CS 462: Winter 2019
Three Definitions of the Shuffle Operation

January 23, 2020

The shuffle of two words (and two languages) is defined in the course text, page 57.

Here are three different definitions of this concept. A good exercise is to convince yourself that all three define the same opertion.

We assume here that all words are in Σ^* and all languages are subsets of Σ^*.

Definition 1

$$\begin{aligned}\text{shuff}(x,y) = \{z \in \Sigma^* \ : \ &\exists n \geq 1 \text{ and words } x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n \in \Sigma^* \text{ such that} \\ &z = x_1 y_1 x_2 y_2 \cdots x_n y_n \text{ and } x = x_1 x_2 \cdots x_n \text{ and } y = y_1 y_2 \cdots y_n\}.\end{aligned}$$

This definition says, in English, that z is formed by splitting x and y into n pieces (of arbitrary size, not necessarily the same) and then "interleaving" them, as in an imperfect shuffle of a card deck. Notice that the pieces can actually be empty.

This is the definition of shuffle of two *words*. It can be extended to the definition of the shuffle of two *languages* as follows:

$$\text{shuff}(L_1, L_2) = \bigcup_{x \in L_1, \ y \in L_2} \text{shuff}(x,y).$$

Or it can be directly modified to get

$$\begin{aligned}\text{shuff}(L_1,L_2) = \{z \in \Sigma^* \ : \ &\exists n \geq 1 \text{ and words } x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n \in \Sigma^* \text{ such that} \\ &z = x_1 y_1 x_2 y_2 \cdots x_n y_n \text{ and } x_1 x_2 \cdots x_n \in L_1 \text{ and } y_1 y_2 \cdots y_n \in L_2\}.\end{aligned}$$

One nice thing about this definition is that it is clear right away that $\text{shuff}(L_1, L_2) = \text{shuff}(L_2, L_1)$.

Definition 2

Some students may prefer a recursive definition of shuffle. Here is one. This is a definition for words, which can then be extended to languages as above.

$$\text{shuff}(x,y) = \begin{cases} xy, & \text{if } x = \epsilon \text{ or } y = \epsilon; \\ a\,\text{shuff}(x',y) \;\cup\; b\,\text{shuff}(x,y'), & \text{if } x, y \text{ are both nonempty, and there exist } a, b \in \Sigma \text{ and } x', y' \in \Sigma^* \text{ such that } x = ax' \text{ and } y = by' \ . \end{cases}$$

One nice thing about this definition is that it gives an explicit recursive algorithm for computing the shuffle of two words (although it is not terribly efficient).

Using either of the previous two definitions, we can construct an NFA for $\text{shuff}(L_1, L_2)$, given DFA's for L_1 and L_2. Suppose these DFA's are $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$, respectively. Our NFA is defined to be $M = (Q, \Sigma, \delta, q_0, F)$ as follows:

$$\begin{aligned} Q &= Q_1 \times Q_2 \\ q_0 &= [q_1, q_2] \\ \delta([p,q],a) &= \{[\delta(p,a),q], [p,\delta(q,a)]\} \text{ for all } p \in Q_1,\ q \in Q_2,\ a \in \Sigma\ ; \\ F &= F_1 \times F_2. \end{aligned}$$

The idea is that we allow the next letter to be read from either a string from L_1 or a string from L_2, and accept if we hit an accepting state in both automata.

Using this we see that the state complexity of shuffle, for regular languages of m and n states, respectively, is bounded above by 2^{mn}. This can be improved a tiny bit — see https://arxiv.org/abs/1512.01187 — but the optimal result for the state complexity of shuffle is still not known!

Definition 3

Finally, the course text, Example 3.3.8, pp. 57–58, gives yet another way to define the shuffle. The advantage to *this* definition is that it expresses the shuffle in terms of operations *known to preserve regularity*. So once we see that this definition is the same as one of the two previous, the theorem that the shuffle of two regular languages is regular follows immediately!

Given an alphabet Σ, define the related alphabet $\Sigma' = \{a' \ : \ a \in \Sigma\}$. Then define morphisms h, h_1, h_2 as follows, for each $a \in \Sigma$:

$$\begin{aligned} h(a) &= h(a') = a \\ h_1(a) &= a \\ h_1(a') &= \epsilon \\ h_2(a) &= \epsilon \\ h_2(a') &= a \end{aligned}$$

Then

$$\text{shuff}(L_1, L_2) = h(h_1^{-1}(L_1) \ \cap \ h_2^{-1}(L_2)).$$

Do you see why?

Exercise. Which of the three definitions would be easiest to modify to define the shuffle of three words (or languages)?