Definition 1.11. The range of a normalized FPN system is a subset of \( \mathbb{R} \),
\[
R(\mathcal{F}) := \{ x : x \in \mathbb{R}, \text{UFL}(\mathcal{F}) \leq |x| \leq \text{OFL}(\mathcal{F}) \}. \tag{1.9}
\]

Example 1.14. Consider a normalized FPN system with the characterization \( \beta = 2, p = 3, L = -1, U = +1 \).

The four FPNs
\[
1.00 \times 2^0, \ 1.01 \times 2^0, \ 1.10 \times 2^0, \ 1.11 \times 2^0
\]
correspond to the four ticks in the plot starting at 1 while
\[
1.00 \times 2^1, \ 1.01 \times 2^1, \ 1.10 \times 2^1, \ 1.11 \times 2^1
\]
correspond to the four ticks starting at 2.

Definition 1.12. Two normalized FPNs \( a, b \) are adjacent to each other in \( \mathcal{F} \) if
\[
\forall c \in \mathcal{F} \setminus \{ a, b \}, \ |a - b| < |a - c| + |c - b|. \tag{1.10}
\]

Lemma 1.13. Let \( a, b \) be two adjacent normalized FPNs satisfying \( |a| < |b| \) and \( ab > 0 \). Then
\[
\beta^{-1} \epsilon_M |a| < |a - b| \leq \epsilon_M |a|. \tag{1.11}
\]

Proof. Consider \( a > 0 \), then \( \Delta a := b - a > 0 \). By Definitions 1.4 and 1.6, \( a = m x \beta^p \) with \( 1.0 \leq m < \beta \). \( a \) and \( b \) only differ from each other at the least significant digit, hence \( \Delta a = \epsilon_M \beta^p \). Since \( \frac{\Delta a}{a} < \frac{\epsilon_M}{a} \leq \epsilon_M \) and thus \( \frac{\Delta a}{a} \in (\beta^{-1} \epsilon_M, \epsilon_M) \). The other cases are similar. \( \square \)

Definition 1.14. The subnormal or denormalized numbers are FPNs of the form \( (1.1) \) with \( e = L \) and \( m \in (0, 1) \). A normalized FPN system can be extended by including the subnormal numbers.

Remark 1.15. Subnormal numbers have inherently lower precision than normalized FPNs because they have fewer significant digits in the fractional part. As another difference from normal FPNs, subnormal numbers are evenly spaced.

Example 1.16. Add subnormal FPNs to the FPN system in Example 1.14 and we have

1.2 Rounding error analysis

As a crucial difference between \( \mathbb{R} \) and \( \mathcal{F} \), \( \mathbb{R} \) is continuous and infinite while \( \mathcal{F} \) is discrete and finite. To perform computing using \( \mathcal{F} \), we have to represent real numbers by the machine numbers in \( \mathcal{F} \). Rounding is the act of associating a real number with a suitable machine number.

1.2.1 Rounding a single number

Definition 1.15 (Rounding). Rounding is a map \( \text{fl} : \mathbb{R} \to \mathcal{F} \cup \{ \text{NaN} \} \). The default rounding mode is round to nearest, i.e. \( \text{fl}(x) \) is chosen to minimize \( |\text{fl}(x) - x| \) for \( x \in R(\mathcal{F}) \). In the case of a tie, \( \text{fl}(x) \) is chosen by round to even, i.e. \( \text{fl}(x) \) is the one with an even last digit \( d_{p-1} \).

Definition 1.16. A rounded number \( \text{fl}(x) \) overflows if \( |x| > \text{OFL}(\mathcal{F}) \), in which case \( \text{fl}(x) = \text{NaN} \), or underflows if \( 0 < |x| < \text{UFL}(\mathcal{F}) \), in which case \( \text{fl}(x) = 0 \). An underflow of an extended FPN system is called a gradual underflow.

