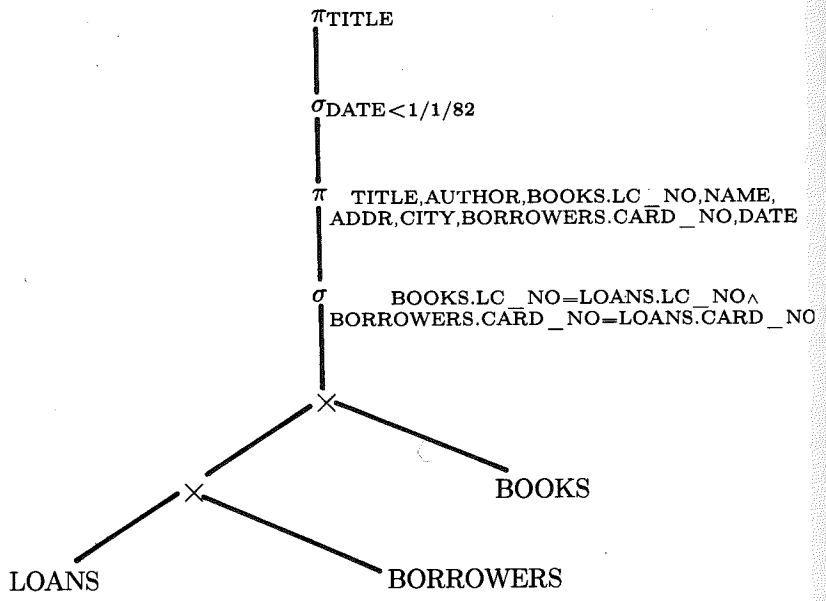

Fig. 8.1. Parse tree of expression.

defined as

$$\pi_S(\sigma_F(\text{LOANS}\times\text{BORROWERS}\times\text{BOOKS}))$$

where

$$F = \text{BORROWERS.CARD\_NO=LOANS.CARD\_NO}$$
$$\textit{AND}\ \text{BOOKS.LC\_NO=LOANS.LC\_NO}$$

while

$$S = \text{TITLE, AUTHOR, PNAME, LC\_NO, NAME,}$$
$$\text{ADDR, CITY, CARD\_NO, DATE}$$

We might wish to list the books that been borrowed before some date in the distant past, say 1/1/82 by:

$$\pi_{\text{TITLE}}\sigma_{\text{DATE}<1/1/82}(\text{XLOANS})$$

After substituting for XLOANS, the expression above has the parse tree shown in Fig. 8.1.

The first step of the optimization is to split the selection F into two, with conditions BOOKS.LC_NO = LOANS.LC_NO and

$$\text{BORROWERS.CARD\_NO} = \text{LOANS.CARD\_NO}$$

respectively. Then we move each of the three selections as far down the tree

as possible. The selection $\sigma_{\text{DATE}<1/1/82}$ moves below the projection and the two selections by rules (4) and (5). This selection then applies to the product (LOANS × BORROWERS) × BOOKS. Since DATE is the only attribute mentioned by the selection, and DATE is an attribute only of LOANS, we can replace

$$\sigma_{\text{DATE}<1/1/82}((\text{LOANS}\times\text{BORROWERS})\times\text{BOOKS})$$

by

$$(\sigma_{\text{DATE}<1/1/82}(\text{LOANS}\times\text{BORROWERS}))\times\text{BOOKS}$$

then by

$$((\sigma_{\text{DATE}<1/1/82}(\text{LOANS}))\times\text{BORROWERS})\times\text{BOOKS}$$

We have now moved this selection as far down as possible. The selection with condition BOOKS.LC_NO = LOANS.LC_NO cannot be moved below either Cartesian product, since it involves an attribute of BOOKS and an attribute not belonging to BOOKS.† However, the selection on

$$\text{BORROWERS.CARD\_NO} = \text{LOANS.CARD\_NO}$$

can be moved down to apply to the product

$$\sigma_{\text{DATE}<1/1/82}(\text{LOANS})\times\text{BORROWERS}$$

Note that LOANS.CARD_NO is the name of an attribute of

$$\sigma_{\text{DATE}<1/1/82}(\text{LOANS})$$

since it is an attribute of LOANS, and the result of a selection takes its attributes to be the same as those of the expression to which the selection is applied.

Next, we can combine the two projections into one, π_{TITLE}, by rule (3). The resulting tree is shown in Fig. 8.2. Then by the extended rule (5) we can replace π_{TITLE} and $\sigma_{\text{BOOKS.LC\_NO=LOANS.LC\_NO}}$ by the cascade

$$\pi_{\text{TITLE}}$$
$$\sigma_{\text{BOOKS.LC\_NO=LOANS.LC\_NO}}$$
$$\pi_{\text{TITLE,BOOKS.LC\_NO,LOANS.LC\_NO}}$$

We apply rule (9) to replace the last of these projections by

$$\pi_{\text{TITLE,BOOKS.LC\_NO}}$$

applied to BOOKS, and $\pi_{\text{LOANS.LC\_NO}}$ applied to the left operand of the higher Cartesian product in Fig. 8.2.

† We could use the commutative and associative laws of products and then move this selection down one level, but then we could not move the selection on BORROWERS.CARD_NO = LOANS.CARD_NO down.

