses" & 8 "animals" so "More animals"

Ahr & Youniss (1970); Winer (1974). 5 horses 3 dogs

L*: "Horses, Dogs, Animals" S*: HORSES, DOGS
L1: "Horses"/HORSES-Match/DOGS-Mismatch/
L2: "Dogs"/HORSES-Mismatch/DOGS-Match/
L3: "Animals"/HORSES-Match/DOGS-Match/
Result: 5 "horses" & 8 "animals" so "More animals"

McLanahan (1976). 4 horses, 2 dogs, 2 cats, 2 pigs.

L*: "Horses,Animals" S*: HORSES, DOGS, CATS, PIGS
L1: "Horses"/HORSES-Match/DOGS-Mismatch/
CATS-Mismatch/PIGS-Mismatch/
L2: "Animals"/HORSES-Match/DOGS-Match/
CATS-Match/PIGS-Match/
Result: 4 "horses" & 8 "animals" so "More animals"

Carson & Abrahamson (1976) atypical-atypical condition. 5 bees & 3 flies.

L*: "Bees, Animals" S*: BEES, FLIES
L1: "Bees"/BEES-Match/FLIES-Mismatch/
L2: "Animals"/BEES-Mismatch/FLIES-Mismatch/
Result: 5 "bees" & 0 "animals" so "More bees"

Carson & Abrahamson (1976) atypical-typical condition. 5 bees & 3 dogs.

L*: "Bees, Animals" S*: BEES, DOGS
L1: "Bees"/BEES-Match/DOGS-Mismatch/
L2: "Animals"/BEES-Mismatch/DOGS-Match/
Result: 5 "bees" & 3 "animals" so "More bees"

Carson & Abrahamson (1976) typical-atypical condition. 5 horses & 3 flies.

L*: "Horses, Animals" S*: HORSES, FLIES
L1: "Horses"/HORSES-Match/FLIES-Mismatch/
L2: "Animals"/HORSES-Match/FLIES-Mismatch/
Result: 5 "horses" & 5 "animals" so "Same"

Figure 4. Process B as a model of older children's performance in several class inclusion studies: Highlights of possible sequences.