State Final Examination (Sample Questions)

2024-02-09

1 Representing a context-free language (shared topics)

Consider the following language over the alphabet $\{0, 1, \#\}$:

$$L = \{w\#s^R \mid w, s \in \{0,1\}^* \text{ and the word } s \text{ is a subword of the word } w\}$$

(Note: s^R denotes the word s written in reverse; a subword is a contiguous substring, including an empty substring and the whole word.)

1. Give a formal definition of a context-free grammar and a formal definition of a pushdown automaton.
2. Construct a context-free grammar generating the language L.
3. Construct a pushdown automaton accepting the language L.

Solution sketch

1. A *context-free grammar* is $(V, T, \mathcal{P}, S)$, where V and T are finite nonempty disjoint sets, $S \in V$, and $\mathcal{P}$ is a finite set of production rules of the form $H \to \beta$, where $H \in V$ and $\beta \in (V \cup T)^*$.

   A *pushdown automaton* is $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where Q, Σ, and Γ are finite nonempty sets, $\delta\colon Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to P_{FIN}(Q \times \Gamma^*)$, $q_0 \in Q$, $Z_0 \in \Gamma$, and $F \subseteq Q$ (P_{FIN} means finite subsets; F can be omitted if accepting by empty stack).
2. For example, the language L is generated by the context-free grammar $G = (\{S, X, A\}, \{0, 1, \#\}, \mathcal{P}, S)$ with production rules $\mathcal{P} = \{S \to AX, X \to 0X0 \mid 1X1 \mid A\#, A \to 0A \mid 1A \mid \lambda\}$ (where the variable A can generate an arbitrary word over $\{0, 1\}$).
3. The automaton can be constructed by converting the grammar G. The standard procedure results in the pushdown automaton $(\{q_0\}, \{0, 1, \#\}, \{0, 1, \#, S, X, A\}, \delta, q_0, S)$ accepting the language L by empty stack, where the transition function consists of transitions corresponding to production rules and transitions for reading input symbols:

   $$\delta = \{((q_0, \lambda, H), \beta) \mid H \to \beta \in \mathcal{P}\} \cup \{((q_0, a, a), \lambda) \mid a \in \{0, 1, \#\}\}$$

   Alternatively, the automaton can be constructed directly. While reading the word w we store its symbols in the stack. After reading $\#$, we first delete an arbitrary number of symbols from the stack, then we read s^R checking if it agrees with the stack, and finally we empty the stack. It is also possible to construct an automaton accepting by final state.

2 Disk driver (shared topics)

Implement functions of a disk driver for reading a single block.

The disk is controlled via memory mapped registers, individual commands (and their parameters) are written and device status can be read to and from these registers (assume no interrupt support, service requires active waiting).

Offset	Register	Type	Description
0	Status	R	Bit field with current status of the controller (read only, writes are ignored). 31 | 29 | ... | 2 | 1 | 0 0 | 0 | ... | 0 | BUSY | ERR `BUSY` – Set when command is written, cleared when the operation completes. `ERR` – Set when an error occured.
4	Size	R	Disk size in blocks (read only, writes are ignored).
8	Command	W	Writing to the register initiates disk operation (`1` read, `2` write).
12	LBA	W	Logical block address for operation issued through *Command*.
16	DMA	W	Physical memory address for storing data read from the disk.

Blocks are addressed in the usual way using LBA, individual blocks have a fixed size of 512 bytes. The data transfer during reading or writing uses DMA (from the point of view of this question it is therefore sufficient to configure the physical address for storing of the result, actual data transfer is completely managed by the disk controller). When the operation completes, the status register is updated accordingly (we assume 32-bit system, disk size is limited to 2 TB).

Write the implementation of the two functions below and design a data structure for storing the information about the disk device in the system (assume several disks can be attached to the same system, the operating system kernel will distinguish them using different instances of the `disk_t` structure). Both functions return `true` when the operation is succesful, otherwise `false` (without further details).

Your implementation must address the posibility of multiple processes accessing the disk concurrently, trivial solutions are accepted.

```
typedef struct {  ...  } disk_t;

bool disk_init(disk_t *disk, uint32_t register_address) {  ...  }

bool disk_read_block_waiting(disk_t *disk, size_t lba, uint32_t data_phys_addr) {  ...  }
```

