Chapter 3: Compression Coding

Lecture 12

Define

source S	with	symbols	$s_1, \ldots, s_q$
	with	probabilities	$p_1, \ldots, p_q$
code C	with	codewords	$\mathbf{c}_1, \ldots, \mathbf{c}_q$
		of lengths	$\ell_1, \ldots, \ell_q$
		and radix r	

A code C is

- uniquely decodeable (UD) if it can always be decoded unambiguously
- instantaneous if no codeword is the prefix of another
  Such a code is an I-code.

Decision trees can represent I-codes.

- Branches are numbered from the top down.
- Any radix r is allowed.
- Two codes are equivalent if their decision trees are isomorphic.
- By shuffling source symbols, we may assume that $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$.

Define

source S	with	symbols	$s_1, \ldots, s_q$
	with	probabilities	$p_1, \ldots, p_q$
code C	with	codewords	$\mathbf{c}_1, \ldots, \mathbf{c}_q$
		of lengths	$\ell_1, \ldots, \ell_q$
		and radix r	

By shuffling source symbols, we may assume that $p_1 \geq p_2 \geq \cdots \geq p_q$.

The (expected or) average length and variance of codewords in C are

$$L = \sum_{i=1}^{q} p_i \ell_i \qquad\qquad V = \left(\sum_{i=1}^{q} p_i \ell_i^2 \right) - L^2$$

By shuffling source symbols, we may assume that $p_1 \geq p_2 \geq \cdots \geq p_q$.

The (expected or) average length and variance of codewords in C are

$$L = \sum_{i=1}^{q} p_i \ell_i \qquad\qquad V = \left(\sum_{i=1}^{q} p_i \ell_i^2\right) - L^2$$

Example

A code C has the codewords 0, 10, 11 with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$.

Its average length and variance are

$$L = \frac{1}{2} \times 1 \; + \frac{1}{4} \times 2 \; + \frac{1}{4} \times 2 \; = \frac{3}{2}$$

$$V = \frac{1}{2} \times 1^2 + \frac{1}{4} \times 2^2 + \frac{1}{4} \times 2^2 - L^2 = \frac{5}{2} - \left(\frac{3}{2}\right)^2 = \frac{1}{4}$$

By shuffling source symbols, we may assume that $p_1 \geq p_2 \geq \cdots \geq p_q$.

The (expected or) average length and variance of codewords in C are

$$L = \sum_{i=1}^{q} p_i \ell_i \qquad\qquad V = \left(\sum_{i=1}^{q} p_i \ell_i^2 \right) - L^2$$

A UD-code is minimal with respect to $p_1, \ldots, p_q$ if it has minimal length.

Example

A code C has the codewords 0, 10, 11 with probabilities $\dfrac{1}{2}, \dfrac{1}{4}, \dfrac{1}{4}$.

Its average length and variance are

$$L = \frac{3}{2} \qquad\qquad V = \frac{1}{4}$$

It is easy to see that C is minimal with respect to $\dfrac{1}{2}, \dfrac{1}{4}, \dfrac{1}{4}$.

Example

A code C has the codewords 0, 10, 11 with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$.
Its average length and variance are

$$L = \frac{3}{2} \qquad\qquad V = \frac{1}{4}$$

It is easy to see that C is minimal with respect to $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$.

Example

A code C' has the codewords 10, 0, 11 with probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$.
Its average length is

$$L = \frac{1}{2} \times 2 + \frac{1}{4} \times 1 + \frac{1}{4} \times 2 = \frac{7}{4} > \frac{3}{2}$$

We see that C' is not minimal with respect to $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$.

By shuffling source symbols, we may assume that $p_1 \geq p_2 \geq \cdots \geq p_q$.

The (expected or) average length and variance of codewords in C are

$$L = \sum_{i=1}^{q} p_i \ell_i \qquad\qquad V = \left(\sum_{i=1}^{q} p_i \ell_i^2\right) - L^2$$

A UD-code is minimal with respect to $p_1, \ldots, p_q$ if it has minimal length.

