University of Central Lancashire
UCLan

Logic expressivity: Tutorial 4.2 problem

CO2412

8th March 2022

False for *n* valuations: Complexity of the problem

Problem 4.2: Create a statement that becomes False for *n* valuations.

What is the **complexity of the problem**, *i.e.*, the best possible asymptotic efficiency of an algorithm that solves it? Let us establish some **lower bounds**:

- For n False entries in the truth table, the size of the truth table must be at least n. Therefore, $m = O(\log n)$ **atomic statements** are needed.

  Example: $n = 37$; truth table size: 64; no. of atomic statements: $m = 6$.

- Space for **encoding one atomic statement: $O(\log m)$** $= O(\log \log n)$.

  Example: $n = 2^{10{,}000}$; $m = 10{,}000$; atomic statements $p_0, \ldots, p_{9998}, p_{9999}$.

  Remark: Each atomic statement must occur (be written) at least once …

False for n valuations: Complexity of the problem

Problem 4.2: Create a statement that becomes False for n valuations.

What is the **complexity of the problem**, *i.e.*, the best possible asymptotic efficiency of an algorithm that solves it? Let us establish some **lower bounds**:

- For n False entries in the truth table, the size of the truth table must be at least n. Therefore, **$m = O(\log n)$ atomic statements** are needed.

  Example: $n = 37$; truth table size: 64; no. of atomic statements: $m = 6$.

- Space for **encoding one atomic statement: $O(\log m)$** $= O(\log \log n)$.

  Example: $n = 2^{10,000}$; $m = 10,000$; atomic statements $p_0, \ldots, p_{9998}, p_{9999}$.

- Space & time **for the whole statement: $O(m \log m)$** $= O(\log n \cdot \log \log n)$.

False for *n* valuations: Complexity of the problem

Problem 4.2: Create a statement that becomes False for *n* valuations.

What is the **complexity of the problem**, *i.e.*, the best possible asymptotic efficiency of an algorithm that solves it? $m = O(\log n)$ atomic statements needed.

Lower bound: Requires at least $O(m \log m) = O(\log n \cdot \log \log n)$ space & time.

Is there an algorithm that solves the problem in $O(\log n \cdot \log \log n)$ time? **Yes.**

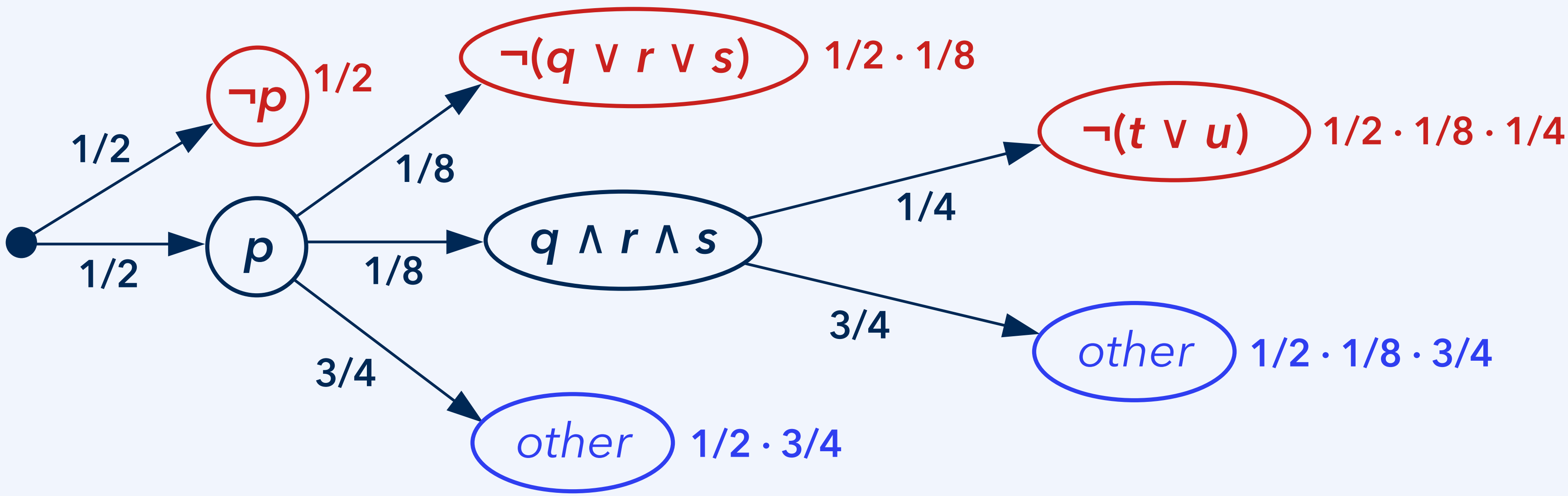
Example: $n = 37$; statement must be False for 37 out of 64 valuations:

