Mathematisches Institüt der Universität Bonn
winter semester 2010-11
Hand in at 28.10.2010

Higher set theory

Formal derivations and natural proofs exercise sheet 2

1. Write a grammar that represents arithmetic expressions without superfluous brackets (i.e. expressions like $(a+b)\cdot(b-c)$).

> *Deterministic finite automata:*
> A deterministic finite automaton M is a 5-tuple $(Z, \Sigma, \delta, z_0, E)$, where Z is a finite set called the set of states, Σ is a finite set of input symbols called the alphabet ($Z \cap \Sigma = \emptyset$), $z_0 \in Z$ is called the start state, E is a finite subset of Z called the set of accepted states or end states, and $\delta : Z \times \Sigma \to Z$ is called the transition function.
>
> For this M we define a function $\hat{\delta} : Z \times \Sigma^* \to Z$ inductively as follows: $\hat{\delta}(z, \epsilon) = z$, and $\hat{\delta}(z, ax) = \hat{\delta}(\delta(z, a), x)$, where $z \in Z$, $x \in \Sigma^*$ (the set of all finite strings of elements of Σ), and $a \in \Sigma$. We say that the language that M accepts is the set $T(M) := \{x \in \Sigma^* \, ; \, \hat{\delta}(z_0, x) \in E\}$.

2. Write a deterministic finite automaton to accept all strings in the alphabet $\{0, 1\}$ which do not contain three consecutive ones.

3. Write a Turing machine with the alphabet $\{0, 1\}$ that transforms an input consisting of k consecutive 1's to an input that consists of 2k consecutive 1's.

Extra points:
Given a deterministic finite automaton, write a Turing machine that simulates it.