Bit Sum Problem KPS

No Author Given

No Institute Given

The system for adding two bits is defined as follows:

$$\mathrm{sk}\Pi_{\mathrm{Add}}(n) = (A, L, IO, \mu, C_1, C_2, C_3),$$

where:

- $A = \{a_0, a_1, \ldots, a_{21}\}$ is the alphabet.
- $L = \{1, 2, 3\}$ is the label set for compartments.
- IO: No external input/output symbols are required, as the result is stored in compartment C_3.
- $\mu = [[[\ ]_3]_2]_1$ is the membrane structure.
- Compartments:
  - $C_1 = (1, w_{1,0}, R_1)$, where $w_{1,0} = \{a_x, a_y\}$ contains the input bits.
  - $C_2 = (2, w_{2,0}, R_2)$, where $w_{2,0} = \{\}$ serves as an intermediary workspace.
  - $C_3 = (3, w_{3,0}, R_3)$, where $w_{3,0} = \{\}$ stores the final result.

Rules for Each Compartment

Rules in C_1 (Initialization and Data Movement)

$$r_{1,1} : a_x \to (a_x, 2), \quad a_y \to (a_y, 2).$$

These rules send the input bits a_x, a_y to C_2.

Rules in C_2 (Addition Logic) For every a_i, the addition rules are:

$$r_{2,i} : 2a_i \to a_{i+1}(3), \quad \forall i \in \{0, 1, \ldots, 20\}.$$

These rules consume two identical input symbols and produce the next symbol, which is sent to C_3.

Rules in C_3 (Result Storage)

$$r_{3,1} : a_i \to a_i, \quad \forall i \in \{0, 1, \ldots, 21\}.$$

This rule simply stores the result.

Step-by-Step Process

1. **Initialization:** Input bits a_x, a_y are placed in C_1. Rules $r_{1,1}$ transfer the bits to C_2.
2. **Addition:** In C_2, rules $r_{2,i}$ perform the addition. Two symbols a_x and a_y are consumed to produce the result a_{x+y}, which is sent to C_3.
3. **Result:** C_3 contains the final result a_{x+y}, which represents the sum of the two bits.

References