$$\frac{37}{64} = \frac{32}{64} + \frac{4}{64} + \frac{1}{64} = \frac{1}{2} + \left(\frac{1}{2} \cdot \frac{1}{8}\right) + \left(\frac{1}{2} \cdot \frac{1}{8} \cdot \frac{1}{4}\right)$$

$$37 = (\underset{32}{\mathbf{1}}\ 0\ 0\ \underset{4}{\mathbf{1}}\ 0\ \underset{1}{\mathbf{1}})_2$$

False for *n* valuations: Example, $n = 37$

Problem 4.2: Create a statement that becomes False for *n* valuations.



Example: $n = 37$; statement must be **False for 37** out of 64 valuations:

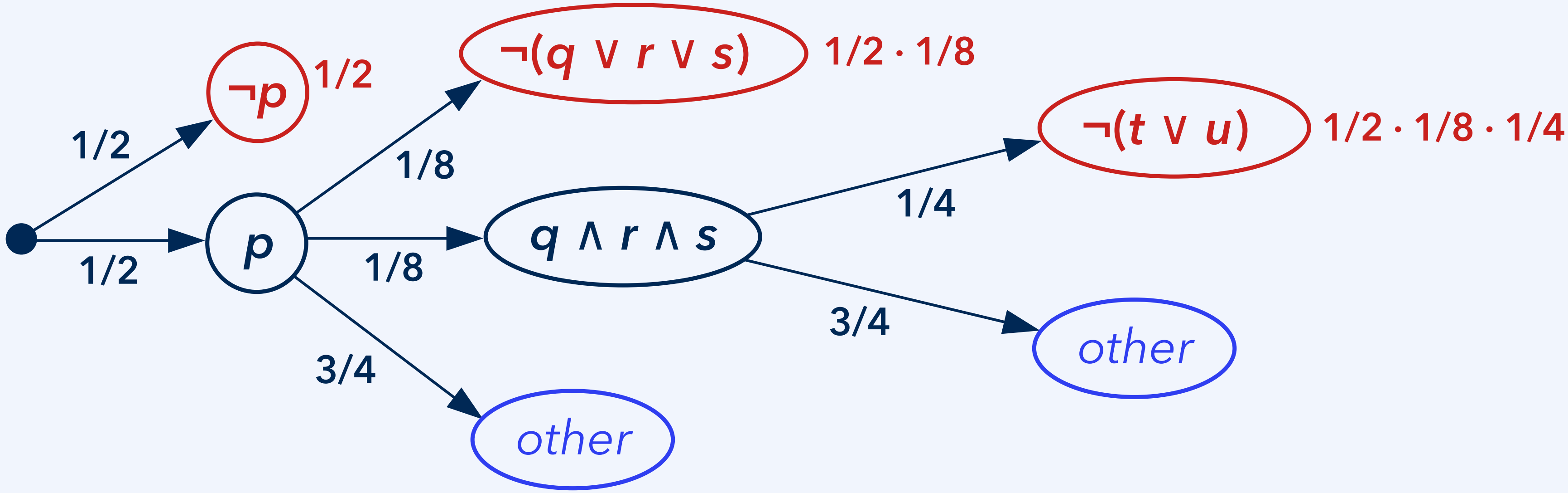
$$\frac{37}{64} = \frac{32}{64} + \frac{4}{64} + \frac{1}{64} = \frac{1}{2} + \left(\frac{1}{2}\cdot\frac{1}{8}\right) + \left(\frac{1}{2}\cdot\frac{1}{8}\cdot\frac{1}{4}\right)$$

$$37 = (1\ 0\ 0\ 1\ 0\ 1)_2$$

32 4 1

False for *n* valuations: Example, *n* = 37

Problem 4.2: Create a statement that becomes False for *n* valuations.



$$p \wedge (q \vee r \vee s) \wedge ((q \wedge r \wedge s) \rightarrow (t \vee u))$$

$$p_0 \wedge (p_1 \vee p_2 \vee p_3) \wedge ((p_1 \wedge p_2 \wedge p_3) \rightarrow (p_4 \vee p_5))$$

$$37 = (\mathbf{1}\ 0\ 0\ \mathbf{1}\ 0\ \mathbf{1})_2$$

0 1–2–**3** 4–**5**