Algorithms Reading Group March 17, 2006

The DJ's Problem

Lecturer: Steve Skiena *Scribe: Valentin Polishchuk*

A *song* is a word over the alphabet $\{0, 1\}$. A *record* is also a word over the alphabet $\{0, 1\}$. We view a record $\mathcal{R} = r_1, \ldots, r_R$ as an undirected graph (a path) $G_\mathcal{R} = (V, E)$ with $V = \{0 \ldots R\}$ and $E = \{(i-1, i), i = 1 \ldots R\}$; the edge $(i-1, i) \in E$ has label r_i. Every walk $\mathcal{W} = w_0, w_1 \ldots, w_W$ in $G_\mathcal{R}$ defines a song $\mathcal{S} = s_1, \ldots, s_W$ with s_i, $i = 1 \ldots W$, being the label of the edge (w_{i-1}, w_i). By the analogy with what (we believe) a hip-hop DJ does we say that in this case the song $\mathcal{S}$ is *played* by the walk $\mathcal{W}$. We say that a song $\mathcal{S}$ *can be played* by a record $\mathcal{R}$ if there exists a walk $\mathcal{W}$ in $G_\mathcal{R}$ that starts from 0 and is such that $\mathcal{S}$ is played by $\mathcal{W}$.

The problem we considered is as follows:

> Given two songs $\mathcal{S} = s_1, \ldots, s_S$ and $\mathcal{T} = t_1, \ldots, t_T$ is there a record $\mathcal{R} = r_1, \ldots, r_R$ which can play both of them?

We call the songs from a "Yes" instance of the problem *compatible*.

We made the following observations:

1. The problem is related to finite-state automata.
2. It is important that the walk starts at 0, o.w. both songs can be trivially played by $\mathcal{R} = \mathcal{S}|\mathcal{T}$, where $|$ denotes string concatenation.
3. If R, the size of $\mathcal{R}$, is bounded by a number K, we can enumerate all records in time $O(2^K)$ and check for each whether each of the songs can be played by it (see notes by Janet Braunstein of November 12, 2004).
4. Obviously, a "Yes" instance of the problem has $s_1 = t_1 = r_1$, i.e., $\mathcal{S}$ and $\mathcal{T}$ have a common prefix of non-zero length. We started thinking in the direction of using that prefix to construct $\mathcal{R}$.
5. The problem might actually become easier if the record has to end at the end of each song, i.e., if it is required that if $\mathcal{W}_\mathcal{S}$ and $\mathcal{W}_\mathcal{T}$ are the walks in $G_\mathcal{R}$ that play $\mathcal{S}$ and $\mathcal{T}$, then both walks end at R. In this case $\mathcal{S}$ and $\mathcal{T}$ have also a non-empty common suffix.

Maybe, in this case, the relation can play is transitive? Also, Rob Jonson suggested that in this case for any song $\mathcal{S}$ there exists a record $\mathcal{R}$ that can play $\mathcal{S}$ and is *most powerful* in the following sense: for any record $\mathcal{R}'$ which can play $\mathcal{S}$ and any song $\mathcal{S}'$ that can be played by $\mathcal{R}'$ it is true that $\mathcal{R}$ can play $\mathcal{S}'$. Rob suggested to prove it by looking at how $\mathcal{R}'$ plays $\mathcal{S}'$ and "un-fold" the retractions of the walk (so that the walk is simple?) to get a "more powerful" record.

6. Several attempts to solve the problem with dynamic programming were made.

7. It cannot hurt to represent the songs in *run-length encoding*, in which a run of m consecutive symbols $s \in \{0, 1\}$, $\underbrace{s \dots s}_{m \text{ times}}$, is denoted by s^m. For instance, if $\mathcal{S} = 0^{a_1}1^{a_2}0^{a_3}\dots$, then (by the "folding" argument, see the notes by Janet Braunstein)

$$\mathcal{R} = 0^{(2-(a_1 \mod 2))}1^{(2-(a_2 \mod 2))}0^{(2-(a_3 \mod 2))}\dots$$

can play $\mathcal{S}$. In particular, it means that if the run-length encoding of $\mathcal{S}$ is a substring of the run-length encoding of $\mathcal{T}$ (up to the parity of the powers), than $\mathcal{S}$ and $\mathcal{T}$ are compatible. The converse is not true:

Example 1. 0001000 *and* 011100*.* 010 *can play both.*

8. It is not true that one of the compatible songs can play the other; see the above example.

9. If two songs are compatible, there exists a record that plays both and is such that there exists no substring of the form ww^Tw in the record (see the notes by Janet Braunstein).