Definition 1.17. The unit roundoff of \( \mathcal{F} \) is the number
\[
\epsilon_u := \frac{1}{2} \epsilon_M = \frac{1}{2} \beta^{1-p}. \tag{1.12}
\]

Remark 1.17. The unit roundoff in (1.12) corresponds to “round to nearest” because for this rounding mode \( \epsilon_u \) is simply the maximum of the minimization problem \( |\text{fl}(x) - x| \) for the normalized significand of all \( x \in R(\mathcal{F}) \). Note that some books do not distinguish machine precision from the unit roundoff; we do in this class.

Lemma 1.18. For rounding to nearest, the unit roundoff for an FPN system with precision \( p + k \) is \( \beta^{p-1-k} \epsilon_u \epsilon_M \).

Proof. According to Definitions 1.15 and 1.17, the unit roundoff for an FPN system with precision \( p + k \) is
\[
\frac{1}{2} \beta^{p-k} = \frac{1}{2} \beta^{1-p} \beta^{1-p-1-k} = \beta^{p-1-k} \epsilon_u \epsilon_M,
\]
where the last step follows from Definitions 1.8 and 1.17. \( \square \)

Theorem 1.19. For \( x \in R(\mathcal{F}) \) as in (1.9), we have
\[
\text{fl}(x) = x(1 + \delta), \quad |\delta| < \epsilon_u. \tag{1.13}
\]

Proof. By Definition 0.16, \( \mathcal{R}(\mathcal{F}) \) is a subset of \( \mathbb{R} \) and is thus a chain. Therefore \( \forall x \in R(\mathcal{F}), \exists x_L, x_R \in \mathcal{F} \) s.t.

- \( x_L \) and \( x_R \) are adjacent,
- \( x_L \leq x \leq x_R \).

If \( x = x_L \) or \( x_R \), then \( \text{fl}(x) = x = 0 \) and (1.13) clearly holds. Otherwise \( x_L < x < x_R \). Then Lemma 1.13 and Definitions 1.12 and 1.15 yield
\[
|\text{fl}(x) - x| \leq \frac{1}{2} |x_R - x_L| \leq \epsilon_u \min(|x_L|, |x_R|) < \epsilon_u |x|. \tag{1.14}
\]
Hence \( -\epsilon_u |x| < \text{fl}(x) - x < \epsilon_u |x| \), which yields (1.13). \( \square \)

Theorem 1.20. For \( x \in R(\mathcal{F}) \), we have
\[
\text{fl}(x) = \frac{x}{1 + \delta}, \quad |\delta| \leq \epsilon_u. \tag{1.15}
\]

Proof. The proof is the same as that of Theorem 1.19, except that we replace the last inequality “\( -\epsilon_u |x| \)” in (1.14) by “\( \leq \epsilon_u |\text{fl}(x)| \)” Consequently, the equality in (1.15) holds when \( x = \frac{1}{2}(x_L + x_R) \) and \( \text{fl}(x) = x_L \) has \( m = 1.0 \). \( \square \)

Remark 1.18. The condition \( x \in R(\mathcal{F}) \) for (1.13) and (1.15) in Theorems 1.19 and 1.20 is necessary: for any positive \( x < \text{UFL}(\mathcal{F}) \), the LHS \( \text{fl}(x) = 0 \) while the RHS is definitely not zero.
Exercise 1.19. Find \( x_L, x_R \) of \( x = \frac{2}{3} \) in normalized single-precision IEEE 754 standard, which of them is \( \text{fl}(x) \)?

Solution. By Exercise 1.7, we have
\[
\begin{align*}
2 &= (0.1010101\cdots)_2 = (1.0101010\cdots)_2 \times 2^{-1}, \\
x_L &= (1.010\cdots10)_2 \times 2^{-1}, \\
x_R &= (1.010\cdots11)_2 \times 2^{-1}. \\
x - x_L &= \frac{2}{3} \times 2^{-24} \text{ and } x_R - x_L = 2^{-24}, \\
x &= (x_R - x_L) - (x - x_L) = \frac{1}{3} \times 2^{-24}.
\end{align*}
\]
Thus \( \text{fl}(x) = x_R. \)

1.2.2 Binary floating-point operations

Remark 1.20. Definitions 1.21, 1.23, and 1.25 in this section model the binary floating-point operations of computer arithmetic. The steps in these definitions may be slightly different for varying hardware architectures. However, we emphasize that Lemmas 1.22, 1.24, 1.26 and Theorem 1.27 are consistent with these Definitions. In other words, we illustrate the model of machine arithmetic using one specific hardware implementation.