Solution sketch Source code sketch (without symbolic constants and comments):

```
typedef volatile struct {
    uint32_t status;
    uint32_t size;
    uint32_t command;
    uint32_t lba;
    uint32_t dma;
} disk_regs_t;

typedef struct {
    size_t max_lba;
    disk_regs_t *ctl;
    mutex_t mutex;
} disk_t;

static bool disk_is_ok(disk_t *disk) { return disk->ctl->status & 1 == 0; }
static bool disk_is_ready(disk_t *disk) { return disk->ctl->status & 2 == 0; }
static bool disk_wait_for_ready(disk_t *disk) {
    while (!disk_is_ready(disk)) {
        if (!disk_is_ok(disk)) {
            return false;
        }
   }
   return true;
}

bool disk_init(disk_t *disk, uint32_t register_address) {
```

```
    disk->ctl = (disk_regs_t *) register_address;
    disk->max_lba = disk->ctl->size;
    mutex_init(&disk->mutex);
    return disk_wait_for_ready(disk);
}

bool disk_read_block_waiting(disk_t *disk, size_t lba, uint32_t data_phys_addr) {
    if (lba >= disk->max_lba) {
        return false;
    }
    mutex_lock(&disk->mutex);
    bool ok = disk_wait_for_ready(disk);
    if (!ok) {
        mutex_unlock(&disk->mutex);
        return false;
    }

    disk->ctl->lba = lba;
    disk->ctl->dma = data_phys_addr;
    disk->ctl->command = 1;

    ok = disk_wait_for_ready(disk);
    mutex_unlock(&disk->mutex);

    return ok;
}
```

3 Inner product (shared topics)

1. Verify that $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \operatorname{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B})$ is an inner product on $\mathbb{R}^{m\times n}$, whereas the so called *trace od a matrix* denoted by trace is defined for square matrices of order n by $\operatorname{trace}(\boldsymbol{C}) = \sum_{i=1}^{n} c_{ii}$.
2. Decide whether the following matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{4\times 6}$ are mutually orthogonal with respect to the given inner product, where $\boldsymbol{A} = \begin{pmatrix} 1 & -2 & 3 & -4 & 5 & -6 \\ -2 & 3 & -4 & 5 & -6 & 7 \\ 3 & -4 & 5 & -6 & 7 & -8 \\ -4 & 5 & -6 & 7 & -8 & 9 \end{pmatrix}$ and $\boldsymbol{B}$ is the matrix with all entries equal to -13.
3. State Cauchy–Schwarz inequeality and decide whether for a square matrix $\boldsymbol{A}$ of order n it holds that : $(\operatorname{trace}(\boldsymbol{A}))^2 \leq n \cdot \operatorname{trace}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$.

4 Discontinuous function (shared topics)

Let $f : (0,1) \to \mathbb{R}$ be a real function.

1. Define what it means that f is *discontinuous* in the point $\frac{1}{2}$.
2. Is the function f, when defined on this interval by $f(x) = \frac{2x-1}{1-2x}$ for $x \neq \frac{1}{2}$ and by $f(\frac{1}{2}) = -1$, discontinuous in the point $\frac{1}{2}$?
3. Suppose we define f by the formula $f(x) = \frac{2x-1}{1-2x}$ on the whole interval $(0,1)$. Is f discontinuous in $x = \frac{1}{2}$?

Justify your answers in parts 2 and 3.

5 Statistical tests – Student's t-test (specialization UI-SU)

A cookie producer makes cookies with declared weight of 100 grams. During quality control 100 cookies were randomly selected and weighted. Their average weight was 102 grams with standard deviation of 2 grams. We want to use the t-test to statistically evaluate, if the average weight is different from the declared weight.

1. What is the null and alternative hypothesis of the one-sample t-test?
2. Compute the value of the test statistic for the data given above. What probabilistic distribution does it have?
3. Is the weight of the cookies statistically significantly different from the declared weight of the cookies? *Hint:* The critical value of the test statistic at the significance level $\alpha = 0.05$ is approx. 2.

Solution sketch

1. The null hypothesis is $H_0 : \bar{X} = \mu$, where $\bar{X}$ is the average weight of the cookies and μ is the tested mean. Alternative hypothesis is $H_1 : \bar{X} \neq \mu$.
2. $t = \frac{102-100}{2/\sqrt{100}} = 10$. It has t-distribution with 99 degrees of freedom.
3. The computed t-value is greater than the critical value, we therefore reject the null hypothesis. The weight of the cookies differs significantly from the declared weight.

6 Linear regression and regularization (specialization UI-SU)

We want to fit linear model $y = \beta_1 x_1 + \beta_2 x_2 + \beta_0$ to the data below. We want to use the gradient method and L2 regularization. We will use the learning rate $\alpha = 0.1$ and regularization constant $\lambda = 0.5$.

x_1	x_2	y
0	2	3
2	3	9
1	1	2
2	0	2

1. Define the loss function for the model given above (residual sum of squares with L2 regularization).
2. Assume that the current estimate of the model coefficients is $\beta_1 = 1$, $\beta_2 = 3$, $\beta_0 = -1$. Perform a single step of the gradient method. What are the new values of the coefficients?

