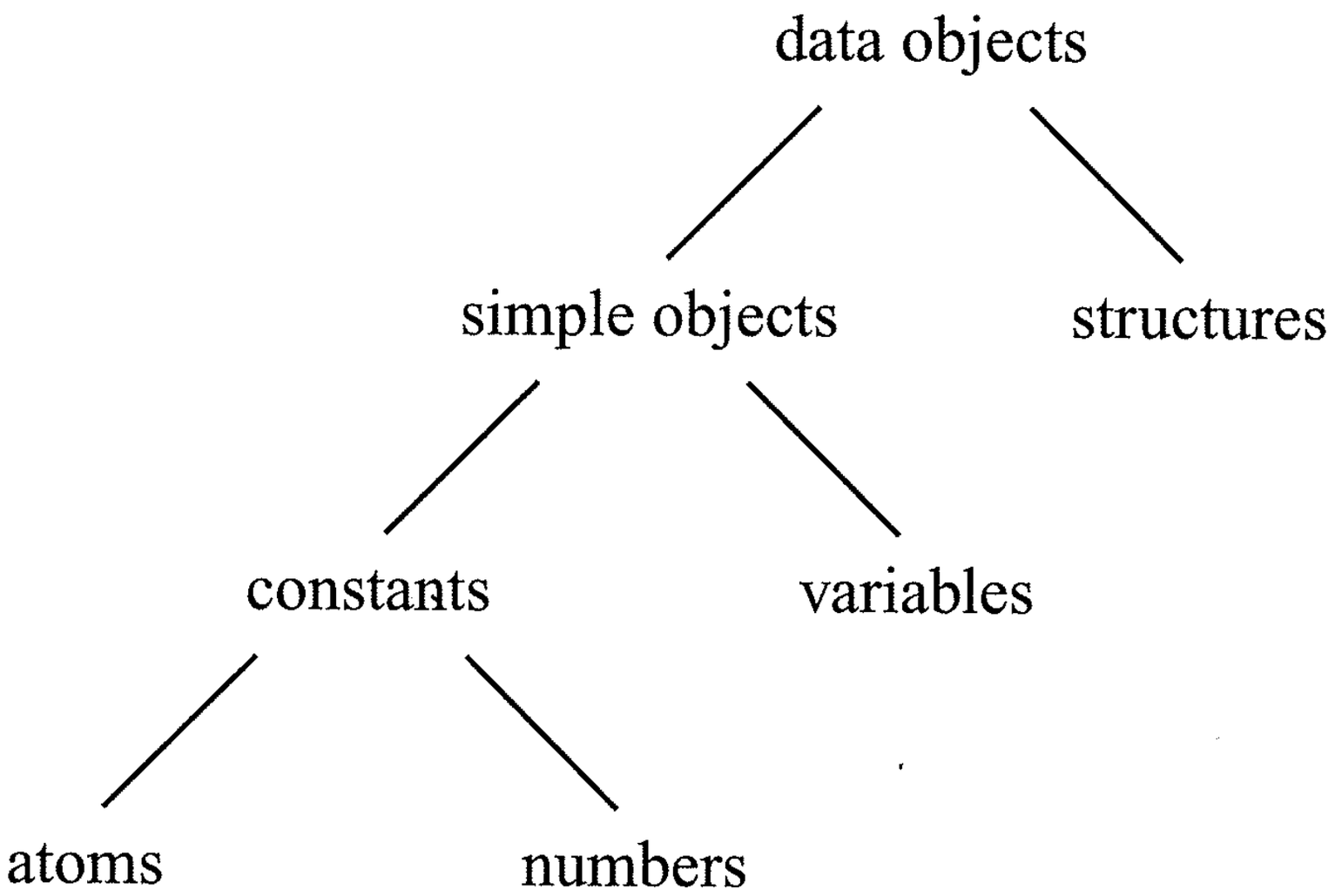

Figure 2.1 Data objects in Prolog.

