State Final Examination (Sample Questions)

2024-09-03

1 Complement of a Regular Expression (shared topics)

Consider the following regular expression over the alphabet $\Sigma = \{a, b\}$:

$$R = ((a + b)(a + b))^* ab$$

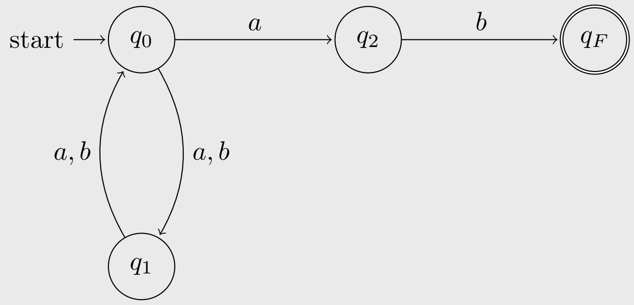
Let L be the regular language described by the expression R.

1. Construct a *nondeterministic* finite automaton A (as small as possible) recognizing the language L.
2. Use the subset construction to convert the automaton A to a *deterministic* finite automaton B.
3. From the automaton B, construct a *deterministic* finite automaton C recognizing the *complement* of the language L.

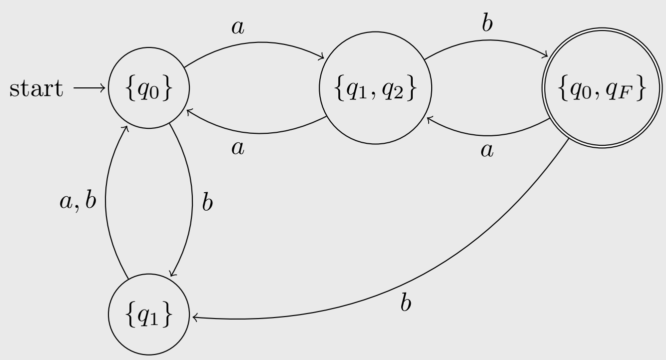
Draw state diagrams of the constructed automata A, B, and C.

Solution sketch

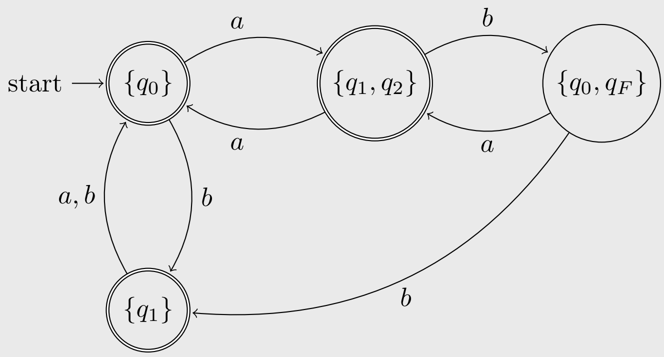
1. The smallest NFA has 4 states: $A = (\{q_0, q_1, q_2, q_F\}, \{a, b\}, \delta_A, q_0, \{q_F\})$, where δ_A is described by the state diagram:



2. The result of the subset construction is the following DFA: $B = (\{\{q_0\}, \{q_1\}, \{q_1, q_2\}, \{q_0, q_F\}\}, \{a, b\}, \delta_B, q_0, \{\{q_0, q_F\}\})$, where δ_B is described by the state diagram:



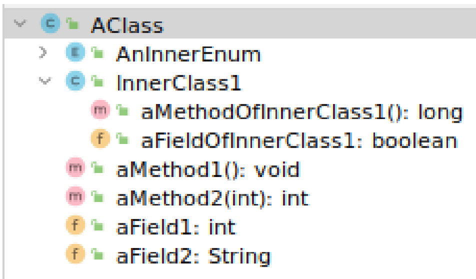
3. It suffices to swap accepting and non-accepting states:



(Note that the transition function of B is total, so there is no need to add a FAIL state.)

