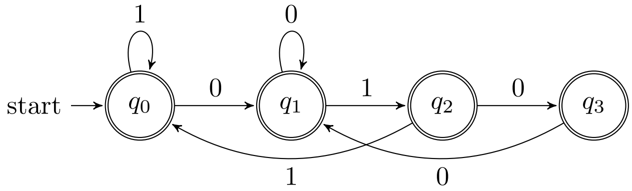
State Final Examination (Sample Questions)

2025-02-04

1 Intersection of a context-free and a regular language (shared topics)

1. Give a formal definition of a *pushdown automaton*.
2. Consider the following languages over the alphabet $\Sigma = \{0, 1\}$:
   - L_1 is the language generated by the context-free grammar $G = (\{S\}, \Sigma, \mathcal{P}, S)$ with production rules $\mathcal{P} = \{ S \to SS \mid 0S1 \mid \epsilon \}$ (where ϵ denotes the empty word),
   - L_2 is the language recognized by the deterministic finite automaton $A = (\{q_0, q_1, q_2, q_3\}, \Sigma, \delta_A, q_0, \{q_0, q_1, q_2, q_3\})$ whose transition function δ_A si given by the following state diagram:



   Construct a pushdown automaton recognizing the intersection $L = L_1 \cap L_2$ of the lanugages L_1 and L_2.

Solution sketch

1. A pushdown automaton is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where
   - Q is a finite, nonempty set of states,
   - Σ is a finite, nonempty set of input symbols,
   - Γ is a finite set of stack symbols,
   - δ is a transition function $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to P_{FIN}(Q \times \Gamma^*)$, where $P_{FIN}(.)$ denotes finite subsets,
   - $q_0 \in Q$ is the initial state,
   - $Z_0 \in \Gamma$ is the bottom of the stack symbol,
   - $F \subseteq Q$ is the set of accepting states.

   When accepting by empty stack, we can omit the set of accepting states F.
2. We can use the standard construction. Accepting by the DFA A is simulated using states, generating by the grammar G is verified using the stack in the same way as when converting a grammar to a (single-state) PDA. The resulting pushdown automaton, accepting by empty stack, is $P = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, S\}, \delta_P, q_0, S)$ where δ_P is defined as follows:

$$\delta_P(q_i, a, a) = \{(\delta_A(q_i, a), \epsilon)\} \text{ for } a \in \{0, 1\}, i \in \{0, 1, 2, 3\}$$
$$\delta_P(q_i, \epsilon, S) = \{(q_i, SS), (q_i, 0S1), (q_i, \epsilon)\} \text{ for } i \in \{0, 1, 2, 3\}$$

   The first rule represents reading an input symbol by the automaton A and matching it with the top of the stack. The second rule simulates one step of the derivation from the grammar G.

(Alternatively, one can directly construct the pushdown automaton recognizing $L_1 \cap L_2$, or construct a context-free grammar generating $L_1 \cap L_2$ and then convert it to a pushdown automaton.)

2 A matrix computation library (shared topics)

Notes on how to answer the question:

- Answer the question using Java, C++, or C#.
- Try to use the latest syntax (and idioms, etc.) of the chosen language.

Imagine you are developing a **library for computing with matrices**.

1. Define a contract (without any implementation) to represent a two-dimensional matrix. The contract provides *means* (i.e., methods, or functions, or properties, etc., based on the chosen language) at least for:
   - getting sizes of the dimensions,
   - getting a value from the matrix based on the coordinates of the value,
   - performing arithmetic operations (adding another matrix, subtracting, multiplying, etc.).
2. Create an implementation of your contract that is suitable for very large but **sparse** matrices (i.e., matrices containing mostly zeros). Comment your implementation so it is clear how it works. From methods (functions, etc.), implement only the ones that return the sizes of the dimensions and a value from the matrix based on the coordinates of the value.