Terms

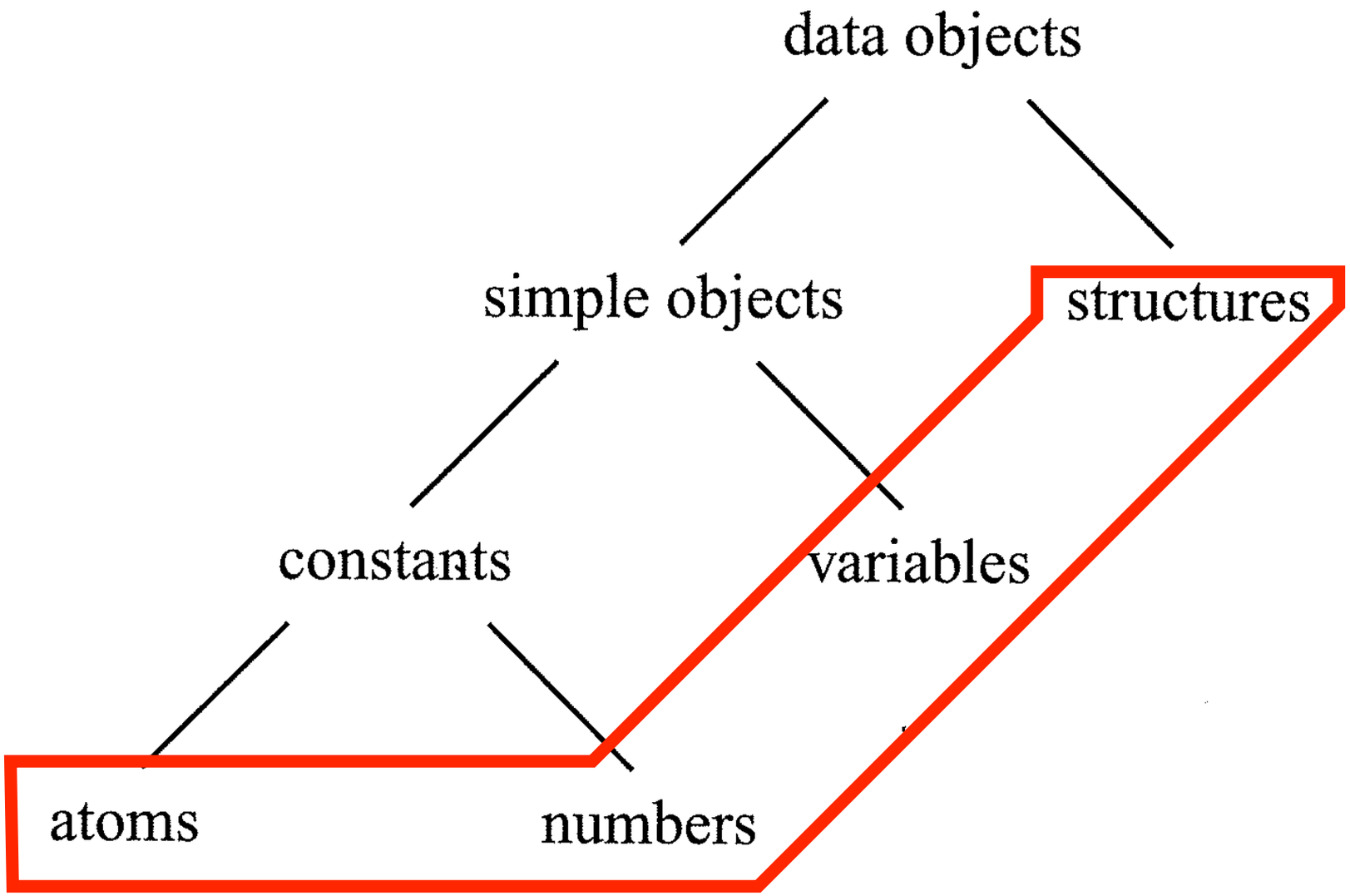

Figure 2.1 Data objects in Prolog.

Atoms, numbers, variables and structures are all terms.

- The type of a term can be tested by the following predicates:

var(X)	X is a (non-instantiated) variable
nonvar(X)	X is not a variable
atom(X)	X is an atom
integer(X)	X is an integer
float(X)	X is a real number
atomic(X)	X is either an atom or a number
compound(X)	X is a structure

- Terms can be constructed or decomposed:

Term =.. [Functor | ArgumentList]
functor(Term, Functor, Arity)
arg(N, Term, Argument)
name(Atom, CharacterCodes)

- Terms can be compared:

X = Y	X and Y match
X == Y	X and Y are identical
X \== Y	X and Y are not identical
X =:= Y	X and Y are arithmetically equal
X =\= Y	X and Y are not arithmetically equal
X < Y	arithmetic value of X is less than Y (related: =<, >, >=)
X @< Y	term X precedes term Y (related: @=<, @>, @>=)

- A Prolog program can be viewed as a relational database that can be updated by the following procedures:

assert(Clause)	add **Clause** to the program
asserta(Clause)	add at the beginning
assertz(Clause)	add at the end
retract(Clause)	remove a clause that matches **Clause**

If you want to have a clause in your database and you want to be able to add or remove it dynamically (that is, when you query), you must declare it to be `dynamic` in gprolog with the `:- dynamic` designation. The compiler needs this designation when it consults your database.

