Simplified Surreal Numbers Implementation

1 Data Structure

The simplified surreal number structure is defined as an algebraic data type:

$$\text{Surreal} = \begin{cases} \text{Zero} \\ \text{One} \\ \text{NegOne} \\ \text{Succ}(x) & \text{where } x \text{ is Surreal} \\ \text{Pred}(x) & \text{where } x \text{ is Surreal} \\ \text{Sum}(x, y) & \text{where } x, y \text{ are Surreal} \\ \text{Prod}(x, y) & \text{where } x, y \text{ are Surreal} \end{cases} \tag{1}$$

2 Basic Operations

2.1 Ordering

For surreal numbers x and y, the ordering is defined as:

$$\begin{aligned}
\text{compare}(\text{Zero}, \text{Zero}) &= \text{EQ} \\
\text{compare}(\text{One}, \text{One}) &= \text{EQ} \\
\text{compare}(\text{NegOne}, \text{NegOne}) &= \text{EQ} \\
\text{compare}(\text{Succ}(x), \text{Succ}(y)) &= \text{compare}(x, y) \\
\text{compare}(\text{Pred}(x), \text{Pred}(y)) &= \text{compare}(x, y) \\
\text{compare}(\text{Zero}, \text{One}) &= \text{LT} \\
\text{compare}(\text{One}, \text{Zero}) &= \text{GT} \\
\text{compare}(\text{Zero}, \text{NegOne}) &= \text{GT} \\
\text{compare}(\text{NegOne}, \text{Zero}) &= \text{LT}
\end{aligned}$$

2.2 Normalization

The normalization function reduces surreal numbers to their simplest form:

$$
\begin{aligned}
\text{normalize}(\text{Zero}) &= \text{Zero} \\
\text{normalize}(\text{One}) &= \text{One} \\
\text{normalize}(\text{NegOne}) &= \text{NegOne} \\
\text{normalize}(\text{Succ}(\text{Zero})) &= \text{One} \\
\text{normalize}(\text{Pred}(\text{Zero})) &= \text{NegOne} \\
\text{normalize}(\text{Sum}(x, y)) &= \text{addNormalized}(\text{normalize}(x), \text{normalize}(y)) \\
\text{normalize}(\text{Prod}(x, y)) &= \text{multNormalized}(\text{normalize}(x), \text{normalize}(y))
\end{aligned}
$$

3 Arithmetic Operations

3.1 Addition

For normalized surreal numbers x and y:

$$\text{addNormalized}(x, y) = \text{fromInt}(\text{evalToInt}(x) + \text{evalToInt}(y)) \tag{2}$$

3.2 Multiplication

For normalized surreal numbers x and y:

$$\text{multNormalized}(x, y) = \text{fromInt}(\text{evalToInt}(x) \times \text{evalToInt}(y)) \tag{3}$$

3.3 Negation

$$
\begin{aligned}
\text{negate}(\text{Zero}) &= \text{Zero} \\
\text{negate}(\text{One}) &= \text{NegOne} \\
\text{negate}(\text{NegOne}) &= \text{One} \\
\text{negate}(\text{Succ}(x)) &= \text{Pred}(\text{negate}(x)) \\
\text{negate}(\text{Pred}(x)) &= \text{Succ}(\text{negate}(x)) \\
\text{negate}(\text{Sum}(x, y)) &= \text{Sum}(\text{negate}(x), \text{negate}(y)) \\
\text{negate}(\text{Prod}(x, y)) &= \text{Prod}(\text{negate}(x), y)
\end{aligned}
$$

4 Integer Conversion

The conversion between integers and surreal numbers is defined as:

$$\text{fromInt}(n) = \begin{cases} \text{Zero} & \text{if } n = 0 \\ \text{One} & \text{if } n = 1 \\ \text{NegOne} & \text{if } n = -1 \\ \text{Succ}(\text{fromInt}(n-1)) & \text{if } n > 0 \\ \text{Pred}(\text{fromInt}(n+1)) & \text{if } n < 0 \end{cases} \tag{4}$$

5 Successor Function

The nth successor of a surreal number x is defined as:

$$\text{succ}'(n, x) = \underbrace{\text{Succ}(\text{Succ}(\cdots \text{Succ}(x) \cdots))}_{n \text{ times}} \tag{5}$$