For languages A and B, let the perfect shuffle of A and B be the language: PerfectShuffle(A, B) = \{ w | w = a_1 b_1 ... a_k b_k, where a_1...a_k is in A and b_1...b_k is in B with each a_i, b_i in \Sigma \}.

Prove that regular languages are closed under perfect shuffle.

If you’ll be creating an FSA, provide a picture (screendump from JFLAP is OK), as well as a formal description.

Solution

note: A and B are both regular languages means that A is recognized by some finite automaton, and B is recognized by some finite automaton.

Prove: if A and B are regular languages, so is PerfectShuffle(A,B).

To prove that PerfectShuffle(A,B) is regular we demonstrate a finite automaton, M, that recognizes PerfectShuffle(A,B).
Proof by Construction.
(see page 46 of Sipser)
We will construct NFA’s:
$N_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$
$N_1$ recognizes $A$.
$N_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$
$N_2$ recognizes $B$.
$N = (Q, \Sigma, \delta, q_0, F)$
We’ll construct $N$ which is the NFA that recognizes $\text{PerfectShuffle}(A,B)$.
The idea is to construct an NFA which keeps track of the states of $A$ and $B$ and uses the $\delta$ function from either $A$ or $B$ to determine the next state for this DFA. The $\delta$ function for this DFA will use $\delta_1$ (from the DFA that recognizes $A$) when the third character of the state is a ‘1’, and will use $\delta_2$ when the character is a ‘2’.
1. $Q = Q_1 \times Q_2 \times \{1,2\}$

2. $\Sigma = \Sigma_1 \cup \Sigma_2$

3. $\delta$:
   $\delta((q_{A_i}, q_{B_j}, 1), \sigma) = (\delta_1(q_{A_i}, a), q_{B_j}, 2)$
   $\delta((q_{A_i}, q_{B_j}, 1), \sigma) = (\emptyset)$ (empty set) $\sigma \notin \Sigma_1$
   $\delta((q_{A_i}, q_{B_j}, 2), \sigma) = (q_{A_i}, \delta_2(q_{B_j}, a), 1)$
   $\delta((q_{A_i}, q_{B_j}, 2), \sigma) = (\emptyset)$ (empty set) $\sigma \notin \Sigma_2$

$q_{A_i} \in Q_1$
$q_{B_j} \in Q_2$
$\sigma \in \Sigma$

4. $q_0 = (q_{A_0}, q_{B_0}, 1)$

5. $F = F_1 \times F_2 \times \{2\}$ (states from $A$ crossed with accepting states from $B$ cross $\{2\}$)
Prove: \( L(N) \subseteq \text{PerfectShuffle}(A,B) \)

Let \( w \in L(N) \).

Then \( w \) must go through states
\( q_0, q_{i_1}, ..., q_{i_k} \) from \( N_1 \) and \( r_0, r_{i_2}, ..., r_{i_k} \) from \( N_2 \)
then the sequence of states for \( N \) to recognize \( \text{PerfectShuffle}(x,y) \) is the start state:
\( \{ q_0, r_0, 1 \} \)
and by the definition of \( \delta \) and \( \delta_1 \) the next state is:
\( \{ q_{i_2}, r_0, 2 \} \)
and by definition of \( \delta \) and \( \delta_2 \) the next state is:
\( \{ q_{i_2}, r_{i_2}, 1 \} \)
... keep alternating the use of \( \delta_1 \) and \( \delta_2 \) until we get to:
\( \{ q_{i_k}, r_{i_k, k-1}, 1 \} \) (the next to last state)
and by the definition of \( \delta \) and \( \delta_1 \) the next state is:
\( \{ q_{i_k}, r_{i_k}, 2 \} \) (accepting state)
So \( w = a_1b_1a_2b_2 ... a_kb_k \) where \( a_1...a_k \in A \) and \( b_1...b_k \in B \)
implies \( w \in \text{PerfectShuffle}(A,B) \).
So \( L(N) \subseteq \text{PerfectShuffle}(A,B) \)
Prove: \( \text{PerfectShuffle}(A,B) \subseteq L(N) \)

Assume \( A \) and \( B \) not empty (otherwise this case is trivial)

By the definition of \( \text{PerfectShuffle}(A,B) \)

\( \epsilon \notin \text{PerfectShuffle}(A,B) \)

And

Any \( w \in \text{PerfectShuffle}(A,B) \) has an even number of character from \( \Sigma \)

For any strings \( s_i \in A \) and \( r_i \in B \) used to form \( w \)

Where \( |s_i| = |r_i| = k \) and

\( w = s_{i_1} r_{i_1} s_{i_2} r_{i_2} \ldots r_{i_k} \)

All \( w \)'s will be in \( \text{PerfectShuffle}(A,B) \).

And these \( w \)'s will be recognized by \( N \).

So \( \text{PerfectShuffle}(A,B) \subseteq L(N) \)

Therefore, since

\( \text{PerfectShuffle}(A,B) \subseteq L(N) \)

And

\( \text{PerfectShuffle}(A,B) \subseteq L(N) \)

\( \text{PerfectShuffle}(A,B) \) is a regular language.

And

Any Regular Languages \( A \) and \( B \) are closed under \( \text{PerfectShuffle}(A,B) \).

Q.E.D.