In this talk I'll survey some results we recently obtained with an automatic prover written by my student Hamoon Mousavi. This software package, written in Java, takes as input an automaton $M$ specifying an infinite word $x$ and a predicate $P(n)$ (expressed in a logical language) about the factors of $x$, and returns an automaton accepting the representation of those $n$ for which the predicate holds. Depending on the size of the initial automaton $M$, the type of representation (base-$k$ for $k \geq 2$; Fibonacci representation; Tribonacci representation), and the complexity of the predicate, many nontrivial assertions can be proven mechanically in a few seconds or minutes.

As an example, we used this prover to show the existence of an aperiodic infinite binary word avoiding the pattern $xxx^R$ and infinitely many distinct primitive words $t$ such that $t^\omega = ttt \cdots$ avoids the same pattern.

This is joint work with Chen Fei Du, Hamoon Mousavi, and Luke Schaeffer. (Received July 31, 2014)