Bayesian networks

Gergely Lukács
Pázmány Péter Catholic University
Faculty of Information Technology
Budapest, Hungary
lukacs / karacs @itk.ppke.hu

Bayes theorem
(Conditional Probabilities)

- Probability of event *H* given evidence *E*:

- *A priori* probability of *H*:
  - Probability of event *before* evidence has been seen
- *A posteriori* probability of *H*:
  - Probability of event *after* evidence has been seen

Bayes theorem - example

- 1% of women have breast cancer, P(H)
  (and therefore 99% do not)
- 80% of mammograms detect breast cancer when it is there (and therefore 20% miss it).
  (P(E|H)
- 9.6% of mammograms detect breast cancer when it's **not** there, P(E|¬H)
  (and therefore 90.4% correctly return a negative result).

H

E	Cancer (1%)	No Cancer (99%)
Test Pos	80%	9.6%
Test Neg	20%	90.4%

Bayes theorem – example 2

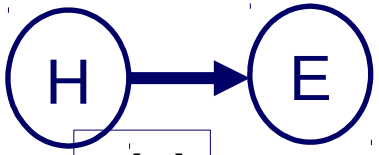
- Positive test result. What are the chances you have cancer?

	Cancer (1%)	No Cancer (99%)
Test Pos	True Pos: 1% * 80%	False Pos: 99% * 9.6%
Test Neg	False Neg: 1% * 20%	True Neg: 99% * 90.4%

- Positive result: we're in the top row of our table:
  true positive or a false positive.
- The chances of a *true positive* = chance you have cancer * chance test caught it = 1% * 80% = .008
- The chances of a *false positive* = chance you don't have cancer * chance test caught it anyway = 99% * 9.6% = 0.09504
- Normalizing (instead of : dividing by Pr[E],
  ensuring that Pr[H|E] + Pr[¬H|E] = 1)
  true positive = 0.008 / (0.008 + 0.09504) = 8 %
  false positive = 0.09504 /(0.008 + 0.09504) = 92 %

Bayesian network

- Bayesian network
  - graphical model for representing multivariate probability distributions
  - Directed acyclic network
    - Nodes: attributes
      - Probability distribution
        - » No incoming edges: Apriori (no conditions)
        - » Incoming edges: conditional
    - Edges show conditions