Solution sketch

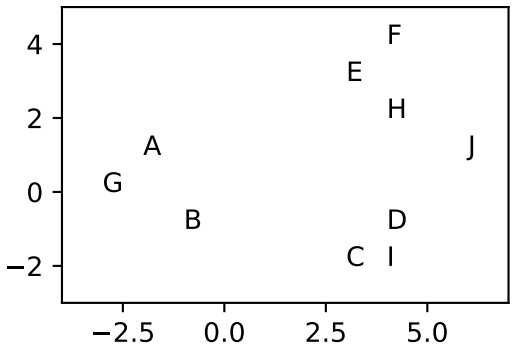
1. The loss function is RSS $= \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, for L2 regularization we also add $\lambda \sum_{i=1}^{m} \beta_i^2$ (N is the number of training samples, $m + 1$ is the number of model parameters, $\hat{y}_i$ is the output of the model for input i). The β_0 coefficient is typically not considered in regularization.
2. We need to compute the gradient of the loss function w.r.t. the parameters of the model. We will count it for a single training sample – the resulting gradient is a sum over all the training samples. The derivation $\frac{dRSS}{d\beta_0} = -2(y_i - \hat{y}_i)$, the derivation $\frac{d\text{RSS}}{d\beta_i} = -2(y_i - \hat{y}_i)x_i$. We also have to add the derivation of the regularization term w.r.t. β_i – it is $\lambda 2\beta_i = \beta_i$. Evaluating for the given values, we get that the derivation of RSS w.r.t. β_0 is 4 + 2 + 2 + (-2) = 6, derivation w.r.t. β_1 is 2, and derivation w.r.t. β_2 is 16. Adding the derivation of the regularization term gives us the derivation of the full loss function. It is 3 w.r.t β_1 and 19 w.r.t β_2. The coefficients after this step are thus $\beta_0 = \beta_0 - \alpha \cdot 6 = -1.6$, $\beta_1 = \beta_1 - \alpha \cdot 3 = 0.7$, and $\beta_2 = \beta_2 - \alpha \cdot 19 = 1.1$.

7 Clustering (specialization UI-SU)

We have data given in the table below and graphically shown in the figure on the right. We want to cluster them using the k-means algorithm.

Note: The names are not part of the data – they are used only for the visualization on the right and to simplify discussion in your answers.

Name	x	y
A	-2	1
B	-1	-1
C	3	-2
D	4	-1
E	3	3
F	4	4
G	-3	0
H	4	2
I	4	-2
J	6	1



1. Describe briefly the steps of the k-means algorithm.
2. Assume that in the first step we have chosen the points A, F, and J as the initial cluster centers. What will be the initial division of the given points into clusters. What will be the new cluster centers? Use the Euclidean distance in your calculations.
3. Will the cluster centers change in the following iterations of the algorithm?

Solution sketch

1. The algorithm first randomly chooses the initial cluster centers. Then it repeats two steps: assigning each of the points to the closest cluster center and re-evaluating the cluster centers (as an average of the points in the cluster).
2. One cluster will contain points $\{A, G, B\}$, another one will contain points $\{E, F, H\}$, and the last one will contain $\{J, D, I, C\}$. New cluster centers will be (respectively) $(-2, 0)$, $(\frac{11}{3}, 3)$, and $(\frac{17}{4}, -1)$.
3. The assignment of points to clusters will not change in the following iteration, so the cluster centers will also not change.

8 Deep Learning in NLP (specialization UI-ZPJ)

1. Describe the main techniques for training of word embeddings (skipgram, CBOW).
2. Describe the main models of recurrent neural networks used in natural language processing (vanilla recurrent neural network, LSTM). What kind of problems appear while training vanilla recurrent neural networks? How do LSTM networks solve these problems?

Solution sketch

1. The main idea of CBOW is predicting the middle word based on the context. The main idea of skipgram is the predicting the surrounding words based on the middle word.
2. The main problem with training of vanilla recurrent neural networks are so called exploding and vanishing gradients. LSMT networks solve this problem by replacing individual neurons with so called LSTM cells that work with the internal state explicitly.
3. Exact formulas for both sub-questions can be found in the 9^{th} lecture of the NPFL124 course (Natural Language Processing).

9 Linear regression and regularization (specialization UI-ZPJ)

We want to fit linear model $y = \beta_1 x_1 + \beta_2 x_2 + \beta_0$ to the data below. We want to use the gradient method and L2 regularization. We will use the learning rate $\alpha = 0.1$ and regularization constant $\lambda = 0.5$.