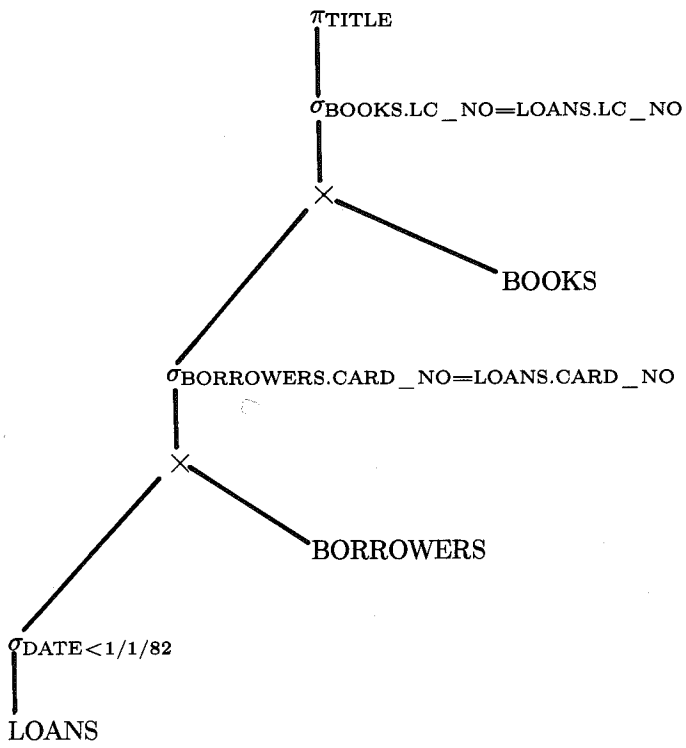

Fig. 8.2. Tree with selections lowered and projections combined.

The latter projection interacts with the selection below it by the extended rule (5) to produce the cascade

$\pi_{\text{LOANS.LC\_NO}}$
$\sigma_{\text{BORROWERS.CARD\_NO=LOANS.CARD\_NO}}$
$\pi_{\text{LOANS.LC\_NO,BORROWERS.CARD\_NO,LOANS.CARD\_NO}}$

The last of these projections passes through the Cartesian product by rule (9) and passes partially through the selection $\sigma_{\text{DATE}<1/1/82}$ by the extended rule (5). We then discover that in the expression

$\pi_{\text{LOANS.LC\_NO,LOANS.CARD\_NO,DATE}}$

the projection is superfluous, since all attributes of LOANS are mentioned. We therefore eliminate this projection. The final tree is shown in Fig. 8.3. In that figure we have indicated groups of operators by dashed lines. Each of the Cartesian products is effectively an equijoin, when combined with the selection above. In particular, the selection on LOANS and the projection of

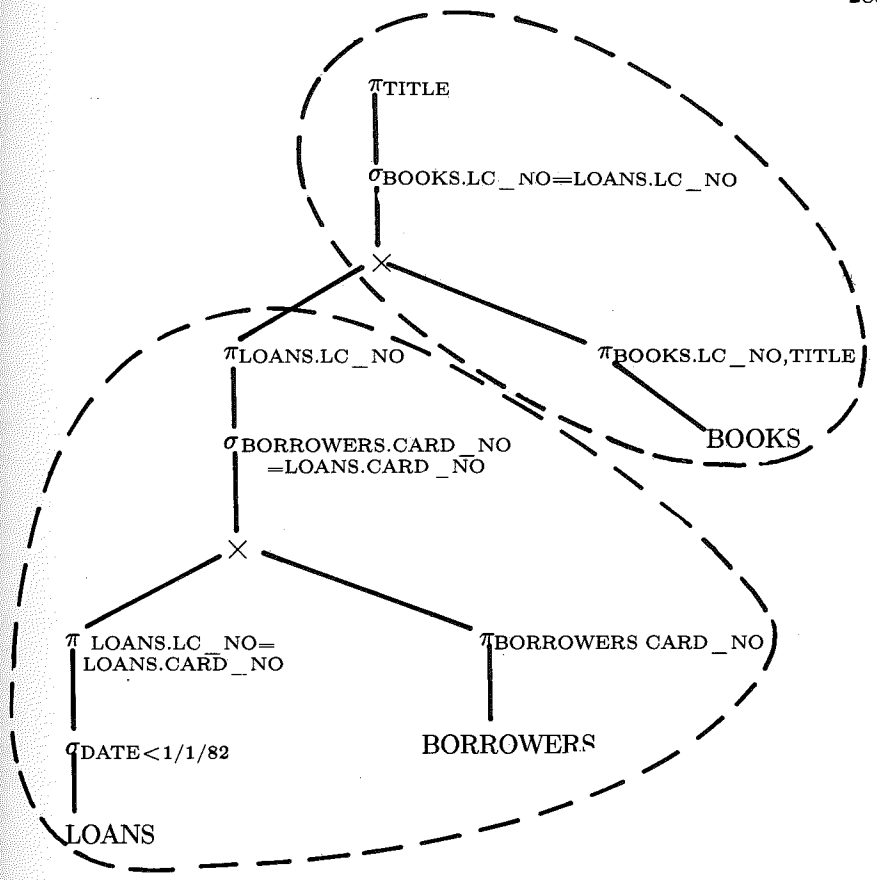

Fig. 8.3. Final tree with grouping of operators.

BORROWERS below the first product can be successfully combined with that product. Obviously a program executing Fig. 8.3 will perform the lower group of operations before the upper. □

8.3 OPTIMIZATION OF SELECTIONS IN SYSTEM R

We shall focus in this section on a problem that is instructive for several reasons. First, it shows a great deal about the opportunities for optimization in even a simple kind of query. Second, it lets us sample the issues at the implementation level, and third, it is representative of the way System R does all its optimization, which is quite different from the methodology followed by most other systems (although the results are largely the same).

The problem we consider is one in which we are given a query of the form