2 Type representation (shared topics)

1. Imagine you are developing an IDE. In the project tree window, you need to display the types and elements of the types (i.e., their methods, fields, inner/nested types – see the figure below). In a chosen language (Java, C++, C#), provide a suitable set of classes/interfaces that allow you to hold information about the types, so that they can be displayed in the window in an IDE. Information includes at least names of the types, their kind, a set of the types' methods and their signatures, the fields, and inner/nested types. Design it to be easily extensible in the future. Create only declarations of classes/interfaces, i.e., without bodies of their methods.



2. Implement a method that takes as parameters (i) a type (as you defined it for the question part above) and (ii) an "operation". The method applies the given operation on the given type and all elements of the type. If you base the traversal of the type on a well-known solution, explicitly name it. For the operation, use a suitable concept offered by your chosen language.

3. Declare a call of your method defined above. The operation passed to the method is "printing out the name of an element but only if the element represents a field or type (including inner/nested types)".

Solution sketch

1. Multiple correct solutions are possible. The typical one is an obvious class hierarchy, e.g., abstract class Element, and then the classes that extend the Element, i.e., class Type, class Method, class Field, etc.

2. Multiple correct solutions are possible. Traversal might be based, e.g., on the visitor pattern, or the types might be declared as sealed (if the language allows it) and traversal uses pattern matching, etc.

   To represent the operation, an interface can be used or a functional type, etc.

3. Based on the operation representation, it will be defined as an anonymous class, lambda expression, etc.

3 Integrals and primitive functions (shared topics)

1. Define the notion of *primitive function* (also known as an *antiderivative*) of a given function f on an open interval I.

2. State the 'per partes' integration rule (also known as integration by parts). For simplicity, you may restrict yourself to the situation when all the relevant functions are continuous.

3. Let $P \subseteq \mathbb{R}^2$ be the region of the plane bounded from below by the x-axis and bounded from above by the graph of the function $f(x) = x\cos(x)$ with $x \in [0, \pi/2]$. More formally,

$$P = \left\{(x, y) \in \mathbb{R}^2;\ 0 \leq x \leq \frac{\pi}{2} \ \wedge\ 0 \leq y \leq x\cos(x)\right\}.$$

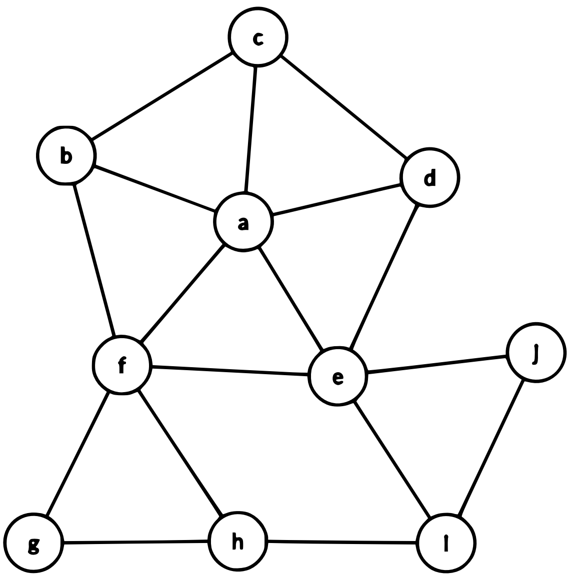
Determine the area of P.

Solution sketch

1. A function F is a primitive function of f on an interval I, if in every point $x \in I$ the function F has a derivative equal to $f(x)$.
2. The 'per partes' integration rule states that if f and g are continuous functions on an interval I, with respective antiderivatives F and G, then
$$\int f(x)G(x)\mathrm{d}x = F(x)G(x) - \int F(x)g(x)\mathrm{d}x.$$
3. The area of P can be expressed as an integral $\int_0^{\pi/2} x\cos(x)\mathrm{d}x$. The value of the integral can then be calculated with the help of the per partes rule:
$$\begin{aligned}\int_0^{\pi/2} x\cos(x)\mathrm{d}x &= [x\sin(x)]_0^{\pi/2} - \int_0^{\pi/2} \sin(x)\mathrm{d}x \\ &= \left(\frac{\pi}{2} - 0\right) - [-\cos(x)]_0^{\pi/2} \\ &= \frac{\pi}{2} - 1.\end{aligned}$$

4 Graph chromatic number (shared topics)

1. Define what is a graph's chromatic number.
2. What are the constraints on the chromatic number of planar graphs?
3. Determine the chromatic number of the graph below.



