The handle [http://hdl.handle.net/1887/37052](http://hdl.handle.net/1887/37052) holds various files of this Leiden University dissertation.

**Author:** Vliet, Rudy van  
**Title:** DNA expressions : a formal notation for DNA  
**Issue Date:** 2015-12-10
Chapter 12

Conclusions and Directions for Future Research

In this thesis, we have introduced DNA expressions as a formal notation for DNA molecules that may contain nicks and gaps. However, there are (formal) DNA molecules that cannot be represented by our expressions, the ones with nicks in both strands. For each expressible formal DNA molecule, there are infinitely many DNA expressions denoting it. We have rigorously analysed the ones with minimal length, the minimal DNA expressions.

For this, we first derived lower bounds on the length of a DNA expression denoting a given formal DNA molecule. We subsequently described how to construct DNA expressions that achieve the lower bounds, and thus are minimal. We also proved that there do not exist minimal DNA expressions other than those obtained from the constructions. Minimal DNA expressions are characterized by six syntactic properties, which can easily be verified. This can be used to decide whether or not a given DNA expression is minimal.

As a combinatorial intermezzo, we determined the number of minimal DNA expressions denoting a given molecule. For almost all types of expressible formal DNA molecules, the number of minimal DNA expressions can be expressed in terms of the Catalan numbers.

We then described a recursive algorithm, which rewrites a given DNA expression into an equivalent, minimal DNA expression. This is useful, e.g., to save space for storing a description of the DNA molecule denoted. In the algorithm, step by step, the DNA expression acquires the six properties that characterize minimal DNA expressions.

We finally introduced a (minimal) normal form for DNA expressions: a well-defined set of properties such that for each expressible formal DNA molecule $X$, there is a unique DNA expression denoting $X$ and satisfying those properties. Or actually, we defined the normal form DNA expressions as specific minimal DNA expressions, and we proved that these DNA expressions are characterized by five syntactic properties. We described a two-step algorithm, which computes the normal form version of a given DNA expression. This can be used, e.g., to decide if two DNA expressions are equivalent. The algorithm first rewrites its input into an equivalent, minimal DNA expression, using the recursive algorithm for minimality mentioned above. After that, it performs some additional rewriting steps to acquire the (remaining) properties of the minimal normal form.

Both the algorithm for minimality and the algorithm for the minimal normal form are elegant, because they do not refer to the semantics of the DNA expression involved. They consist of (local) string manipulations on the DNA expression itself. We proved that both algorithms are correct (they do what they are supposed to do), and that they require linear time and space.
Although the analysis of DNA expressions in this thesis is quite elaborate, one could think of some more aspects to examine. For example, one might consider the detection of submolecules: given two DNA expressions $E_1$ and $E_2$, is $S(E_1)$ a formal DNA submolecule of $S(E_2)$? Of course, this can easily be decided by computing $S(E_1)$ and $S(E_2)$, but can it also be done by local rearrangements at the level of the DNA expressions? Perhaps, the minimal normal form may be helpful for this.

The formal DNA molecules, which form the semantic basis of our notation, are a formalization of the double-word notation for DNA molecules. As mentioned in Section 2.2, a rotation of a double word by an angle of 180° yields another double word representing the same DNA molecule (see Figure 2.10). As it is, the two representations of the molecule have their own DNA expression in minimal normal form. One could argue that only one of the two should be the ‘true’ normal form representation of the molecule. In that case, the definition of the normal form should be adjusted.

An idea we have not worked out in detail in this thesis, is the alternative implementation of the recursive function MakeMinimalNF. We only described it in global terms at the end of Section 11.1. In the process of rewriting a given DNA expression into the minimal normal form, this implementation maintains two DNA expressions, which are (based on) operator-minimal $\uparrow$-expressions and operator-minimal $\downarrow$-expressions, respectively. It would be interesting to work out the details of this implementation, and verify that its complexity is linear, as opposed to that of a natural implementation of MakeMinimalNF.

As we have remarked in Section 4.1, the set of operators $\{\uparrow, \downarrow, \leftrightarrow\}$ that we consider is one of many possible choices. It could be an important research line for the future to investigate other notations for DNA molecules.

One could think of an extension of the current notation, such that all formal DNA molecules (also the ones with nicks in both strands) can be denoted. Perhaps, this can be achieved by a ‘guard-operator’, which protects a DNA submolecule from the effects of other operators. In particular, it may prevent a nick in one strand from being sealed by an operator introducing nicks in the other strand.

Another extension could be an operator that makes two formal DNA molecules with complementary sticky ends anneal. It would also be desirable to define operators that make it possible to denote DNA molecules with a variety of other ‘imperfections’ than nicks and gaps, such as, e.g., hairpin loops and circular strands, see Section 2.2.

Rather than extending the present notation, one could also consider to start with a completely new set of operators. As mentioned in Section 1.2, a possible motivation for considering formal notations for DNA molecules is to provide a formal calculus for the processing of DNA molecules. Such a calculus would have to contain operators that correspond to various biochemical operations on DNA molecules, as well as expressions for sorts of DNA molecules resulting from applications of these operators. This way, the expressions not only denote DNA molecules, but they also implicitly describe how to synthesize them from the basic elements A, C, G and T.

From the mathematical point of view, the set of operators acting on expressions denoting DNA molecules does not have to correspond exactly to the biochemical operations. However, one should be able to express such operations by suitable compositions of mathematical operations.

The DNA expressions from this thesis can be considered as a first step towards a formal calculus for DNA processing, including descriptions for more complex DNA molecules. To finally achieve that goal, there are many more steps to be taken.