A collision-based computer is an architecture-less device, where signals are represented by particles (gliders or self-localizations), and logical operations are calculated in collisions between particles. We detect and classify all possible collisions between particles in one-dimensional cellular automata, and construct a catalogue of binary collisions. Therefore, we develop formal languages via de Bruijn diagrams to represent particles in evolution space of Rule 54 and Rule 110. We use de Bruijn diagrams and tiles to determine the base of the subset of regular expressions that identify each well-known particle of Rule 54 and Rule 110. An important indication is that the subset of regular expressions describing all particles does not include extensions or packages of them. The tool offer a simple procedure to handle collisions between gliders with the OSXLCAU21 system. In this way, we have solved some problems like self-organization of particles, production of glider guns, solitons, big triangles, reproduction of cyclic tag systems in Rule 110 and logical gates in Rule 54.