5 Homogeneous subgraphs (specialization OI-G-PADS, OI-G-PDM, OI-O-PADS, OI-PADS-PDM)

1. State the Ramsey theorem for pairs, with any finite number of colors, in the version for infinite graphs.
2. Does the following statement hold? Let $G = (A, B, E)$ be an infinite bipartite graph, i.e., let A and B be infinite sets and let E be a subset of $A \times B$. Then there exist infinite subsets $A_1 \subseteq A$ and $B_1 \subseteq B$ such that G either contains all edges between A_1 and B_1, or no such edge.

Solution sketch

1. Let p be a positive integer, and let G be an infinite clique G with edges colored by colors $1, \ldots, p$. Then there exists an infinite subset $K \subseteq V(G)$ such that all edges of G between the vertices of K have the same color.
2. This statement is false. Suppose for example that $A = \{a_1, a_2, \ldots\}$, $B = \{b_1, b_2, \ldots\}$, and $E = \{(a_i \in A, b_j \in B) : i < j\}$. For any infinite subsets $A_1 \subseteq A$ and $B_1 \subseteq B$, there exist indices $i_1 < i_2 < i_3$ such that $a_{i_1}, a_{i_3} \in A_1$ and $b_{i_2} \in B_1$, and thus $a_{i_1} b_{i_2}$ is an edge and $a_{i_3} b_{i_2}$ is a non-edge.

6 Set cardinality (specialization OI-G-PDM, OI-PADS-PDM)

1. Write the definitions of the following notions: "the set A has strictly smaller cardinality than the set B" and "the set A is countable". For each of the notions, provide an example of a set or sets satisfying the relation and an example of a set or sets which do not satisfy it.
2. Let us consider the claim "there does not exist an uncountable set of strictly smaller cardinality than real numbers", i.e., the continuum hypothesis. What can you say about the validity of this statement? Explain your answer in terms of the existence of consistent extensions and models of Zermelo–Fraenkel set theory.

Solution sketch

1. The set A has strictly smaller cardinality than the set B if there exists an injection from A to B, but not from B to A. As an example, the set of natural numbers has strictly smaller cardinality than the set of real numbers, but not strictly smaller cardinality than the set of integers.

   The set A is countable if A is finite or has the same cardinality as the set of natural numbers. For example, the set of integers is countable but the set of all real numbers is not.
2. The continuum hypothesis can be neither proven or disproven in the Zermelo–Fraenkel set theory, i.e., it is independent of it. This means that we can extend the Zermelo–Fraenkel set theory by adding the exiom stating either that the continuum hypothesis is valid or that it is false and both extensions are consistent (assuming of course that the Zermelo–Fraenkel set theory by itself is consistent). Thus, there exist models of the Zermelo–Fraenkel set theory in which the continuum hypothesis holds, as well as models in which it is false.

7 Dynamic programming (specialization OI-G-PADS, OI-O-PADS, OI-PADS-PDM)

Consider the following problem: We are given a sequence of integers $a_1, \ldots, a_n \in \{1, \ldots, m\}$ and an integer $s \geq 0$. We want to find a subsequence $a_{i_1}, \ldots, a_{i_k}$ (where $i_1 < \ldots < i_k$) whose sum is equal to s. We will call it the *treasure*.

1. Design an algorithm that decides if a treasure exists.
2. Modify this algorithm to output one of the treasures.
3. Let us accompany each a_i by its cost $c_i \in \mathbb{N}$. Modify the algorithm, so that it finds the cheapest treasure, i.e., the one with the minimum possible sum of costs of its elements. Output this minimum cost.