Definition 1.21 (Addition/subtraction of two FPNs). Express \( a, b \in \mathcal{F} \) as \( a = M_a \times \beta^e_a \) and \( b = M_b \times \beta^e_b \) where \( M_a = \pm m_a \) and \( M_b = \pm m_b \). With the assumption \( |a| \geq |b| \), the sum \( c := \text{fl}(a + b) \in \mathcal{F} \) is calculated in a register of precision at least \( 2p \) as follows.

(i) Exponent comparison:
- If \( e_a - e_b > p + 1 \), set \( c = a \) and return \( c \);
- otherwise set \( e_c := e_a \) and \( M_c := M_b / \beta^{e_a - e_b} \).

(ii) Perform the addition \( M_c := M_a + M_b \) in the register with rounding to nearest.

(iii) Normalization:
- If \( |M_c| = 0 \), return 0.
- If \( |M_c| \geq \beta \), set \( M_c := M_c / \beta \) and \( e_c := e_c + 1 \).
- If \( |M_c| \in (0, 1) \), repeat \( M_c := M_c / \beta, e_c := e_c - 1 \) until \( |M_c| \in [1, \beta) \).

(iv) Check range:
- return NaN if \( e_c \) overflows,
- return 0 if \( e_c \) underflows.

(v) Round \( M_c \) (to nearest) to precision \( p \).

(vi) Set \( c := M_c \times \beta^{e_c} \).

Example 1.21. Consider the calculation of \( c := \text{fl}(a + b) \) with \( a = 1.234 \times 10^4 \) and \( b = 5.678 \times 10^6 \) in an FPN system \( \mathcal{F} : (10, 4, -7, 8) \).

(i) \( b \leftarrow 0.0005678 \times 10^9; e_c \leftarrow 4 \).

(ii) \( m_c \leftarrow 1.2345678 \).

(iii) do nothing.

(iv) do nothing.

(v) \( m_c \leftarrow 1.235 \).

(vi) \( c = 1.235 \times 10^4 \).

For \( b = 5.678 \times 10^{-2}, c = a \) would be returned in step (i).

Remark 1.22. For \( b = 5.678 \times 10^{-1} \), the first case in step (i) would not be invoked. Then the result would be the same as that of returning \( c = a \) in step (i). However, this does not mean that we can replace the condition \( e_a - e_b > p + 1 \) with \( "e_a - e_b > p". \) As the main reason, adding two normalized positive FPNs always yields a significand no less than 1.0 in step (ii) while subtracting two such FPNs might yield a significand less than 1.0. Therefore we need an additional guard digit to ensure the correctness.

Example 1.23. Consider the calculation of \( c := \text{fl}(a + b) \) with \( a = 1.000 \times 10^9 \) and \( b = -9.000 \times 10^{-5} \) in an FPN system \( \mathcal{F} : (10, 4, -7, 8) \).

(i) \( b \leftarrow -0.0000900 \times 10^6; e_c \leftarrow 0 \).

(ii) \( m_c \leftarrow 0.9999100 \).

(iii) \( e_c \leftarrow e_c - 1; m_c \leftarrow 9.9991000 \).

(iv) do nothing.

(v) \( m_c \leftarrow 9.999 \).

(vi) \( c = 9.999 \times 10^{-1} \).

For \( b = -9.000 \times 10^{-6} \), \( c = a \) would be returned in step (i).

Exercise 1.24. Repeat Example 1.21 with \( b = 8.769 \times 10^4, b = -5.678 \times 10^6, \) and \( b = 5.678 \times 10^8 \).

Lemma 1.22. For \( a, b \in \mathcal{F}, a + b \in \mathcal{R}(\mathcal{F}) \) implies
\[
\text{fl}(a + b) = (a + b)(1 + \delta), \quad |\delta| \leq e_a.
\]

Proof. The round-off error in step (v) always dominates that in step (ii), which is nonzero only in the case of \( e_a - e_b = p + 1 \). Then (1.16) follows from Theorem 1.19.