   Define the space complexity of your implementation (in the style $O(\textit{something})$).
3. Imagine you have multiple implementations of your matrix contract (the one for sparse matrices and another for dense matrices). Also, imagine you have a method (or function, etc.) that takes a file with the values of a matrix and returns a suggestion what implementation is the ideal one for the matrix.

   With the help of a suitable *design pattern*, implement a *means* that takes a file and returns an instance of your matrix contract using the ideal implementation.

Solution sketch

1. An interface or abstract class with methods according to the questions, overridden operators (if supported), properties (if supported), etc.
2. There are many possible implementations (for an MxN matrix, a two-dimensional MxN array holding all the values is definitely the wrong implementation). A correct solution is to store only non-zero values (plus some "management" data). Possible solutions: (i) a list holding triplets [row, column, value] and sorted by row index and then by column index, (ii) a list of lists holding tuples [column, value] and sorted, (iii) the compressed sparse row format, i.e., three arrays where one holds nonzero values, the second holds the column in which the corresponding value is located, and the third holds the index (in values and columns) where the given row starts, (iv) etc. Implementation of the getValue method depends on the representations. For CSR, it can be

   ```
   public double getValue(int row, int col) {
       checkBounds(row, col);
       int start = rowPointers[row];
       int end = rowPointers[row + 1];
       for (int idx = start; idx < end; idx++) {
           if (colIndices[idx] == col) {
               return values[idx];
           }
       }
       return 0.0;
   }
   ```

   Complexity depends on the chosen implementation but should always be similar to $O(numberOfNonZeroValues + something)$.

3. It points to the implementation of the *Factory* design pattern, e.g.:

```
public static Matrix create(Path path) {
   return switch (getSuitableType(path)) {
       case "sparse": new SparseMatrix(path);
       case "dense": new DenseMatrix(path);
       default throw new RuntimeException("unknown␣implementation");
   }
}
```

3 Matrices (shared topics)

1. Find a symmetric matrix $S \in \mathbb{R}^{3\times 3}$ such that
$$\mathrm{Ker}(S) = \mathrm{span}\{(1,5,1)^T, (0,2,1)^T\}.$$
2. Define *matrix similarity.* State which of the following quantities similar matrices share: rank, determinant, kernel.
3. The matrix
$$A = \begin{pmatrix} 2 & 2 & 3 \\ 0 & -1 & -4 \\ 1 & 2 & 4 \end{pmatrix}$$
has eigenvalues 3 and 1. Determine whether there exists a basis B such that the matrix of the linear map $f(x) = Ax$, with respect to the basis B, is diagonal.

Solution sketch

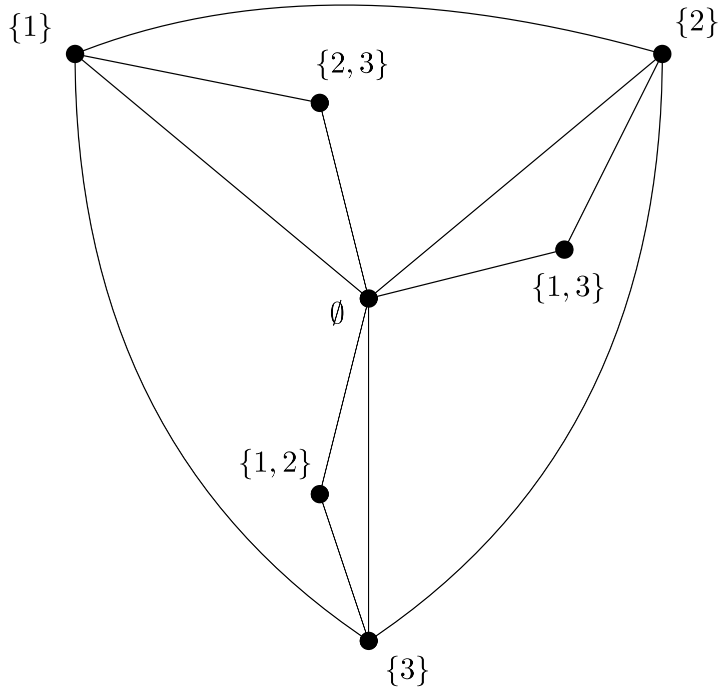
1. The row space of matrix S is orthogonal to its kernel. Thus, the row space of S is spanned by the vector $(3,-1,2)^T$. From the symmetry of S, we obtain that the matrix has the form of
$$S = \alpha(3,-1,2)^T(3,-1,2) = \alpha \begin{pmatrix} 9 & -3 & 6 \\ -3 & 1 & -2 \\ 6 & -2 & 4 \end{pmatrix},$$
where $\alpha \neq 0$.
2. Matrices $A, B \in \mathbb{R}^{n\times n}$ are similar if there is a nonsingular matrix R such that $A = RBR^{-1}$. Similar matrices have the same rank and determinant. However, their kernels may be different; for example, consider the matrices $A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$.
3. No, such a basis does not exist. The eigenvalue 1 has algebraic multiplicity two but geometric multiplicity one (there is only one eigenvector) since $\mathrm{rank}(A - I_3) = 2$. Therefore, matrix A is not diagonalizable.

