

then

This routine first tests that we are at expression level (if not, it fails) and then completes the processing of the algebraic expression following if by calling TAKE and UNSTAK. If is then unstacked and ARITH, E and EXPTYP are restored. The state variable E is then used to decide whether then should be stacked as "then S" or "then E". The word pointer (PP) is also stacked, ready to update the IFJ when the delimiter else is met.

ERRORS

FAIL 97; then in statement

FAIL 69; corresponding if has been omitted
or conditional expression without
an else.