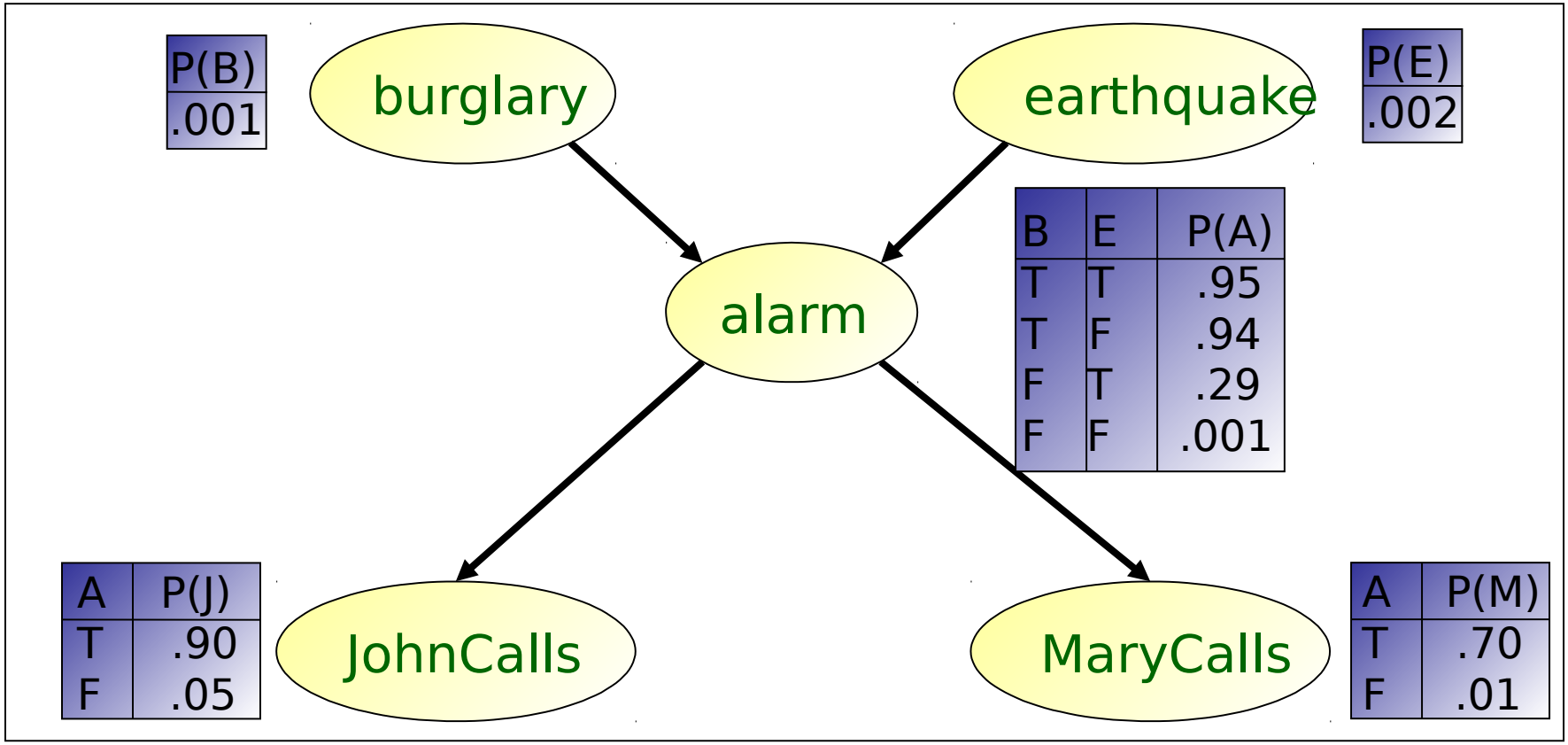


P(H)
0.01

H	P(E|H)
T	0.8
F	0.096

Bayesian Network

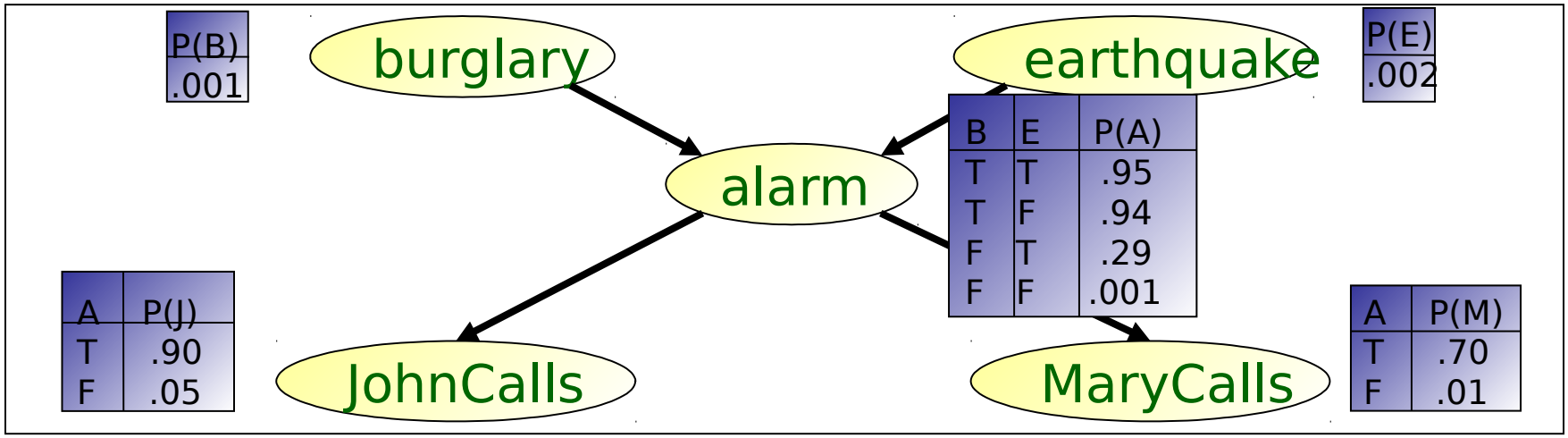
burglary-alarm example:



Acyclic directed network !

Prior probability for roots, conditional (on parents) for lower levels

Inference in Belief Networks



Many (all) types of questions can be answered, using Bayes Rule.

What is the probability that there is no burglary, nor earthquake, but that the alarm went and both John and Mary called?