4 Eulerian graphs (shared topics)

Let n be an integer greater than 1. We define the graph G_n as follows: its vertices are all sets $A \subset \{1,\ldots,n\}$ such that $|A| < n$. Two (distinct) vertices are connected by an edge if and only if the corresponding sets are disjoint. For example, for $n = 3$, we obtain the following graph:

1. For which $n > 1$ is the graph G_n connected?
2. Let $A \subset \{1,\ldots,n\}$ be a set of size less than n. What is the degree of the vertex corresponding to the set A? (Does it depend on the size of the set A, on n, or on something else?)
3. Give a definition of an Eulerian graph and decide for which $n > 1$ is the graph G_n Eulerian?

Provide sufficient justification for your answers.

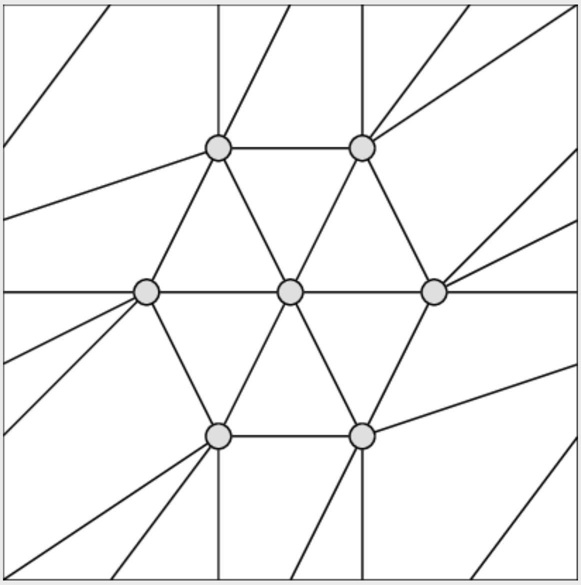


5 Graphs on surfaces (specialization OI-G-PDM, OI-O-PDM, OI-PADS-PDM)

1. Draw K_7 on the torus.
2. Prove that it is not possible to draw K_8 on the torus.
3. Determine the largest possible Euler genus of a surface Σ such that K_5 has a 2-cell drawing on Σ.

Solution sketch

1. One possibility is



2. Generalized Euler's formula implies that every graph G drawn on the torus satisifies $|E(G)| \leq 3|V(G)|$, while K_8 has $28 > 3 \cdot 8$ edges.
3. If a graph G has a 2-cell drawing on a surface of Euler genus g, then generalized Euler's formula states that the number of faces of this drawing is $|E(G)| - |V(G)| - g + 2$. Since the number of faces is at least 1, we get $g \leq |E(G)| - |V(G)| + 1$.