Theorem

If a binary UD-code has minimal average length L with respect to $p_1, \ldots, p_q$, then, possibly after permuting codewords of equally likely symbols,

- $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$
- The code may be assumed to be instantaneous.
- $K = \sum_{i=1}^{q} 2^{-\ell_i} = 1$
- $\ell_{q-1} = \ell_q$
- $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place.

Theorem

If a binary UD-code has minimal average length L with respect to $p_1, \ldots, p_q$, then, possibly after permuting codewords of equally likely symbols,

- $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$
- The code may be assumed to be instantaneous.
- $K = \sum_{i=1}^{q} 2^{-\ell_i} = 1$
- $\ell_{q-1} = \ell_q$
- $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place.

Proof

Suppose that $p_m > p_n$ and $\ell_m > \ell_n$.
Swapping $\mathbf{c}_m$ and $\mathbf{c}_n$ gives a new code with smaller L, a contradiction.

Theorem

If a binary UD-code has minimal average length L with respect to $p_1, \ldots, p_q$, then, possibly after permuting codewords of equally likely symbols,

- $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$
- The code may be assumed to be instantaneous.
- $K = \sum_{i=1}^{q} 2^{-\ell_i} = 1$
- $\ell_{q-1} = \ell_q$
- $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place.

Proof

Use the Kraft-McMillan Theorem.

Theorem

If a binary UD-code has minimal average length L with respect to $p_1, \ldots, p_q$, then, possibly after permuting codewords of equally likely symbols,

- $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$
- The code may be assumed to be instantaneous.
- $K = \sum_{i=1}^{q} 2^{-\ell_i} = 1$
- $\ell_{q-1} = \ell_q$
- $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place.

Proof

If $K < 1$, then the code can be shortened, reducing L, a contradiction.

Theorem

If a binary UD-code has minimal average length L with respect to $p_1, \ldots, p_q$, then, possibly after permuting codewords of equally likely symbols,

- $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$
- The code may be assumed to be instantaneous.
- $K = \sum_{i=1}^{q} 2^{-\ell_i} = 1$
- $\ell_{q-1} = \ell_q$
- $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place.

Proof

We know that $\ell_{q-1} \leq \ell_q$. If $\ell_{q-1} < \ell_q$, then there must be nodes in the decision tree where no choice is made, implying $K < 1$, a contradiction.

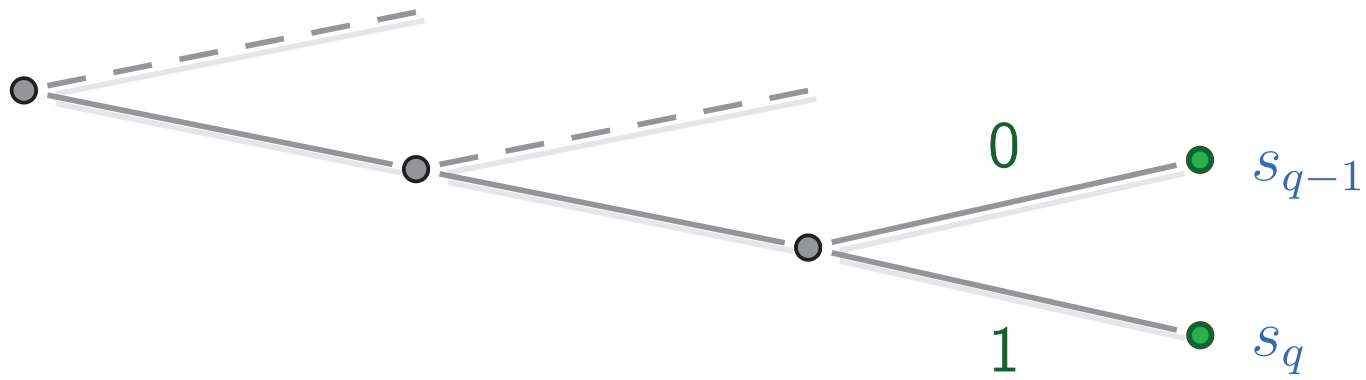
Theorem

If a binary UD-code has minimal average length L with respect to $p_1, \ldots, p_q$, then, possibly after permuting codewords of equally likely symbols,

- $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_q$
- The code may be assumed to be instantaneous.
- $K = \sum_{i=1}^{q} 2^{-\ell_i} = 1$
- $\ell_{q-1} = \ell_q$
- $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place.