$= P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) = 0.00062$

What is the probability that there is a burglary, given that John calls?

$= P(B \mid J) = ?$

0.016 (Bayes)

Entropy

Information theory

- X random variable:

$$H(X) = -\sum_{x} p(x) \log p(x).$$

- Claude E Shannon (1948) *A Mathematical Theory of Communication*, The Bell System Technical Journal, Vol. **27**, pp. 379–423, 623–656, July, October

- How to describe information sources
- How to represent information from a source
- How to transmit information over a channel

How „surprising” is an event?
(amount of information..)

- Case 1: Birmingham on 15 July at midday
  - cloudy
  - sunny

Both reasonable

- Case 2: Cairo on 15 July at midday
  - cloudy

That’s a surprise! (highly unlikely!)

  - sunny

Nothing really new…

Features of entropy

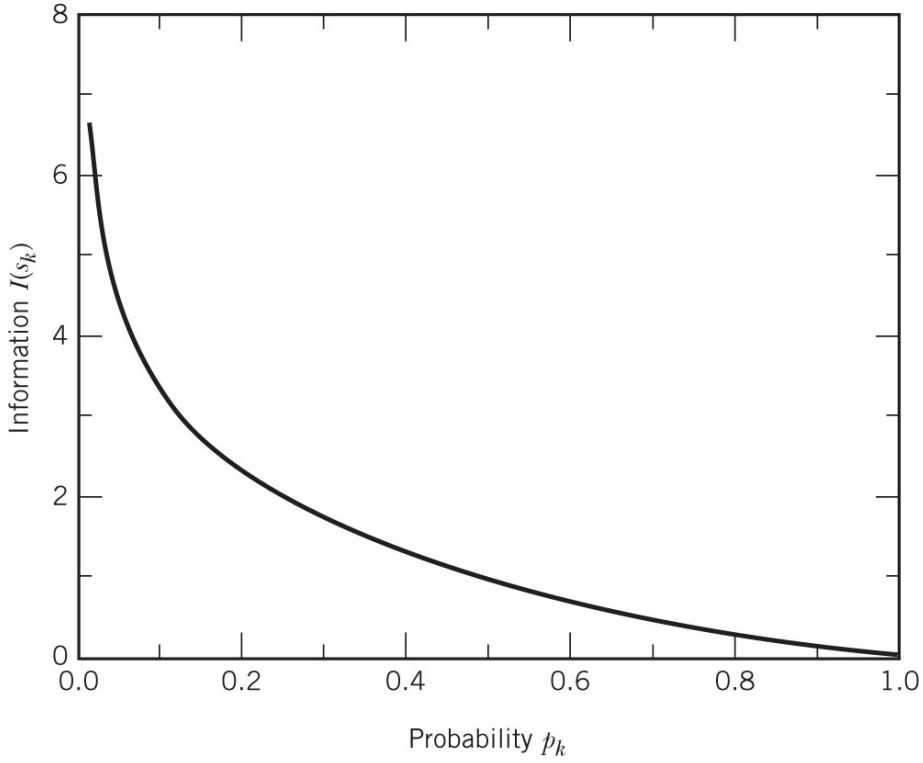
1. Lower probability of event – higher surprise (higher entropy)
2. Continuous in probability
3. If event A has a certain amount of surprise and event B has a certain amount of suprise
   - if event A and event B are independent (P(A ^B)=P(A)*P(B)) :
   the amount of surprise should sum up

$$-\log \mathbb{P}(A)$$

Base of logarithm?

- Base of logarithm?

- The choice of the logarithm base 2, and the arbitrary multiplicative constant of 1, simply give the unit of information, in this case the **bit**