Analyze time and space complexity of all three algorithms. The complexities should be polynomial with respect to n and m.

Solution sketch

1. We compute a matrix $X_{ij} \in \{0,1\}$ for $0 \le i \le n$, $0 \le j \le s$. The value $X_{ij} = 1$ means that there is a subsequence of $a_1, \ldots, a_i$ which sums to j. We will fill in the matrix in order of increasing i.

    1. Initialize $X_{00} \leftarrow 1$ a $X_{0j} \leftarrow 0$ for all $j > 0$.
    2. For $i = 1, \ldots, n$:
    3.   For $j = 0, \ldots, s$:
    4.   If $j \ge a_i$ and $X_{i-1,j-a_i} = 1$, we set $X_{ij} \leftarrow 1$.
    5.   Otherwise we set $X_{ij} \leftarrow X_{i-1,j}$.
    6. We check if $X_{ns} = 1$.

    The cases in steps 4 and 5 correspond to the new item a_i being used in the subsequence, and being unused, respectively.

2. Whenever $X_{ij} = 1$, we record P_{ij}: the index of the item most recently added to the subsequence. It suffices to augment step 4 by $P_{ij} \leftarrow i$ and step 5 by $P_{ij} \leftarrow P_{i-1,j}$. In the last step, we reconstruct the treasure we found from the most recently added item to the earliest one.

3. We re-define X_{ij} to the minimum cost of a subsequence of $a_1, \ldots, a_i$ which sums to j; or ∞ if there is no such subsequence. We initialize $X_{00} \leftarrow 0$ a $X_{0j} \leftarrow \infty$ for all $j > 0$. We replace steps 4 and 5 by $X_{ij} \leftarrow \min(X_{i-1,j-a_i} + c_i, X_{i-1,j})$. At the end, we output X_{ns}.

All three algorithms run in time and space $\Theta(ns)$. If we need not print out the whole treasure, it suffices to keep only the current and preceding row of the matrix, which reduces space to $\Theta(s)$ while keeping time.

8 Closed sets in metric spaces (specialization OI-G-PADS, OI-G-PDM, OI-O-PADS, OI-PADS-PDM)

1. Let (M, d) be a metric space, that is, M is a set and d is a metric on M. State the definition of an open set and the definition of a closed set in the space (M, d).
2. Let us consider the set $\mathbb{R}$ of real numbers with its usual metric d defined by $d(x, y) = |x - y|$. For each of the following two sets, determine and briefly justify whether it is closed in $(\mathbb{R}, d)$:
    (a) the set $X_a = [0, +\infty)$
    (b) the set $X_b = \{\frac{1}{n}; n \in \mathbb{N}\} = \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\}$
3. Let us again consider the set of real numbers $\mathbb{R}$ with its usual metric. Let f be a function from $\mathbb{R}$ to $\mathbb{R}$. We say that a number $x \in \mathbb{R}$ is a *root of f* if $f(x) = 0$. Show that if f is continuous on $\mathbb{R}$, then the set of its roots is a closed set.