Proof

The tree must end with a simple fork:



Therefore, $\mathbf{c}_{q-1}$ and $\mathbf{c}_q$ differ only in their last place. $\square$

Huffman's Algorithm (binary)

Input: a source $S = \{s_1, \ldots, s_q\}$ and probabilities $p_1, \ldots, p_q$
Output: a code C for S, given by a decision tree

Combining phase

- Replace the last 2 symbols s_{q-1} and s_q
  by a new symbol $s_{q-1,q}$ with probability $p_{q-1} + p_q$.
- Reorder the symbols $s_1, \ldots, s_{q-2}, s_{q-1,q}$ by their probabilities.
- Repeat until there is only one symbol left.

Splitting phase

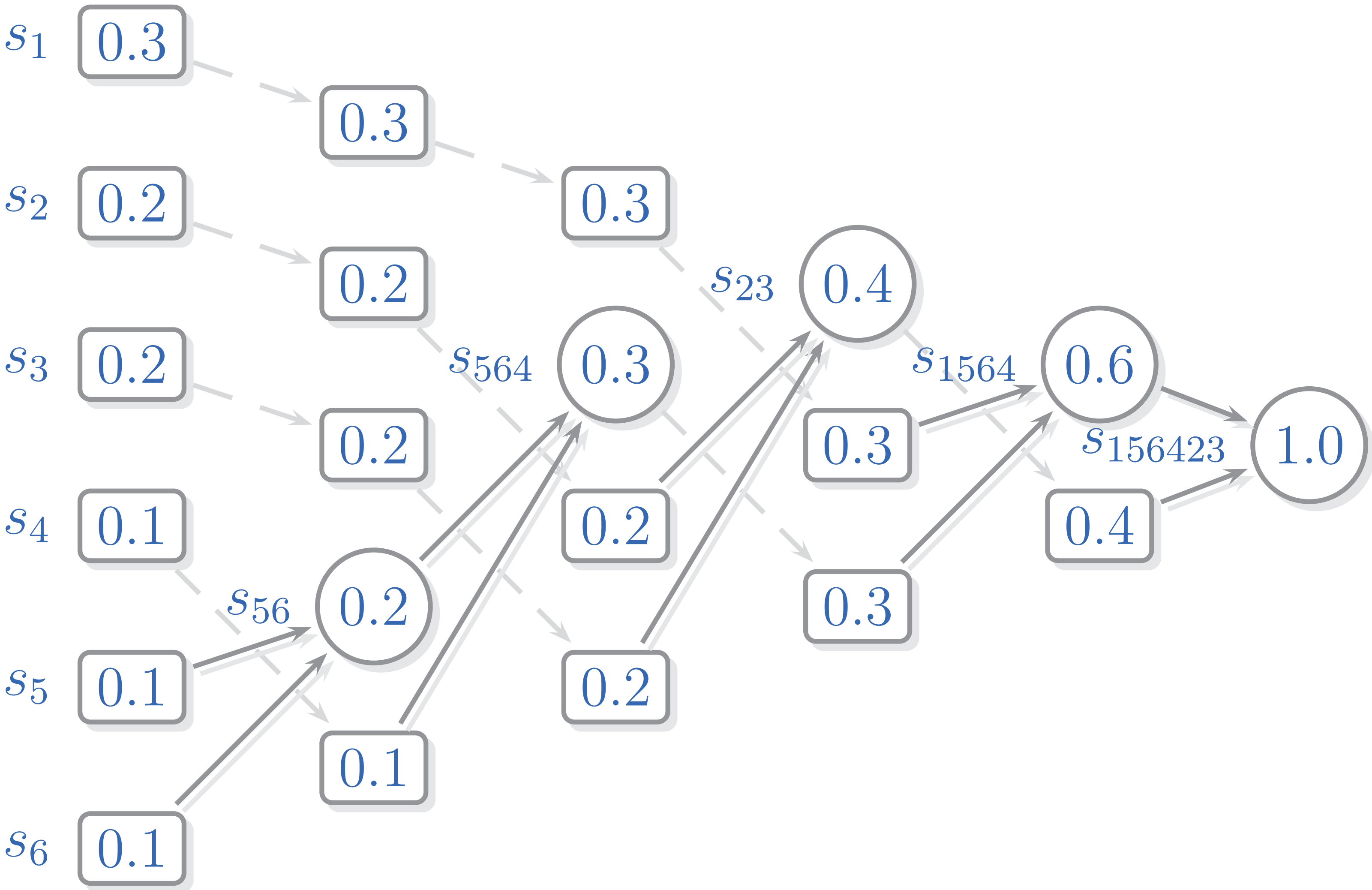
- Root-label this symbol.
- Draw edges from symbol $s_{a,b}$ to symbols s_a and s_b.
- Label edge $s_a s_{a,b}$ by 0 and label edge $s_b s_{a,b}$ by 1.