Expected „surprise” of a ***source***

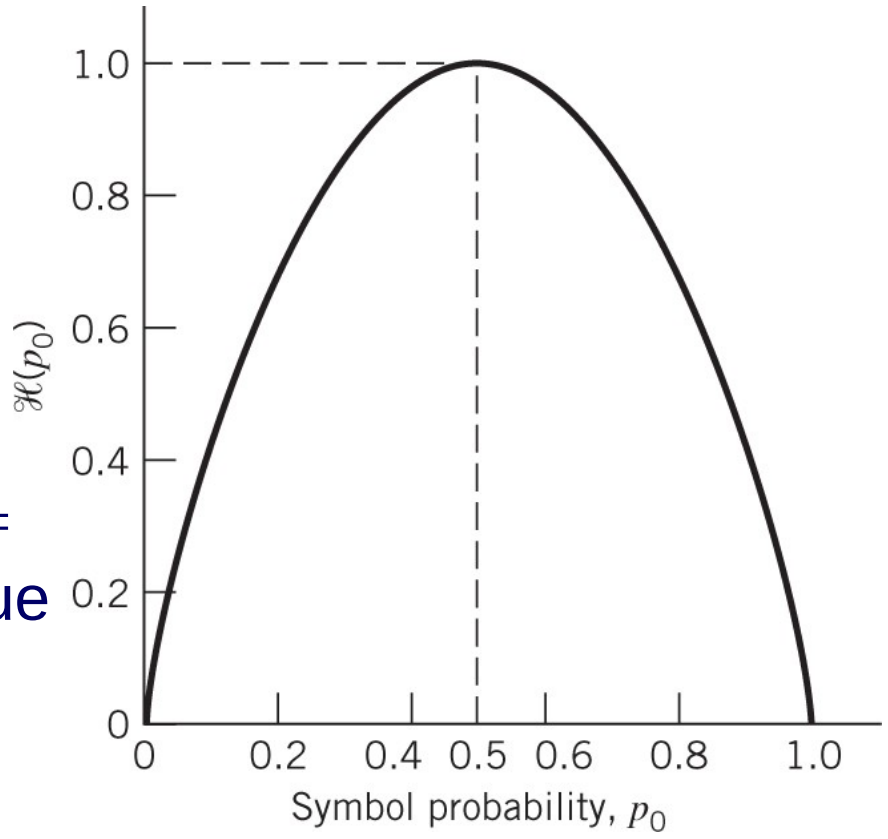
- Entropy of random variable (information source)
  - Expected surprise or
  - How surprised you expect to be on average after sampling it

$$H(X) = -\sum_{x} p(x) \log p(x).$$

  - $\log_2$ -- bits/symbol

Information sources

- Source with two outputs whose probabilities: p and $1 - p$
  - The weather forecast
- The entropy of this information source is:



- When either outcome becomes certain ($p = 0$ or $p = 1$), the entropy takes the value 0
- The entropy becomes maximum when $p = 1 - p = ½$

Information sources

- If (source with 2 outputs) p=0.5 (and 1-p=0.5), then entropy = 1, or average information content per symbol of 1 bit per symbol
- For a binary source, the entropy is maximised when both outcomes are equally likely
- This property is generally true for any number of symbols:
  - If an information source X has k symbols, its maximum entropy is $\log_2 k$ and this is obtained when all k outcomes are equally likely
- Thus, for a k symbol source:

Source coding

- It is intuitively reasonable that an information source of entropy H needs on average only H binary bits to represent each symbol
- The equiprobable binary source generates on average 1 information bit per symbol bit
- Cairo weather example:
  - Suppose the probability of cloud (C) is 0.1 and that of sun (S) 0.9
  - Entropy: 0.47 bits/symbol

Source coding/1 (naive)

- Source coding: codewords for symbols
  e.g., 0/1 for rain/no-rain
  - This representation uses 1 binary bit per symbol
  - Thus we are using more binary bits per symbol than entropy suggests is necessary
  - We want to reduce the number of bits! (== reducing redundancy)

Source coding/2 (sequences)

- Use of **sequences!**
  - codewords are not associated to a single outcome, but to a sequence of outcomes
  - Example: weather on 15, 16 and 17 July coded together

Source coding/3

- The table here shows such a *variable length code* and the probability of each codeword occurring for our weather forecasting example
- It is easy to compute that this code will on average use 1.2 bits/sequence
- Each sequence contains three symbols, so the code uses 0.4 bits/symbol

Sequence	Probability	Codeword
SSS	0.729	0
SSC	0.081	1
SCS	0.081	01
CSS	0.081	10
CCS	0.009	11
CSC	0.009	00
SCC	0.009	000
CCC	0.001	111

Source coding/4

- Using sequences decreases the average number of bits per symbol
- We have found a code that has an average bit usage less than the source entropy ???

- Parsing!
- No codeword may be a prefix of any other codeword
- sequence 011
  01-1 (SCS followed by SSC) or
  0-11 (i.e. SSS followed by CCS) ??
  Ambiguous!!

