Prolog Syntax

Data domain
- **constants**
  - numeric: as usual
  - symbolic: alphanumeric string starting with lowercase letter

- **variables** (unknowns): alphanumeric string starting with uppercase letter

Terms
- constants and variables
- records: \( f(t_1, \ldots, t_n), n \geq 0, \)
  where \( f \) is a valid record name [symbolic constants are used], and \( t_i \)
  \((1 \leq i \leq n)\) is a term; also the usual infix operators are allowed (e.g., +, -, *)
- lists: \([t_1, \ldots, t_n], n \geq 0,\)
  where \( t_i \) \((1 \leq i \leq n)\) is a term; also \([H \mid T]\) is allowed, where \( H \) is a term (the head of the list), and \( T \) is a term (the tail of the list).

Formulas
- **atomic formulas**: \( p(t_1, \ldots, t_n), (n \geq 0 \text{ and for } n=0 \text{ write } p) \)
  where \( t_i \) \((1 \leq i \leq n)\) is a term and \( p \) is a predicate name [symbolic constants are used]; also certain infix "names" are allowed (e.g., <, =)

- (Horn) **clauses**: \( A :\!-\! B_1, \ldots, B_n, (n \geq 0, \text{ and for } n=0 \text{ write } A,) \)
  where \( A \) and \( B_i \) \((1 \leq i \leq n)\) are atomic formulas; \( A \) is called the head of the clause, and \( B_1, \ldots, B_n \) is called the body. If \( n=0 \), the clause is called a fact; for \( n>0 \), the clause is a conditional assertion.

- **logic program**: a finite set of clauses

Definite Clause Grammar (DCG) Rules
- \( A \rightarrow B_1, \ldots, B_n, (n \geq 1) \)
  where \( A \) is an atomic formula (i.e., non-terminal), and \( B_i \) \((1 \leq i \leq n)\) is either an atomic formula or a list constant (including []). The right-hand side represents a list formed as a concatenation of those of the \( B_i \) \((1 \leq i \leq n)\), where non-terminal values are determined by other DCG rules.