For K_5, this gives $g \leq 6$.

Conversely, any drawing of K_5 with exactly one face necessarily has Euler genus 6. Such a drawing can be obtained e.g. by taking the drawing of K_5 on the projective plane (with 6 faces) and adding 5 crosscaps on its edges so that the faces are joined to a single one.

6 DPLL and matching (specialization UI-SU, UI-ZPJ)

Let $G = (V, E)$ be an undirected graph with n vertices. Perfect matching in a graph is a subset of edges $M \subseteq E$ such that every vertex is incident with exactly one edge of M.

1. Write a formula in propositional logic that is satisfiable if and only if the graph G has a perfect matching. Express the formula in conjunctive normal form (CNF) for a given graph G.
2. Describe the terms *Pure symbol heuristics* and *Unit clause heuristics* and their applications in the DPLL algorithm for finding satisfying assignments of CNF formulas.
3. Write the formula from part 1 for the path on four vertices a, b, c, d. Demonstrate how to use the heuristics from part 2 on this formula to find a satisfying assignment.

7 k-nearest neighbors (specialization UI-SU, UI-ZPJ)

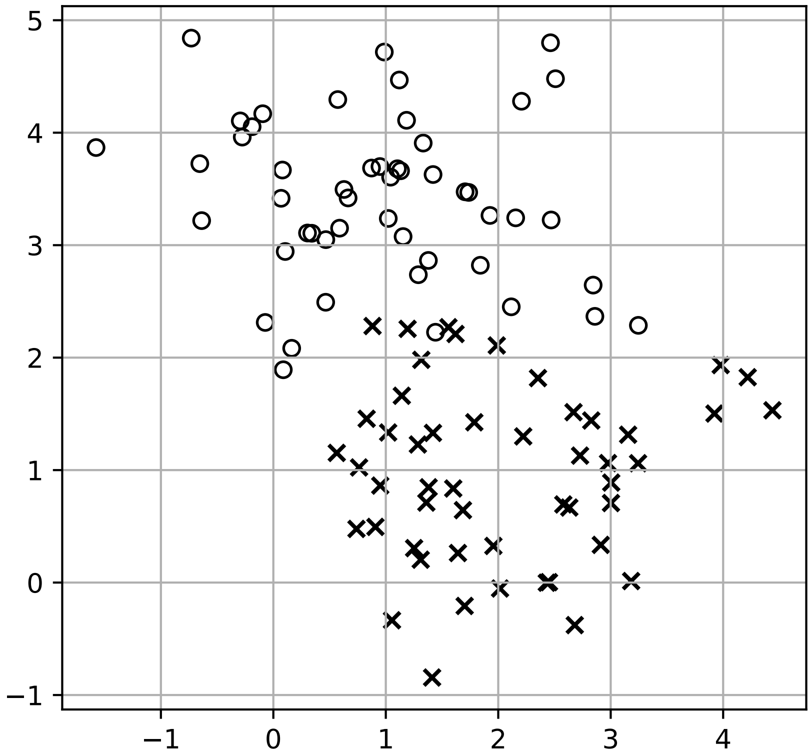
1. Describe the k nearest neighbors method for classification and for regression. How is training performed? How is prediction performed?
2. Consider data with two attributes that we want to classify into two classes ($\circ$, $\times$) shown in the plot below. Assume $k = 3$ (i.e. we use three nearest neighbors). What will be the predicted class for a new input $(3, 2)$? Would the result change, if we change the value of k?
3. Describe how we can find a suitable value of k for our data.

Solution sketch

1. During training, we only save the training data. During classification or regression, we first find the k nearest neighbors. Then, for classification, we predict the most common class for these neighbors, for regression, we take the average of the target values of the neighbors.
2. The new input will be classified as $\circ$, as most of the 3 nearest neighbors belong to that class. If we increase k, many points with class $\times$ will appear among the nearest neighbors, thus changing the prediction (e.g. for $k = 7$).
3. We can use crossvalidation to find a suitable value of k.