Source coding/5

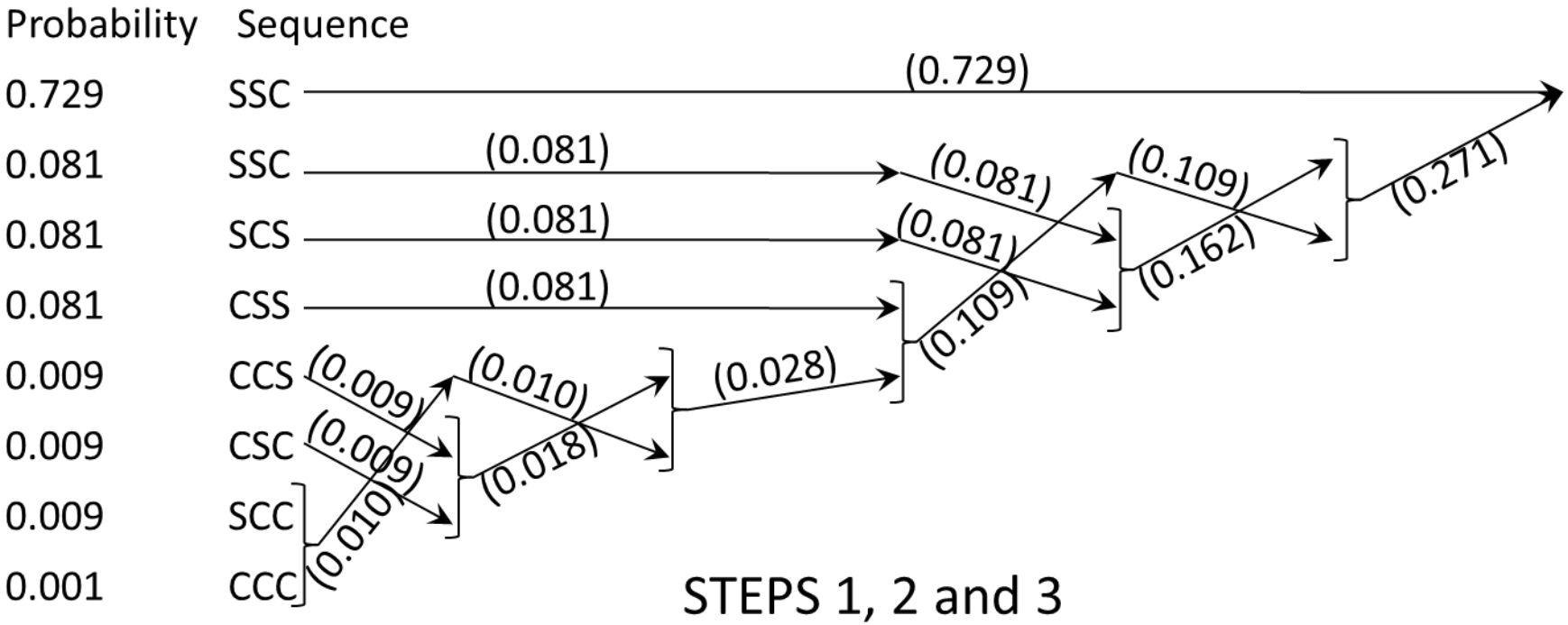
- The table here shows an ***instantaneously parseable variable length code*** and it satisfies the prefix condition
- It is easy to compute that this code uses on average 1.6 bits/sequence
- The code uses 0.53 bits/symbol
- This is a 47% improvement on identifying each symbol with a bit

Sequence	Probability	Codeword
SSS	0.729	1
SSC	0.081	011
SCS	0.081	010
CSS	0.081	001
CCS	0.009	00011
CSC	0.009	00010
SCC	0.009	00001
CCC	0.001	00000

