fail is an atom that always fails. In combination with cut, it can be used to eliminate exceptional cases:

flies(X) :- ostrich(X),!, fail.
flies(X) :- penguin(X),!, fail.
flies(X) :- dead(X),!, fail.
flies(X) :- bird(X).

**Negation as (finite) failure**

Prolog has a special operator `not` that applies to goals (atoms).

It is defined as if by the following code

\[
\text{not}(A) :- A, !, \text{fail}.
\]

\[
\text{not}(A).
\]

i.e. to “prove” `not(A)`, first try to prove `A`. If that succeeds, then `not(A)` fails. If the attempt to prove `A` terminates without success, then conclude `not(A)`.

(If the attempt to prove `A` gets stuck in an infinite loop, so does `not(A)`.)
Negation as finite failure is not true negation

flies(X) :- not(ostrich(X)), not(penguin(X)),
           not(dead(X)), bird(X).

ostrich(ozzie). bird(ozzie).
parakeet(tweety). bird(tweety).

: flies(polly)? :flies(tweety)?
  ** no  ** yes

: flies(X)?
  ** no

odd(3).

: not(not(odd(3)))?
  ** yes

: not(not(odd(X)))?

X = _R0
Debugging in Prolog

trace – trace all execution events
notrace – turn tracing off
spy P – trace just a specific predicate P
nospy P – turn of spying on P

Messages written during tracing:

CALL: <atom> – <atom> has become leftmost atom,
starting work on it

EXIT: <atom> – a proof for <atom> for atom has been found,
starting work on the next atom

REDO: <atom> – backtracking, looking for another
way to prove <atom>

FAIL: <atom> – have exhausted all avenues to prove <atom>,
backtraking out of it.
Commands such as the following can be entered during a trace (depends on which system you use):

- **c** (creep) do an exhaustive trace until the next spy point
- **l** (leap) jump to the next spy point
- **s** (skip) don’t trace this particular atom
- **r** (retry) retry a goal (with a given number) (XSB)
- **a** (abort) terminate execution (XSB)

In addition to this, you could add write statements at specific points of the program – this will tell you if these points are being reached, and what the values of the variables of interest are at those points.