You can add or remove any clause not already in your database without the `:-` designation.

`gprolog` does not have `assert/1`.

- All the objects that satisfy a given condition can be collected into a list by the predicates:

bagof(X, P, L)	L is the list of all X that satisfy condition P
setof(X, P, L)	L is the sorted list of all X that satisfy condition P

Recall the mathematical definition of a bag compared to the definition of a set.
A bag can have duplicates.
A set cannot. For example, {a, b, b} = {a, b}.

- Built-in procedures for reading and writing characters and terms are:

read(Term)	input next term
write(Term)	output **Term**
put(CharCode)	output character with the given ASCII code
get0(CharCode)	input next character
get(CharCode)	input next 'printable' character

Summary of `read( X)`

- The next term `T` is read and matched with `X`.
- If `X` is a variable, then `X` is instantiated to `T`.
- If there is no match, the goad `read( X)` fails with no backtracking.
- If <control-d> is read from the keyboard, or the end of file is reached in a file, `X` is instantiated to `end_of_file`.

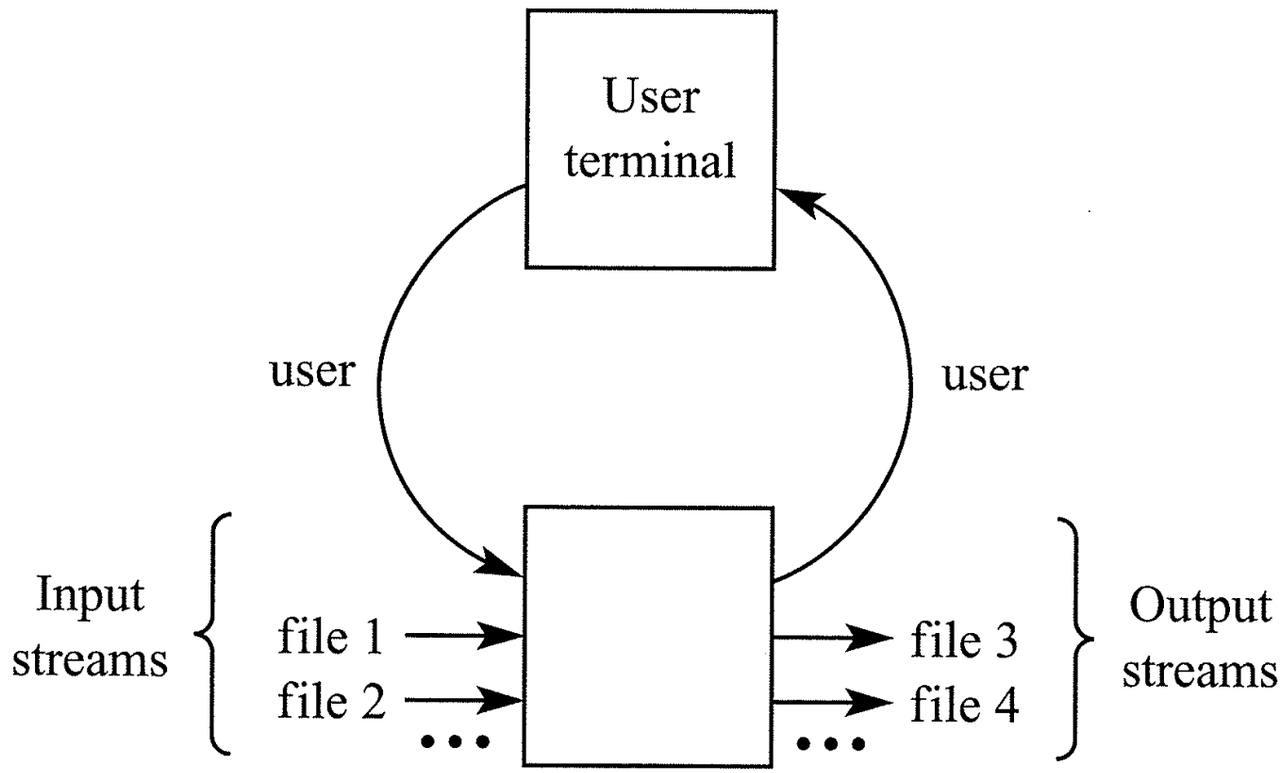

Figure 6.5 Communication between a Prolog program and several files.

- Switching between streams is done by:

see(File)	**File** becomes the current input stream
tell(File)	**File** becomes the current output stream
seen	close the current input stream
told	close the current output stream