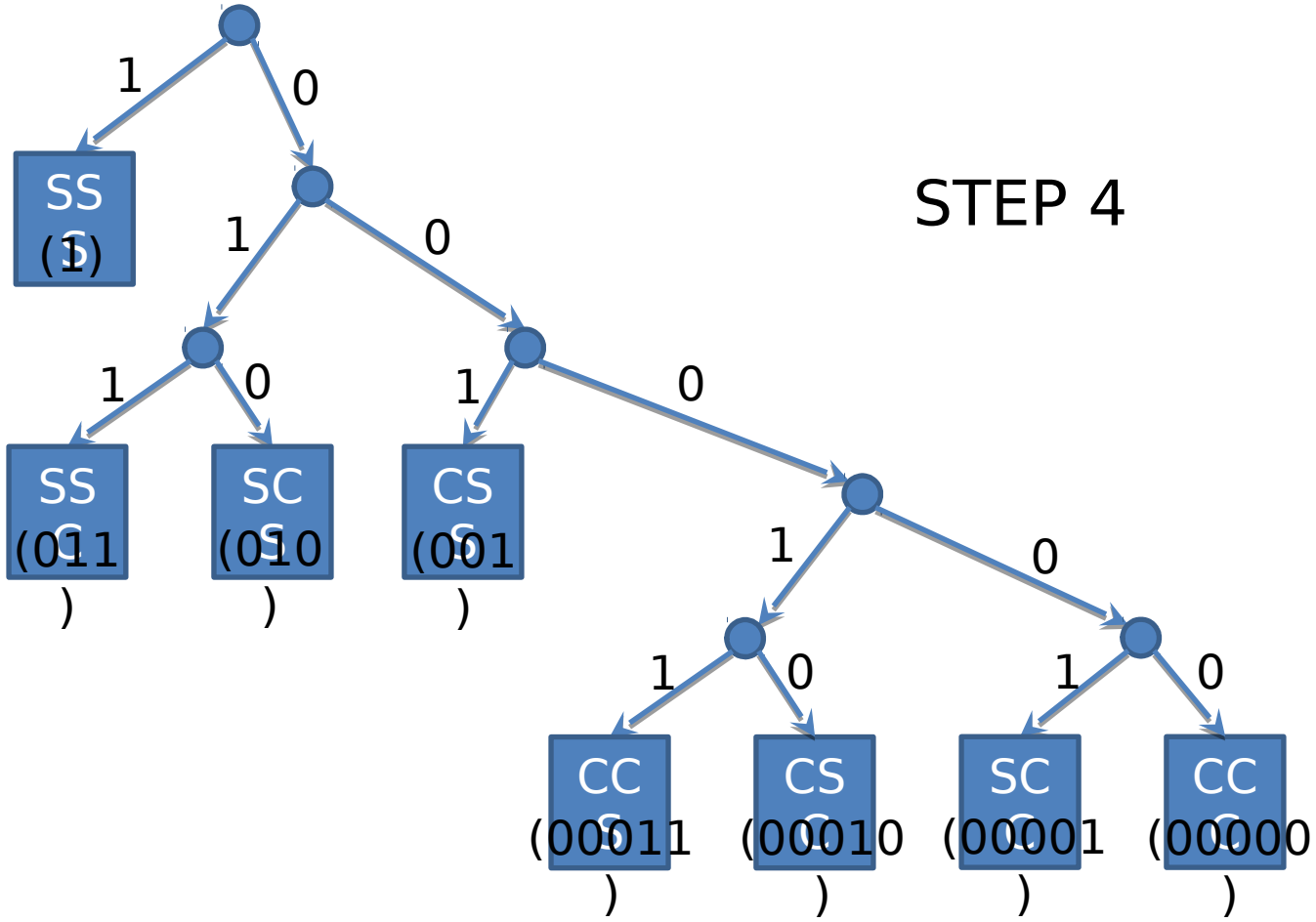
(Source coding)

- The code for each sequence is found by generating the **Huffman code tree** for the sequence
- A Huffman code tree is an unbalanced binary tree
- The derivation of the Huffman code tree is shown in the next slide
  - STEP 1: Have the n symbols ordered according to non-increasing values of their probabilities
  - STEP 2: Group the last two symbols x_{n-1} and x_n into an equivalent "symbol" with probability $p_{n-1} + p_n$
  - STEP 3: Repeat steps 1 and 2 until there is only one "symbol" left
  - STEP 4: Looking at the tree originated by the above steps (see next slide), associate the binary symbols 0 and 1 to each pair of branches at each intermediate node – the code word for each symbol can be read as the binary sequence found when starting at the root of the tree and reaching the terminal leaf node associated with that symbol

(Source coding)



(Source coding)



STEP 4

Summary

- Bayes' theorem (formula, simple example)
- Entropy
  - event
  - source;
  - formula,
  - intuitive explanation
  - upper bound (value, location)