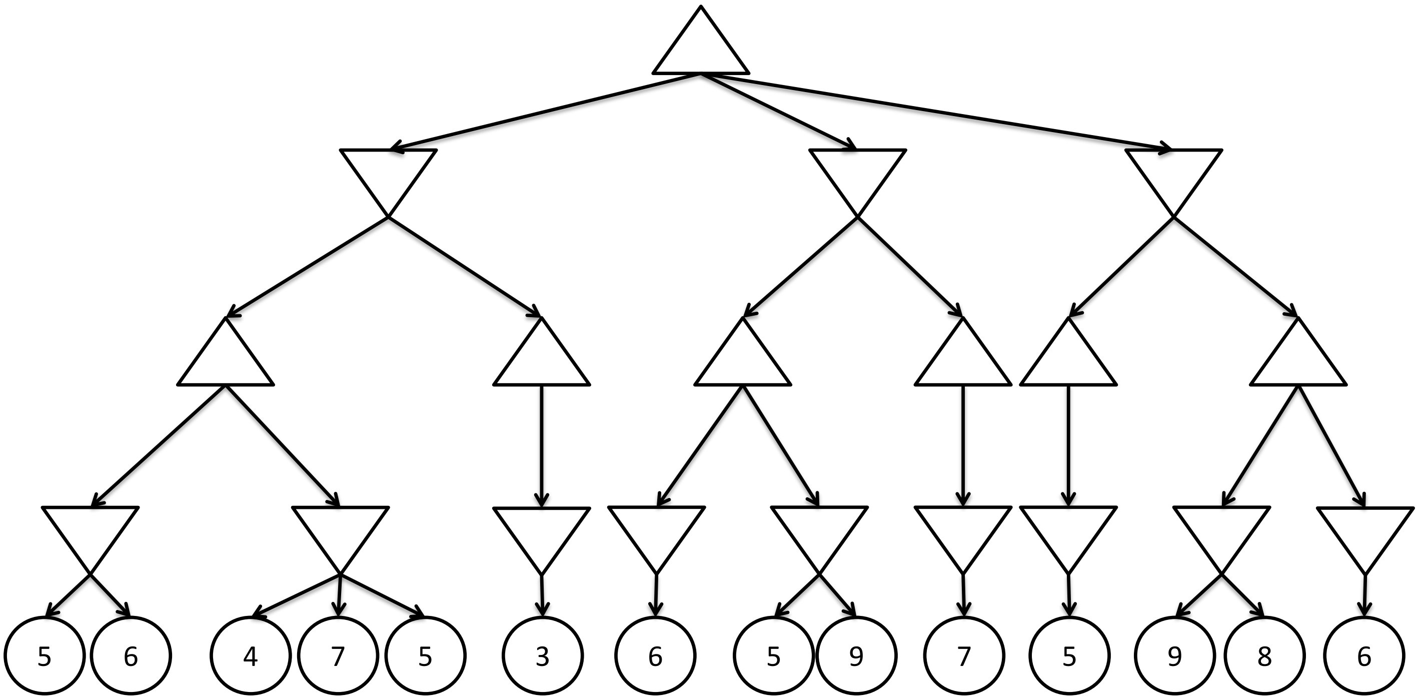
Solution sketch

1. A set X is open in (M, d), if for every $x \in X$ there is a neighborhood of x of positive radius which is a subset in X; more formally, for every $x \in X$ there is an $\varepsilon > 0$ such that the set $\{y \in M;\ d(x, y) < \varepsilon\}$ is a subset of X. A set X is closed if its complement $M \setminus X$ is open. Alternatively, it is also possible to define that a set X is closed if for every convergent sequence in M, if every member of the sequence is in X, then also the limit of the sequence is in X.
2. (a) The set X_a is closed: its complement $(-\infty, 0)$ is clearly an open set.
    (b) The set X_b is not closed: its complement contains zero, but every neighborhood of zero of positive radius intersects X_b. Hence, the complement of X_b is not open and X_b is not closed.
3. Suppose that $f\colon \mathbb{R} \to \mathbb{R}$ is continuous, let R_f be the set of its roots, and let $\overline{R}_f = \mathbb{R} \setminus R_f$ be the complement of R_f. We wish to show that $\overline{R}_f$ is open. Let us fix $x \in \overline{R}_f$ arbitrarily and show that a neighborhood of x of positive radius is in $\overline{R}_f$. From the definition of $\overline{R}_f$, we know that $f(x) \neq 0$, so let us define $\varepsilon = |f(x)|/2 > 0$. By the definition of continuity, there is a $\delta > 0$ such that for every x' with $|x - x'| < \delta$ we have $|f(x) - f(x')| < \varepsilon$. It follows that every x' satisfying $|x - x'| < \delta$ is in $\overline{R}_f$, since $|f(x')| > |f(x)| - |f(x) - f(x')| > 2\varepsilon - \varepsilon > 0$. In particular, the neighborhood of x of radius δ is contained in $\overline{R}_f$, showing that $\overline{R}_f$ is open and R_f is closed.