The resulting code depends on the reordering of the symbols.

Example

In the place-low strategy, we place $s_{a,b}$ as low as possible.

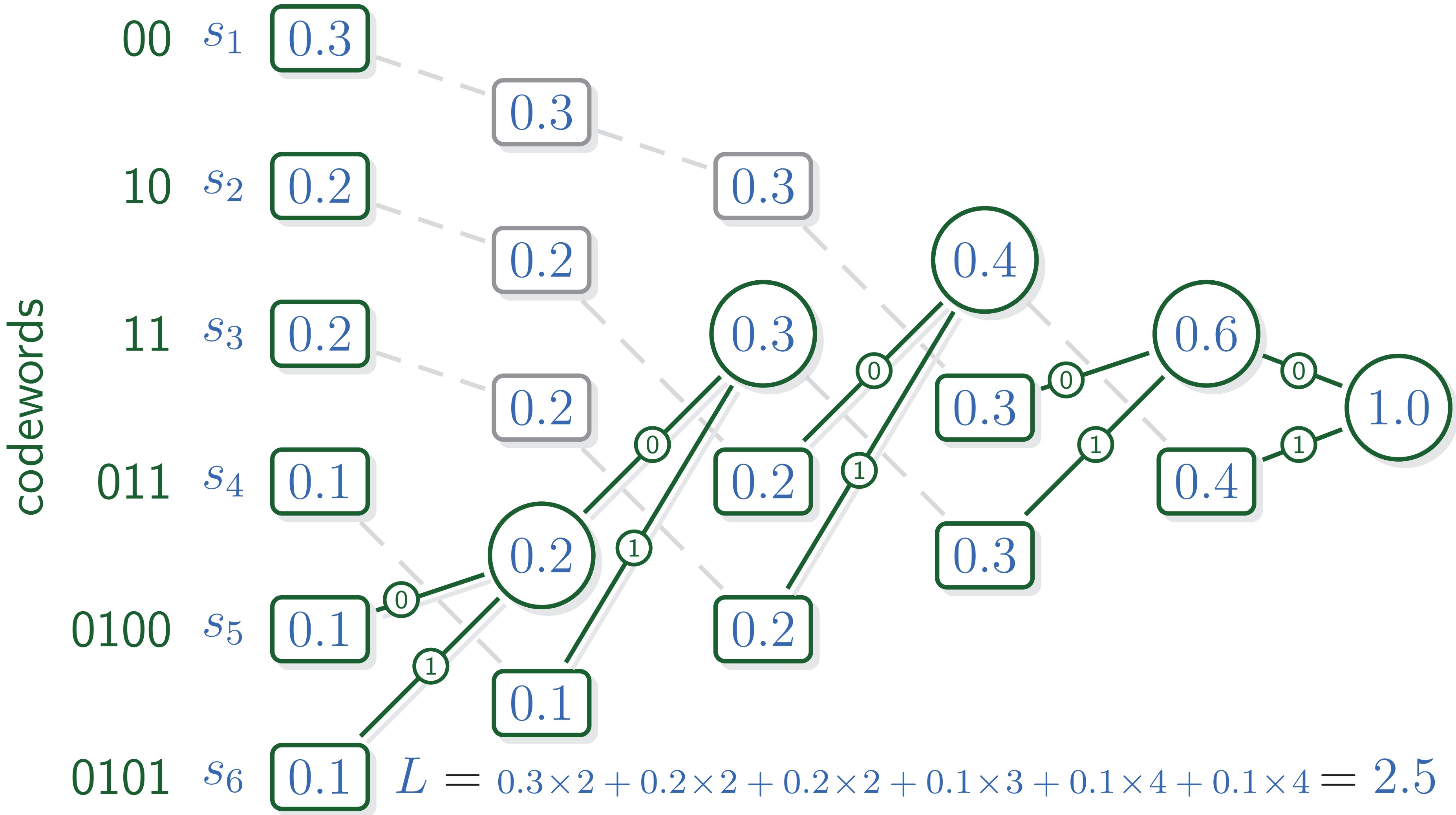
Consider a source $s_1, \ldots, s_6$ with probabilities $0.3,\ 0.2,\ 0.2,\ 0.1,\ 0.1,\ 0.1$.