Definition 1.23 (Multiplication of two FPNs). Express \( a, b \in \mathcal{F} \) as \( a = M_a \times \beta^{e_a} \) and \( b = M_b \times \beta^{e_b} \) where \( M_a = \pm m_a \) and \( M_b = \pm m_b \). The product \( c := \text{fl}(ab) \in \mathcal{F} \) is calculated in a register of precision at least \( p + 2 \) as follows.

(i) Exponent sum: \( e_c \leftarrow e_a + e_b \).

(ii) Perform the multiplication \( M_c \leftarrow M_a M_b \) in the register with rounding to nearest.

(iii) Normalization:
- If \( |M_c| \geq \beta \), set \( M_c := M_c / \beta \) and \( e_c := e_c + 1 \).

(iv) Check range:
- return NaN if \( e_c \) overflows,
- return 0 if \( e_c \) underflows.

(v) Round \( M_c \) (to nearest) to precision \( p \).

(vi) Set \( c := M_c \times \beta^{e_c} \).
Example 1.25. Consider the calculation of $c := \text{fl}(ab)$ with $a = 2.345 \times 10^4$ and $b = 6.789 \times 10^9$ in an FPN system $\mathcal{F} : (10, 4, -7, 8)$.

(i) $e_c \leftarrow 4$.
(ii) $M_c \leftarrow 15.9202$.
(iii) $m_c \leftarrow 1.59202, e_c \leftarrow 5$.
(iv) do nothing.
(v) $m_c \leftarrow 1.592$.
(vi) $c = 1.592 \times 10^5$.

Remark 1.26. For correct results of rounding to nearest, it would not be enough to use a register of precision $p + 1$ in Definition 1.23. Below is a counter-example. Suppose the precision of the register were $p + 1$ and $M_a M_b = 1.59346$ in Example 1.25. Then $M_a$ would be rounded to 1.5935 in step (ii) and to 1.594 in step (v). But this will be an incorrect result for the rounding mode of rounding to nearest.

Lemma 1.24. For $a, b \in \mathcal{F}$, $|ab| \in \mathcal{R}(\mathcal{F})$ implies

$$\text{fl}(ab) = (ab)(1 + \delta), \quad |\delta| \leq \varepsilon_u.$$  

Proof. The error only come from the round-off in steps (ii) and (v). Then (1.17) follows from Theorem 1.19. \hfill \Box

Definition 1.25 (Division of two FPNs). Express $a, b \in \mathcal{F}$ as $a = M_a \times \beta^{e_a}$ and $b = M_b \times \beta^{e_b}$ where $M_a = \pm m_a$ and $M_b = \pm m_b$. The product $c = \text{fl} \left( \frac{a}{b} \right) \in \mathcal{F}$ is calculated in a register of precision at least $2p + 1$ as follows.

(i) If $m_b = 0$, return NaN; otherwise set $e_c \leftarrow e_a - e_b$.
(ii) Perform the division $M_c \leftarrow M_a/M_b$ in the register with rounding mode of rounding to nearest.
(iii) Normalization:

- If $|M_c| < 1$, set $M_c \leftarrow M_c \beta$, $e_c \leftarrow e_c - 1$.
(iv) Check range:

- return NaN if $e_c$ overflows,
- return 0 if $e_c$ underflows.
(v) Round $M_c$ (to nearest) to precision $p$.
(vi) Set $c \leftarrow M_c \times \beta^{e_c}$.

Lemma 1.26. For $a, b \in \mathcal{F}$, $\frac{a}{b} \in \mathcal{R}(\mathcal{F})$ implies

$$\text{fl} \left( \frac{a}{b} \right) = \frac{a}{b}(1 + \delta), \quad |\delta| < \varepsilon_u.$$  

Proof. In the case of $|M_a| = |M_b|$, there is no rounding error in Definition 1.25 and (1.18) clearly holds. Hereafter we denote by $M_{L1}$ and $M_{L2}$ the results of steps (ii) and (v) in Definition 1.25, respectively.