Another solution might use the theorem stating that the continuous preimage of a closed set is again a closed set. Since the set $\{0\}$ is closed, its preimage R_f is closed as well.

9 Minimax (specialization UI-SU, UI-ZPJ)

1. Describe the Minimax algorithm for a two-player game. Explain what practical shortcomings this algorithm has and how they can be addressed.
2. Introduce the Alpha-Beta pruning technique.
3. In the following example of a zero-sum game tree, the symbol $\triangle$ indicates a move by the maximizing player (the edges leading down from this symbol correspond to possible moves by the maximizing player). Similarly, the symbol $\triangledown$ indicates a move by the minimizing player. The symbol $\circ$ represents a terminal state with the given evaluation of that state. Fill in the non-terminal states with the evaluation that the Minimax algorithm would assign using Alpha-Beta pruning. Highlight which parts of the tree will be pruned. Assume that the possible moves are evaluated from left to right.



Solution sketch

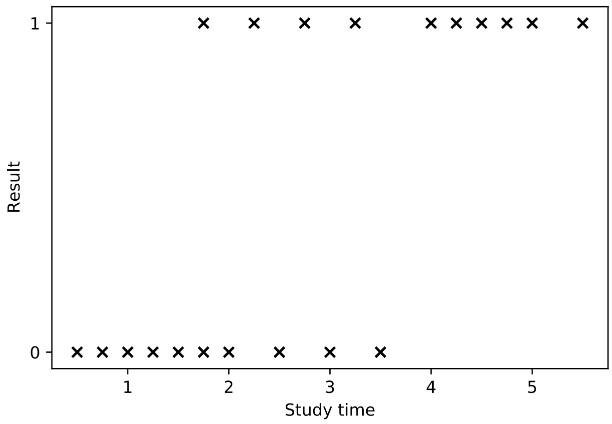
- Minimax is a recursive or backtracking algorithm that traverses a game tree of an n-player game in a DFS manner assigning minimax values to each node (i.e. the best possible outcome from this state). It provides an optimal move for the player assuming the opponent is also playing optimally.
- The main shortcoming is the size of the tree, as the bottom of the tree needs to be reached to get the outcome value propagated towards the root. This may be reduced by setting a cutoff depth where the value is assigned by an evaluation function instead of by the outcome of the game. This, however, introduces the problem of imperfect evaluation and the problem of horizon.

– Alpha-Beta pruning skips some subtrees based on the knowledge that this subtree will not be reached because a more preferred (by one of the players) subtree has been already identified. Using Alpha-Beta does not change the outcome of the algorithm, but can provide a speedup.

– In the example, the value in the root will be 6. The following nodes will not be visited: the 6th node from the left in the third layer (and thus all of the children), the 4th, 5th, and 9th node from the left in the last layer.

10 Logistic Regression (specialization UI-SU, UI-ZPJ)

We want to use logistic regression to create a model that would predict the success of students in an exam based on the time they spent studying. In order to obtain the data, we asked the students how long they studied. Their answers, together with the results of the exam – pass (1) or fail (0), are in the plot below.



1. Describe the logistic regression model. How does it calculate the output? How are its parameters trained? What loss function is used?
2. Draw the approximate output of the model trained on the given data to the plot above. You do not have to calculate anything.
3. How would the model differ if we wanted to classify the data into more than two classes (e.g. if we also wanted to predict the grade)?

