DNA-HASKELL represents a model for DNA computing whose operations were implemented in the laboratory and contributed to the successful solution of a NP-complete problem. Both, the description of NP-complete problem solving DNA algorithms and the simulation of computational complete universal models is possible with DNA-HASKELL. The model is also able to include the description of another existing algorithmic implementations reducing some side effects because of its closeness to the laboratory, so to say, these effects belong to the definitions of the operations. Beyond DNA-HASKELL is suitable for description of established mathematical models for DNA computing, it fills the gap between models with a high abstraction level and practical implementations in the laboratory. The concept of DNA-HASKELL arose by direct observations of molecular processes specifying the according functions and forming the operational semantics of DNA-HASKELL. The computational completeness of DNA-HASKELL can be assumed by simulation of Turing machines and distributed splicing systems for recursive enumerable languages.

Splicing operation in DNA-HASKELL

The splicing operation forms the core at all types of splicing systems and embeds an abstract formal simulation of DNA recombinant techniques with cut restriction enzymes (digestion) and ligate. It is based on elements of mostly infinite sets that express DNA strands, named words of formal languages. The description of the splicing operation on words or formal languages also leads to a generalization of the effect that is caused by digestion and ligation. The generalization supposes certain DNA strands resp. words that can really additional occur during the ligation process as side effects. Integration propose a sequence of DNA-HASKELL operations that simulate the splicing operation on linear data structures defined by a splicing system as above (using functional DNA-HASKELL syntax and flowchart).

Algorithm in DNA-HASKELL

The DNA algorithm for solution to the knapsack problem produces all nonempty combinations of the input DNA double strands (5'-phosphorylated, end compatible) encoding \( w_i \) at most once and represents a possible knapsack weight. Starting from a “Starter” fragment (one side compatible, other side 5'-phosphorylated), the combinations are generated by consecutively processing operational loops. Each loop embodies a split and combine strategy, adding a new \( w_i \) and doubling the number of combinations. The DNA pool is split into two hosts. One half is 5'-phosphorylated and merged with a new \( w_i \) and with the other half. A subsequent ligation produces new combinations. After including all \( w_i \), a terminating fragment is added to all combinations. A subsequent PCR with starter and terminator sequences as primers extracts all valid combinations. Whether or not a combination of the length is occoured and consequently ligated by gel electrophoresis. Figure B shows the result of a simplified example without starter because of appropriate input strand lengths.