8 Logistic regression for binary classification (specialization UI-SU, UI-ZPJ)

1. Describe how we can use the logistic regression model for binary classification. Assume we have a data set with input features $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, and with target values $\boldsymbol{t} = (t_1, \ldots, t_N)$. Then, what is the prediction of the model? How can we independently estimate the error of the model on unseen data?
2. Write the definition of the loss function. Then write its derivative and show how we can use the stochastic gradient descent method for training.
3. When we use the stochastic gradient descent method, how can we prevent overfitting?



Solution sketch

1. Model: $p(C_1) = \sigma(\boldsymbol{x}^T \boldsymbol{w})$, prediction: if (y > 0.5) 1 else 0.

   Error estimate: Split the data into training, validation, and test subsets. Then the validation part is used for hyper-parameters tuning, and the test part is used for reporting the error estimate.

2. Negative log-likelihood loss function: $E(\boldsymbol{w}) = \frac{1}{N} \sum_i -\log(p(C_{t_i} \mid \boldsymbol{x}_i; \boldsymbol{w}))$.

   Reformulate it according to Bernoulli distribution, and derive by $\boldsymbol{w}$.

   Then the derivative is $\nabla_{\boldsymbol{w}} E(\boldsymbol{w}) = (\sigma(\boldsymbol{x}^T \boldsymbol{w}) - t)\boldsymbol{x}$.

   SGD: Initialize weights (either by zeroes, or randomly), then iterate over the data set, in each step subtract $\alpha \cdot \nabla E(\boldsymbol{w})$.

3. Overfitting can be prevented by: L^2 regularization, early stopping (check the validation error, and stop the training process when it increases), suitable learning rate schedule.

9 Bayesian learning (specialization UI-SU)

A supermarket gets boxes of apples from three different providers – half of them from A, one third from B, and one sixth from C. The apples from different providers can have different diameters – 9cm, 10cm, or 11cm. We have randomly selected samples of apples from different providers and measured theirs diameters. The results of this experiment, number of apples of different diameters from each of the providers, are shown in the table below.

Provider	9cm	10cm	11cm
A	5	4	1
B	9	8	3
C	1	8	1

1. Compute the maximum likelihood estimation of the probability distribution of the apple sizes for each provider.
2. Define the term *"maximum likelihood hypothesis"*.
3. We randomly selected two apples from the same box and measured them – one had diameter 10cm and the other one had diameter 11cm. Which of the hypotheses – "The apples are from A.", "The apples are from B.", and "The apples are from C." is the maximum likelihood one? Explain.
4. What is the *Bayesian optimal prediction* of the probability the next apple from the same box will have 10cm diameter (given our obsevations 10cm and 11cm from the same box)? Explain.

Solution sketch

1. The sizes are in discrete intervals. Maximum likelihood estimation for discrete distribution are the relative frequencies, i.e. the probabilities are:

$P(velikost\|h)$	9cm	10cm	11cm
A	.5	.4	.1
B	.45	.4	.15
C	.1	.8	.1

2. The maximum likelihood hypotheses for observation 10 cm and 11 cm is C giving the probability of observation 8% compared to 4% for A and 6% for B. See the third column of the table below.
3. We compute the aposteriori probability of the individual hypotheses and use it to weight their predictions. The prediction for diameter 10cm is 50%. The computation is in the table below.

h	$P(h)$	$P(data\|h)$	$P(h,data)$	$P(h\|data)$	prediction 10cm $*P(h\|data)$
A	.5	.04	.02	3/8	$.4 * 3/8$
B	1/3	.06	.02	3/8	$.4 * 3/8$
C	1/6	.08	.04/3	2/8	$.8 * 2/8$
			$P(data) = .16/3$		$\Sigma = 0.5$