Solution sketch

1. The model expresses the probability that a point x belongs to a class C_x as $P(C_x|x) = \sigma(x^T w + b)$, where w and b are trainable parameters and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. The loss function is $E(w) = \frac{1}{N}\sum_i -\log(P(C_{x_i}|x_i; w))$, where N is the size of the training set, and C_{x_i} is the target class for input x_i. The training algorithm uses the gradient of this loss to minimize the loss function. The exact derivation of the formula can be found in the slides for the NPFL129, Lecture 3.
2. The prediction has the shape of the sigmoid function that is close to 0 for the leftmost observation, and close to 1 for the rightmost one.
3. In multiclass classification, the model has mutliple outputs (one for each of the possible classes). The loss function is generalized to the softmax function $\text{softmax}(z)_j = \frac{e^{z_j}}{\sum_i e^{z_i}}$.

11 Binary classifier evaluation (specialization UI-SU, UI-ZPJ)

A test set contains 1,000 examples. A binary classifier classified 400 test examples as positive and 600 test examples as negative with 70 % accuracy and 80 % specificity.

1. Write the whole confusion matrix.

2. Compute the classifier's precision and recall.
3. Given that data, is it possible to exactly calculate F-score? If yes, how?

12 Decision tree (specialization UI-SU)

We have data with three attibutes A, B, and C and a target class Y given in the tables below.

1. Describe the decision tree learning algorithm and apply it to the training data in the table. Continue even with small amounts of data in split nodes and/or leaves.
2. For the attribute selected to the root, write down the formula for selection criterion and evaluate it. In case your selection criterion uses logarithms, you may use approximations provided here.

x	1/5	1/3	2/5	3/5	2/3	4/5
$\approx log_2(x)$	-2.32	-1.58	-1.32	-0.74	-0.58	-0.32

3. Describe a standard pruning method. If the method is based on validation data, use the data provided in the table below.

Training data:

	A	B	C	Y
0	1	2	6	n
1	1	2	6	n
2	3	2	8	y
3	3	4	6	y
4	3	4	8	y
5	3	4	8	n
6	3	4	8	n
7	3	4	6	y

Validation data:

	A	B	C	Y
8	1	2	8	n
9	3	4	6	y
10	3	4	6	y
11	3	4	8	y

Solution sketch The basic selection criteria are entropy gain or gini gain. The (binary) entropy is defined as

$$H(a,b) = -\frac{a}{a+b}\log_2(\frac{a}{a+b}) - \frac{b}{a+b}\log_2(\frac{b}{a+b})$$

for example

$$H(2/6,4/6) = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) \approx -\frac{-1.58}{3} - \frac{2\cdot(-0.58)}{3} = \frac{2.74}{3} \approx 0.91$$

