The best-path algorithm for a WFSA is

- **Initialize:**
  \[ \delta_0(i) = \begin{cases} 
  1 & i = \text{initial state} \\
  0 & \text{otherwise} 
  \end{cases} \]

- **Iterate:**
  \[ \delta_k(j) = \underset{t : n[t] = j, \ell[t] = s_k}{\text{best}} \delta_{k-1}(p[t]) \otimes w[t] \]
  \[ \psi_k(j) = \underset{t : n[t] = j, \ell[t] = s_k}{\text{argbest}} \delta_{k-1}(p[t]) \otimes w[t] \]

- **Backtrace:**
  \[ t^*_k = \psi(q^*_{k+1}), \quad q^*_k = p(t^*_k) \]

where \( k \) is the number of input words that have been observed, and \( j \) is the state index. Unlike an HMM, \( \delta_k(j) = 0 \) for most states at most times. We only need to keep track of \( \delta_k(j) \) and \( \psi_k(j) \) for \((k, j)\) at which \( \delta_k(j) \neq 0 \).

Create a table:
- with columns indexed by \( k \), \( 0 \leq k \leq 5 \),
- for the utterance \([s_1, \ldots, s_5] = [\text{A, dog, is, very, hungry}]\),
- for the FSA shown above, whose transition weights are given in surprisal form.
- In each column: list the states \( j \) for which \( \delta_k(j) \neq 0 \) (\( \delta_k(j) < \infty \), since we’re using surprisals).
For each such state, list its $\delta_k(j)$ (as a surprisal), and

- list its backpointer, $\psi_k(j)$, which should be a transition, in the format $t = (p, \ell, w, n)$ showing the previous state, label, weight, and next state.

Solution:

<table>
<thead>
<tr>
<th>$k$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j$</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>$\delta_k(j)$</td>
<td>0</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
<td>3.2</td>
<td>4.1</td>
</tr>
<tr>
<td>$\psi_k(j)$</td>
<td>-</td>
<td>(0, A, 1.6, 1)</td>
<td>(1, dog, 0, 3)</td>
<td>(3, is, 0, 4)</td>
<td>(4, very, 1.6, 4)</td>
<td>(4, hungry, 0.9, 6)</td>
</tr>
<tr>
<td>$j$</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta_k(j)$</td>
<td>1.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\psi_k(j)$</td>
<td>(0, A, 1.2, 1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Problem 5.2

Show that if $u, v, x, y, z$ are surprisals, then

$$\min(u, v, x, y, z) - \ln(5) \leq u \oplus v \oplus x \oplus y \oplus z \leq \min(u, v, x, y, z)$$

Specify the values of $u, v, x, y, z$ that cause the lower bound to be met with equality. Specify the values of $u, v, x, y, z$ that cause the upper bound to be met with equality.

Solution:

Without loss of generality, assume that $u = \min(u, v, x, y, z)$. Then

$$u \oplus v \oplus x \oplus y \oplus z = -\ln \left( e^{-u} + e^{-v} + e^{-x} + e^{-y} + e^{-z} \right)$$

$$= u - \ln \left( e^0 + e^{u-v} + e^{u-x} + e^{u-y} + e^{u-z} \right)$$

Each of the terms is $0 \leq e^{u-v} \leq 1$, where the upper and lower bounds correspond to the values $v = u$ and $v = \infty$, respectively. Therefore

$$u - \ln(5) \leq u \oplus v \oplus x \oplus y \oplus z \leq u - \ln(1)$$

The lower bound is satisfied if

$$u = v = x = y = z$$

The upper bound is satisfied if one of the variables has a finite value, and all of the others are $+\infty$.

Problem 5.3

Consider the problem of translating from English into French, and then back into English again. The English-to-French WFST is called E2F. With its edge weights written as surprisals (in this case, $-\log_2 p(t)$), it is written as

![Diagram of WFST](image-url)
The French-to-English WFST is called F2E. With its edge weights written as surprisals (in this case, $-\log_2 p(t)$), it is written as

\[
\begin{array}{c}
\text{chat} : \epsilon / 1 \\
\text{cool} : \text{cool} / 0 \\
\text{cat} : \epsilon / 2 \\
\text{cool} : \epsilon : \text{cool} / 0 \\
\text{day} : \epsilon / 2 \\
\text{cool} : \text{day} / 0 \\
\text{day} : \epsilon / 0 \\
\end{array}
\]

Find the WFST $E2E = E2F \circ F2E$. You do not need to show the disconnected transitions (the transitions that can’t be reached from the start state).

**Solution:** A valid solution will only accept two input sentences: either “cool cat” or “cool day.” In response to either such sentence, it will produce the same sentence as output. If you follow the `fstcompose` algorithm given in lecture, you’ll wind up with a lot of disconnected transitions, but the important connected transitions should show as follows:

\[
\begin{array}{c}
\text{cool} : \epsilon / 0 \\
\text{cat} : \epsilon / 2 \\
\text{cool} : \epsilon : \text{cool} / 0 \\
\text{day} : \epsilon / 2 \\
\text{cool} : \text{day} / 0 \\
\end{array}
\]

**Problem 5.4**

Suppose you have two WFSTs, $A = \{\Sigma_A, \Omega_A, Q_A, E_A, i_A, F_A\}$ and $B = \{\Sigma_B, \Omega_B, Q_B, E_B, i_B, F_B\}$. Suppose we want to create $C = A \circ B = \{\Sigma_C, \Omega_C, Q_C, E_C, i_C, F_C\}$, where $Q_C = Q_A \times Q_B$ means that the states $Q_C$ are tuples of the form $q_C = (q_A, q_B)$. Let the transitions be defined in the standard way,

\[
t_A = (p[t_A], i[t_A], o[t_A], w[t_A], n[t_A]) \\
t_B = (p[t_B], i[t_B], o[t_B], w[t_B], n[t_B]) \\
t_C = (p[t_C], i[t_C], o[t_C], w[t_C], n[t_C])
\]

In each of the following cases, you’re considering a pair of transitions $t_A$ and $t_B$, and deciding how to create one or more transitions $t_C$. Specify:

- the previous state, $p[t_C]$, as a tuple: one state from $Q_A$, and one from $Q_B$ (for example, you might specify $p[t_C] = (p[t_A], p[t_B])$).
- Specify $n[t_C]$ in the same way.
- Specify also the input string $i[t_C]$, output string $o[t_C]$, and weight $w[t_C]$.

Specify $(p[t_C], i[t_C], o[t_C], w[t_C], n[t_C])$ under each of the following three cases:

(a) $t_A$ has an $\epsilon$ output string ($o[t_A] = \epsilon$).

**Solution:** $(p[t_A], p[t_B]), (n[t_A], p[t_B]), i[t_A], \epsilon, w[t_A])$
(b) \( t_B \) has an \( \epsilon \) input string (\( i[t_B] = \epsilon \)).

**Solution:** \( ((p[t_A], p[t_B]), (p[t_A], n[t_B]), \epsilon, o[t_B], w[t_B]) \)

(c) \( t_A \) and \( t_B \) have matching non-epsilon strings (\( i[t_B] = o[t_A] \neq \epsilon \)).

**Solution:** \( ((p[t_A], p[t_B]), (n[t_A], n[t_B]), i[t_A], o[t_B], w[t_A] \otimes w[t_B]) \)