07 - Bioinformatik

The lecture about bioinformatics given by Ph.D. Henning Bordihn concentrated on two main topics: Lindenmayer systems and DNA computing. The former are a mathematical formalism similar to formal grammars. Simple parts of an object get recursively substituted by production rules to eventually form a more complex object. Developed by A. Lindenmayer in the 1960s, Lindenmayer Systems are used to simulate the growth and cell division of simple multicellular organisms (e.g. the moss phascum cupsidatum) and to create fractals and even realistic plant models in computer graphics.

To be more specific, a L-system without interaction is a triple which consists of a finite alphabet, a set of substitution rules and a nonempty string to begin with. For every character in the alphabet there is a rule specifying a string which replaces the character. If there is more than one rule for each character, the L-system is called non-deterministic. Accordingly, it is called deterministic if there is only one such rule. After the first step the characters of the start string are all replaced by strings whose characters get replaced in subsequent steps. This way it is possible that the system grows to infinite size which is measured by a growth function returning the length of the string after a given number of steps.

To sum things up, L-systems are a very interesting concept to describe phenomena in nature and computer science. That is possibly why several years ago every decent man did something with L-systems.

The second topic of the lecture was DNA computing which uses DNA and methods of biochemistry instead of the common silicon based computer technologies for computation. The DNA consists of two base pairs which themselves consist of four complementary bases: adenine, thymine, cytosine and guanine. These base pairs make it possible to store and manipulate information in the DNA. Bimolecular techniques such as polymerase chain reaction or gel electrophoresis are used to duplicate DNA, split double-stranded DNA into single-stranded and to filter certain strands out. In the 1990s, L. Adleman demonstrated a proof-of-concept use of DNA-computing by solving the NP complete Hamiltonian path problem only by biochemical means. He created a huge number of DNA strands representing all possible paths through a pre-chosen graph and filtered out all paths which did not have the correct node number, start node and end node. At last, for every node of the graph, he filtered out all paths not containing this node. The DNA strands left represent a solution to the Hamiltonian path problem. If there are no strands, there shouldn't be a solution. The advantage of this form of computation is that it uses the many different DNA strands to try many different possibilities at once being very similar to parallel computing. Another advantage is that a little amount of DNA can store huge amounts of data in terabyte regions.

Although there are these advantages and Adleman's experiment proofed that DNA computing is possible, it is far away from practical and economical usage. The processes Adleman had to carry out in the experiment are very complex and took much time even though the "computations" in the test-tube were fast. It is widely believed that DNA computers won't replace our silicon based computers because the manipulation of DNA molecules is impractical and not fast enough for our modern computation needs.