In the case of $|M_a| > |M_b|$, the condition $a, b \in \mathcal{F}$, Definition 1.8, and $|M_a|, |M_b| \in [1, \beta)$ imply

$$\left| \frac{M_a}{M_b} \right| \geq \frac{\beta - \varepsilon_M}{\beta - 2\varepsilon_M} > 1 + \beta^{-1} \varepsilon_M, \quad (1.19)$$

which further implies that the normalization step (iii) in Definition 1.25 is not invoked. By Lemma 1.18, the unit roundoff for the register is $\beta^{-2} \varepsilon_u \varepsilon_M$. Therefore we have

$$M_{L1} = M_{L1} + \delta_2, \quad |\delta_2| \leq \varepsilon_u$$

$$= \frac{M_a}{M_b} + \delta_1 + \delta_2, \quad |\delta_1| \leq \beta^{-2} \varepsilon_u \varepsilon_M$$

$$= \frac{M_a}{M_b}(1 + \delta);$$

$$|\delta| = \frac{\delta_1 + \delta_2}{M_a/M_b} \leq \varepsilon_u \left( 1 + \beta^{-2} \varepsilon_M \right) < \frac{\varepsilon_u}{1 + \beta^{-1} \varepsilon_M} < \varepsilon_u,$$

where we have applied (1.19) and the triangular inequality in deriving the first inequality of the last line.

Consider the last case $|M_a| < |M_b|$. It is impossible to have $|M_{L1}| = 1$ in step (ii) because

$$\frac{|M_a|}{|M_b|} \leq \frac{\beta - 2\varepsilon_M}{\beta - \varepsilon_M} = 1 - \frac{\varepsilon_M}{\beta - \varepsilon_M} < 1 - \beta^{-1} \varepsilon_M$$

and the precision of the register is greater than $p + 1$. Therefore $|M_{L1}| < 1$ must hold and in Definition 1.25 step (iii) is invoked to yield

$$M_{L2} = \frac{M_a}{M_b} + \delta_1, \quad |\delta_1| \leq \beta^{-2} \varepsilon_u \varepsilon_M;$$

$$M_{L2} = \beta \cdot M_{L1} + \delta_2, \quad |\delta_2| \leq \varepsilon_u$$

$$= \frac{\beta M_a}{M_b} \left( 1 + \frac{\beta \delta_1 + \delta_2}{\beta M_a/M_b} \right),$$

where the denominator in the parentheses satisfies

$$\beta \left| \frac{M_a}{M_b} \right| \geq \frac{\beta}{\beta - \varepsilon_M} > 1 + \beta^{-1} \varepsilon_M.$$

Hence we have

$$|\delta| = \frac{\beta \delta_1 + \delta_2}{\beta M_a/M_b} \leq \frac{\beta^{-1} \varepsilon_u \varepsilon_M + \varepsilon_u}{1 + \beta^{-1} \varepsilon_M} = \varepsilon_u. \quad \Box$$

Remark 1.27. The precision requirements of the register in Definitions 1.21, 1.23, and 1.25 are not only sufficient but also necessary for Lemmas 1.22, 1.24, and 1.26. In other words, one can show by counter-examples that the conclusions in these lemmas does not hold if the register does not have enough significant digits.

Theorem 1.27 (Model of machine arithmetic). Denote by $\mathcal{F}$ a normalized FPN system with precision $p$. For each arithmetic operation $\odot = +, -, \times, /$, we have

$$\forall a, b \in \mathcal{F}, a \odot b \in \mathcal{R}(\mathcal{F}) \Rightarrow \text{fl}(a \odot b) = (a \odot b)(1 + \delta) \quad (1.20)$$

where $|\delta| \leq \varepsilon_u$ if and only if these binary operations are performed in a register with precision $2p + 1$.

Proof. This follows from Lemmas 1.22, 1.24, and 1.26. \hfill \Box

Remark 1.28. The model of machine arithmetic states that the relative round-off error is less than the unit roundoff upon some reasonable assumptions. It is emphasized that in the model of machine arithmetic $a, b$ are machine numbers in $\mathcal{F}$. So this model should not be confused by the catastrophic cancellation of subtracting two numbers $x, y \in \mathbb{R}$ that are not exactly represented by machine numbers.