# Mathematical Formulation of the LPProblem

Given a linear programming problem, we aim to either minimize or maximize a linear objective function subject to a set of linear constraints.

Let:

- $\mathbf{c} \in \mathbb{R}^n$ be the vector of objective function coefficients.
- $\mathbf{A} \in \mathbb{R}^{m \times n}$ be the constraint matrix.
- $\mathbf{b} \in \mathbb{R}^m$ be the vector representing the right-hand side of the constraints.
- $\mathbf{l} \in \mathbb{R}^n$ be the vector of lower bounds for the variables.
- $\mathbf{u} \in \mathbb{R}^n$ be the vector of upper bounds for the variables.
- $\mathbf{x} \in \mathbb{R}^n$ be the vector of decision variables.
- $\text{vars} = \{x_1, x_2, \ldots, x_n\}$ be the variable names.
- $\text{constraint\_types} = \{c_1, c_2, \ldots, c_m\}$ where $c_i \in \{\leq, \geq, =\}$ represents the type of each constraint.

The general mathematical formulation is as follows:

## Objective Function

$$\text{Minimize or Maximize:} \quad \mathbf{c}^\top \mathbf{x}$$

depending on the value of `is_minimize` (if True, minimize; if False, maximize).

## Constraints

$$\mathbf{A}\mathbf{x} \begin{cases} \leq \mathbf{b}, & \text{if } c_i = \leq \\ \geq \mathbf{b}, & \text{if } c_i = \geq \\ = \mathbf{b}, & \text{if } c_i == \end{cases} \quad \text{for } i = 1, 2, \ldots, m$$

## Variable Bounds

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ are the decision variables, and the bounds are element-wise.

## Code Formulation

```julia
struct LPProblem
    is_minimize::Bool  # True if the objective is to minimize
    c::Vector{Float64}  # Objective function coefficients
    A::SparseMatrixCSC{Float64, Int64}  # Constraint matrix
    b::Vector{Float64}  # Right-hand side of constraints
    l::Vector{Float64}  # Variable lower bounds
    u::Vector{Float64}  # Variable upper bounds
    vars::Vector{String}  # Variable names
    constraint_types::Vector{Char}  # Constraint types
end
```