PS1 - Part 2

Joe Brown and Morgan Marks

January 29, 2014

0 : SHORTEST SUPERSTRING

0.1 PROVIDING A 2-APPROXIMATION FOR GREEDY

There is a family of strings for which lucky greedy gets "locked out," causing a $\frac{3}{2}$ OPT solution. The fundamental building blocks of this family are:

- L : The **L**eft-side lockout character
- R : The **R**ight-side lockout character
- S : The **S**eparator character
- O : The repeated character.

To generate the simplest version of the lockout, we use three strings:

$$\begin{aligned} &LO^k \\ &O^kR \\ &O^{k-1}SO^{k-1} \end{aligned}$$

$$\begin{aligned} Greedy: &\quad LO^kRO^kSO^{k-1} \\ OPT: &\quad LO^kSO^kR \end{aligned}$$

Greedy chooses to merge the first two strings, because they have a value of k, then ends up with the middle string locked out. OPT instead chooses to sandwich the third string.
To further hinder Greedy, we have to add another layer of abstraction. In essence, we multipy the effect of the lockout by multiplying the O^kS portion of the strings.

$$
\begin{array}{l}
L(O^kS)^nO^k \\
(O^kS)^nO^kR \\
O^{k-1}S(O^kS)^{n-1}O^{k-1}
\end{array}
$$

$$
\begin{array}{rl}
Greedy: & L(O^kS)^nO^kRO^{k-1}S(O^kS)^{n-1}O^{k-1} \\
OPT: & L(O^kS)^{n+1}O^kR
\end{array}
$$

As n becomes very large, Greedy is a 2·OPT approximation. □

0.2 Exceeding a 2-approximation for Greedy

In all of our sketching and stretching, papers spilling from the couch and the coffee table to the floor, we are left ragged and empty-handed. The amazement at our complete lack of progress resonates in the empty $\frac{11}{23}$, and we are left with the solemn question of how the *(redacted)* can this upper bound of two be so hard to prove?

0.3 Alphabet Restriction Contradiction

The general Shortest Superstring problem (for any alphabet size) can be α-approximated in polynomial time.
Given$_1$: ∃ polynomial-time α-approximation for Shortest Superstring *for instances over a binary alphabet.* Given$_2$: Any alphabet Σ can be binary encoded with the following formula:

replace the i_{th} character in Σ with $0^i(01)^{k+1-i}1^i$, where k := |Σ|.

By using this encoding to produce Σ', no distinct character in Σ' can overlap at all with any other distinct character in Σ'
The idea shares a few traits of the fundamental building blocks we use in 0.1

- 0*… : The Left-side lockout character
- …1* : The Right-side lockout character
- 01* : The Separator/Uniqueness character

By locking out overlap by anything but identical characters, we are effectively running a multicharacter alphabet Σ emulator in Σ'. Since we know that the algorithm is an α-approximation in (0,1)*, we know that this emulation is also α-approximable. □

1 : O(1)-APPROXIMATION FOR NUMBER OF MOVES IN CHUNK SORTING