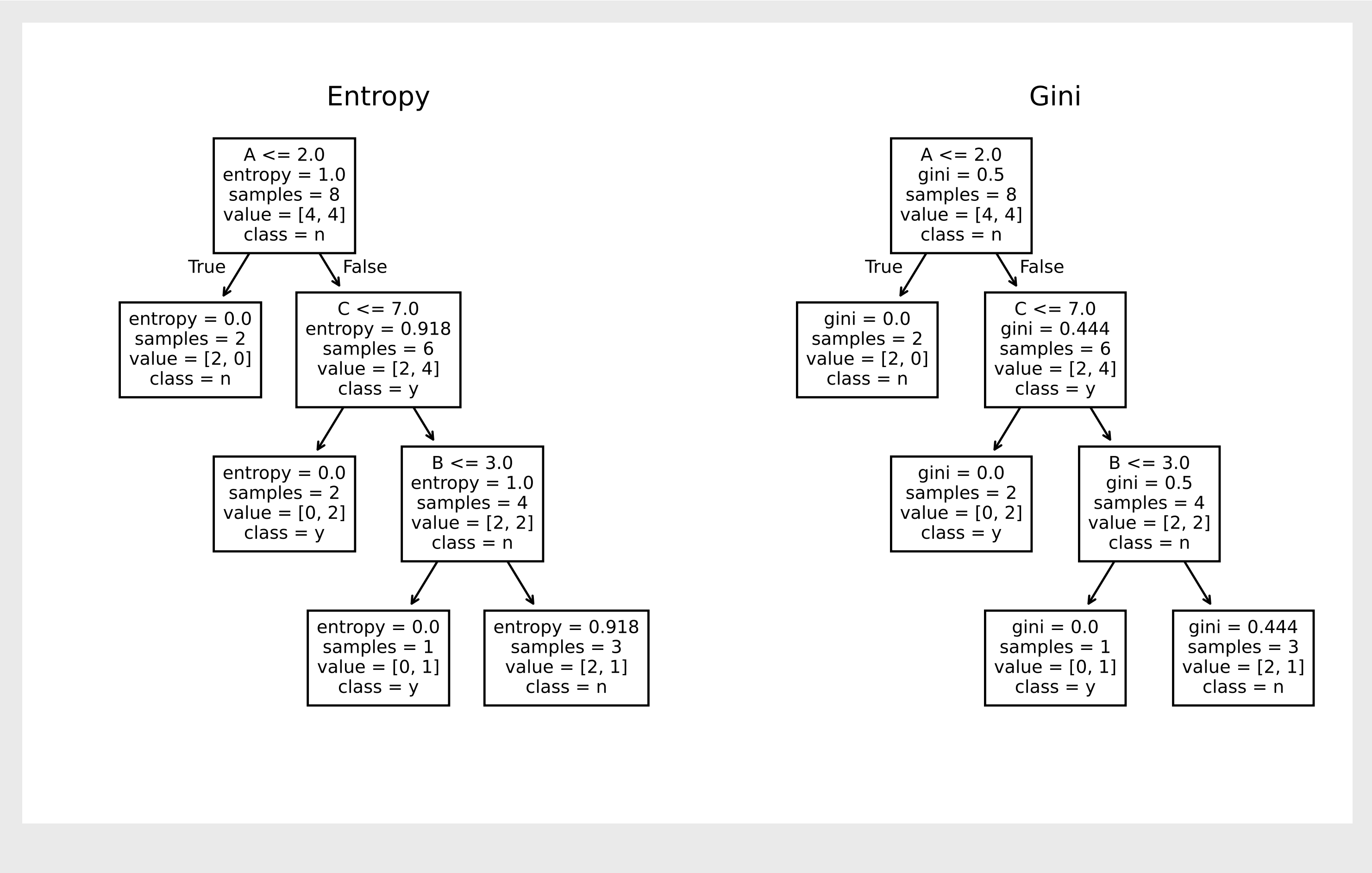
The entropy gain is the difference between the entropy before the split and the weighted average entropy after the split. For the root attribute A the entropy gain is:

$$H(4,4) - \frac{2}{8}\cdot 0 - \frac{6}{8}H(2,4) \approx 1 - \frac{6}{8}\cdot 0.91 \approx 0.32.$$

The resulting tree is the same based on entropy or gini, see the figure.

Based on validation data the node C should be replaced by a leaf since this decreases the validation error.

Alternatively, the cost complexity pruning may be used.


Entropy
A <= 2.0
entropy = 1.0
samples = 8
value = [4, 4]
class = n
True
False
entropy = 0.0
samples = 2
value = [2, 0]
class = n
C <= 7.0
entropy = 0.918
samples = 6
value = [2, 4]
class = y
entropy = 0.0
samples = 2
value = [0, 2]
class = y
B <= 3.0
entropy = 1.0
samples = 4
value = [2, 2]
class = n
entropy = 0.0
samples = 1
value = [0, 1]
class = y
entropy = 0.918
samples = 3
value = [2, 1]
class = n
Gini
A <= 2.0
gini = 0.5
samples = 8
value = [4, 4]
class = n
True
False
gini = 0.0
samples = 2
value = [2, 0]
class = n
C <= 7.0
gini = 0.444
samples = 6
value = [2, 4]
class = y
gini = 0.0
samples = 2
value = [0, 2]
class = y
B <= 3.0
gini = 0.5
samples = 4
value = [2, 2]
class = n
gini = 0.0
samples = 1
value = [0, 1]
class = y
gini = 0.444
samples = 3
value = [2, 1]
class = n