Example

In the place-low strategy, we place $s_{a,b}$ as low as possible.

Consider a source $s_1, \ldots, s_6$ with probabilities $0.3,\ 0.2,\ 0.2,\ 0.1,\ 0.1,\ 0.1$.



$$L = 0.3\times 2 + 0.2\times 2 + 0.2\times 2 + 0.1\times 3 + 0.1\times 4 + 0.1\times 4 = 2.5$$

$$V = 0.3\times 2^2 + 0.2\times 2^2 + 0.2\times 2^2 + 0.1\times 3^2 + 0.1\times 4^2 + 0.1\times 4^2 - L^2 = 0.65$$

Example

In the place-low strategy, we place $s_{a,b}$ as low as possible.
Consider a source $s_1, \ldots, s_6$ with probabilities $0.3,\ 0.2,\ 0.2,\ 0.1,\ 0.1,\ 0.1$.
The generated code C has codewords

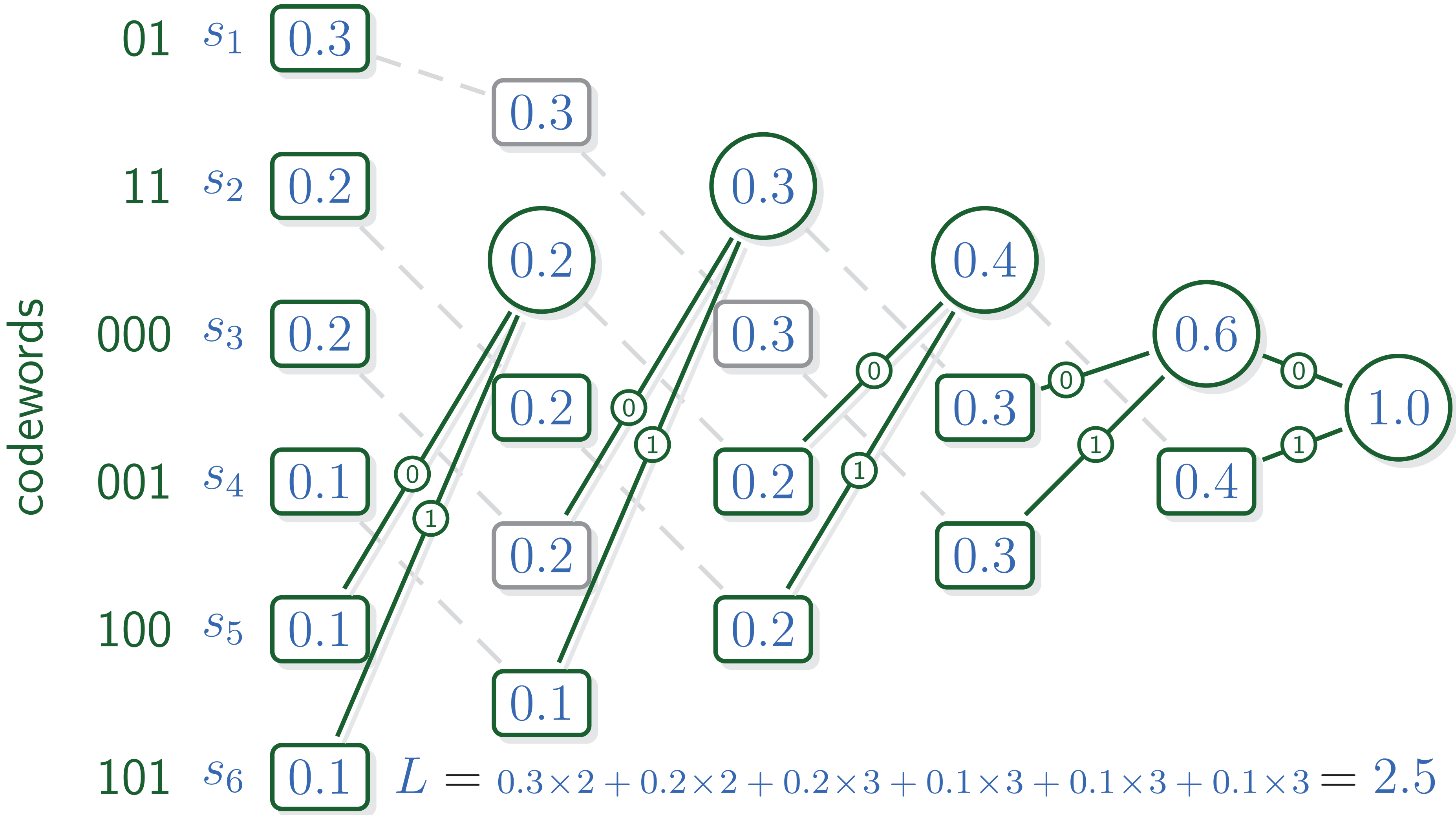
$$00 \quad 10 \quad 11 \quad 011 \quad 0100 \quad 0101$$

and average length $L = 2.5$ and variance $V = 0.65$.

Example

In the place-high strategy, we place $s_{a,b}$ as high as possible.

Consider a source $s_1, \ldots, s_6$ with probabilities $0.3,\ 0.2,\ 0.2,\ 0.1,\ 0.1,\ 0.1$.



$$L = 0.3\times 2 + 0.2\times 2 + 0.2\times 3 + 0.1\times 3 + 0.1\times 3 + 0.1\times 3 = 2.5$$

$$V = 0.3\times 2^2 + 0.2\times 2^2 + 0.2\times 3^2 + 0.1\times 3^2 + 0.1\times 3^2 + 0.1\times 3^2 - L^2 = 0.25$$

Example

In the place-high strategy, we place $s_{a,b}$ as high as possible.
Consider a source $s_1, \ldots, s_6$ with probabilities $0.3,\ 0.2,\ 0.2,\ 0.1,\ 0.1,\ 0.1$.
The generated code C has codewords

01 11 000 001 100 101

and average length $L = 2.5$ and variance $V = 0.25$.

The average length is the same as for the place-low strategy
- but the variance is smaller. It turns out that this is always the case,
so we will use only use the place-high strategy.

The Huffman Code Theorem

For any given source S and corresponding probabilities,
the Huffman algorithm yields an instantaneous minimum UD-code.