## **prec_com**$^{9,38}$

Suggested uses for generic precedences are:
(examples given in ()'s)

Inmain_prec:
preop 1 arg term with prefix text (not, minus, squash)
postop 1 arg term with postfix text (pi1, pi2)
inop 2 arg term with infix text (compose, append)
wk_preop term with prefix and maybe infix text (mon_reduce)
wk_postop term with postfix and maybe infix text
bd_preop term with prefix and maybe infix text (exists, for, lambda)
bd_postop term with postfix and maybe infix text

The bd_*op are intended for terms that bind 1 or more variables in their rightmost argument. Having these weak precedences corresponds with the convention that common binding terms have maximal scope.

The exposed slot of pre and postop terms (including wk_* and bd_*) should have E paren control so that ops can be stacked without adding parenthesization.

terms that use inop should use the infix_df_gen display form generator from the boot theory. This uses iteration control to suppress parenthesization of right associated term rather than E paren control, since then multiple dfs can use inop and will always be parenthesized when nested inside one another.

Inlogic_prec:

atomrel: atomic relations with infix and maybe postfix text (equal, member, lt)

A good place to add pointers to new auxiliary precedence objects is
in parallel with *arith_prec*.