Entropy Coders Based on the Splitting of Lexicographic Intervals

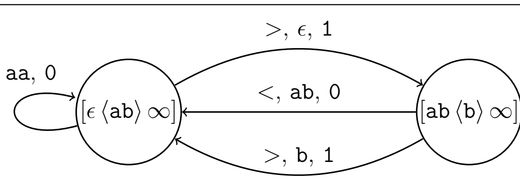
Danny Dubé (Danny.Dube@ift.ulaval.ca)

Université Laval, Quebec City, Canada

We propose a technique that performs entropy coding by splitting lexicographic intervals. We call it SLICE (for "Splitting of Lexicographic Intervals for the Coding of Entropy"). The coding proceeds as in a word-guessing game in which all the questions must be of the form "Is the secret word lexicographically smaller than w?" Syntactically, an automaton is completely defined by a set of split intervals $\widehat{\mathcal{Q}} \subset \widehat{\mathcal{I}}$. Let us first define $\mathcal{Q}$ as $\{\mathrm{U}(I) \mid I \in \widehat{\mathcal{Q}}\}$. The automaton must has the following properties. **Finiteness** $\widehat{\mathcal{Q}}$ is finite. **Existence of a start state** $[\epsilon,\, \infty] \in \mathcal{Q}$. **Closure** For any $[r\,\langle s\rangle\, t] \in \widehat{\mathcal{Q}}$, we have both $\mathrm{T}([r,\, s]) \in \mathcal{Q}$ and $\mathrm{T}([s,\, t]) \in \mathcal{Q}$. **Determinism** For any $[r,\, t] \in \mathcal{Q}$, there exists $s \in \mathcal{B}$ such that $\mathrm{S}_{\widehat{\mathcal{Q}}}([r,\, t]) = \{[r\,\langle s\rangle\, t]\}$. The coding rate $R(\widehat{\mathcal{Q}})$ is measured in source symbols per output bit.

$$
\begin{array}{rcl}
\Sigma & \triangleq & \{\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\} \qquad\qquad \mathbf{2} \triangleq \{\mathtt{0}, \mathtt{1}\} \\
\hline
\mathcal{B} & \triangleq & \{\epsilon\} \cup \Sigma^* \cdot (\Sigma - \{\mathtt{a}\}) \cup \{\infty\} \\
\hline
\mathcal{I} & \triangleq & \{[r,\, t] \mid r, t \in \mathcal{B} \text{ and } r < t\} \\
\widehat{\mathcal{I}} & \triangleq & \{[r\,\langle s\rangle\, t] \mid r, s, t \in \mathcal{B} \text{ and } r < s < t\} \\
\hline
\mathrm{X}([r,\, t]) & = & \mathrm{X}([r\,\langle s\rangle\, t]) \;\triangleq\; \{\omega \in \Sigma^\infty \mid r < \omega < t\} \\
\hline
\mathrm{U}([r\,\langle s\rangle\, t]) & \triangleq & [r,\, t] \qquad\qquad \mathrm{S}_{\widehat{\mathcal{F}}}([r,\, t]) \triangleq \left\{[r\,\langle s\rangle\, t] \in \widehat{\mathcal{F}}\right\} \\
\hline
\mathrm{T}([\epsilon, \mathtt{a} \cdot w]) & = & \mathrm{T}([\epsilon, w]) \\
\mathrm{T}([\epsilon, \mathtt{b}]) & = & [\epsilon, \infty] \\
\mathrm{T}([\epsilon, d \cdot w]) & = & [\epsilon, d \cdot w], \quad \text{if } (d = \mathtt{b} \text{ and } w \neq \epsilon) \text{ or } d > \mathtt{b} \\
\mathrm{T}([\epsilon, \infty]) & = & [\epsilon, \infty] \\
\mathrm{T}([c \cdot v, c \cdot w]) & = & \mathrm{T}([v, w]) \\
\mathrm{T}([c \cdot v, d]) & = & \mathrm{T}([v, \infty]), \quad \text{if } \mathrm{succ}(c) = d \\
\mathrm{T}([c \cdot v, d \cdot w]) & = & [c \cdot v, d \cdot w], \quad \text{if } (\mathrm{succ}(c) = d \text{ and } w \neq \epsilon) \text{ or } \mathrm{succ}(c) < d \\
\mathrm{T}([c \cdot v, \infty]) & = & [c \cdot v, \infty], \quad \text{if } c < \mathtt{z} \\
\mathrm{T}([\mathtt{z} \cdot v, \infty]) & = & \mathrm{T}([v, \infty]) \\
\hline
\mathrm{X}([r,\, t]) & = & \{w\} \cdot \mathrm{X}(\mathrm{T}([r,\, t])), \qquad \text{where } w \text{ is the LCP of } \mathrm{X}([r,\, t]) \\
\hline
R(\widehat{\mathcal{Q}}) & = & \displaystyle\sum_{[r\,\langle s\rangle\, t] \in \widehat{\mathcal{Q}}} \left(\begin{array}{l} |\mathrm{LCP}(\mathrm{X}([r,\, s]))| \cdot \Pr(r < \Omega < s \mid r < \Omega < t) \\ +|\mathrm{LCP}(\mathrm{X}([s,\, t]))| \cdot \Pr(s < \Omega < t \mid r < \Omega < t) \end{array} \right) \cdot \hat{p}([r\,\langle s\rangle\, t])
\end{array}
$$

Example of a coding automaton $\widehat{\mathcal{Q}}$:



$R(\widehat{\mathcal{Q}}) =$ 1.126 sym./bit