1. Define the loss function for the model given above (residual sum of squares with L2 regularization).

x_1	x_2	y
0	2	3
2	3	9
1	1	2
2	0	2

2. Assume that the current estimate of the model coefficients is $\beta_1 = 1$, $\beta_2 = 3$, $\beta_0 = -1$. Perform a single step of the gradient method. What are the new values of the coefficients?

Solution sketch

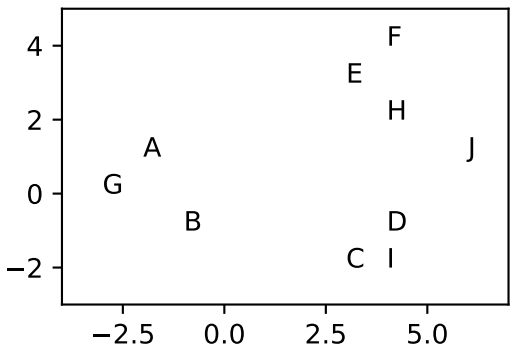
1. The loss function is RSS $= \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$, for L2 regularization we also add $\lambda \sum_{i=1}^{m} \beta_i^2$ (N is the number of training samples, $m+1$ is the number of model parameters, $\hat{y}_i$ is the output of the model for input i). The β_0 coefficient is typically not considered in regularization.
2. We need to compute the gradient of the loss function w.r.t. the parameters of the model. We will count it for a single training sample – the resulting gradient is a sum over all the training samples. The derivation $\frac{dRSS}{d\beta_0} = -2(y_i - \hat{y}_i)$, the derivation $\frac{d\text{RSS}}{d\beta_i} = -2(y_i - \hat{y}_i)x_i$. We also have to add the derivation of the regularization term w.r.t. β_i – it is $\lambda 2\beta_i = \beta_i$. Evaluating for the given values, we get that the derivation of RSS w.r.t. β_0 is 4 + 2 + 2 + (-2) = 6, derivation w.r.t. β_1 is 2, and derivation w.r.t. β_2 is 16. Adding the derivation of the regularization term gives us the derivation of the full loss function. It is 3 w.r.t β_1 and 19 w.r.t β_2. The coefficients after this step are thus $\beta_0 = \beta_0 - \alpha \cdot 6 = -1.6$, $\beta_1 = \beta_1 - \alpha \cdot 3 = 0.7$, and $\beta_2 = \beta_2 - \alpha \cdot 19 = 1.1$.

10 Clustering (specialization UI-ZPJ)

We have data given in the table below and graphically shown in the figure on the right. We want to cluster them using the k-means algorithm.

Note: The names are not part of the data – they are used only for the visualization on the right and to simplify discussion in your answers.

Name	x	y
A	-2	1
B	-1	-1
C	3	-2
D	4	-1
E	3	3
F	4	4
G	-3	0
H	4	2
I	4	-2
J	6	1



1. Describe briefly the steps of the k-means algorithm.
2. Assume that in the first step we have chosen the points A, F, and J as the initial cluster centers. What will be the initial division of the given points into clusters. What will be the new cluster centers? Use the Euclidean distance in your calculations.
3. Will the cluster centers change in the following iterations of the algorithm?

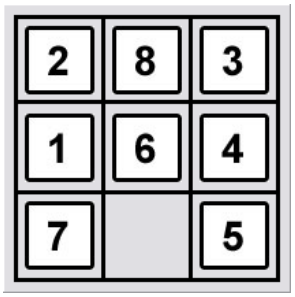
Solution sketch

1. The algorithm first randomly chooses the initial cluster centers. Then it repeats two steps: assigning each of the points to the closest cluster center and re-evaluating the cluster centers (as an average of the points in the cluster).

2. One cluster will contain points $\{A, G, B\}$, another one will contain points $\{E, F, H\}$, and the last one will contain $\{J, D, I, C\}$. New cluster centers will be (respectively) $(-2, 0)$, $(\frac{11}{3}, 3)$, and $(\frac{17}{4}, -1)$.
3. The assignment of points to clusters will not change in the following iteration, so the cluster centers will also not change.

11 Informed search (specialization UI-SU, UI-ZPJ)

Let's consider the puzzle in the picture below. It is a 3×3 grid with 8 labeled tiles and one empty space. Our goal is to rearrange the tiles from the given initial state to a predetermined goal state, where the tiles are arranged from 1 to 8 from left to right, top to bottom. At each step, we can slide exactly one tile to the adjacent empty space. Additionally, we want to perform the minimum possible number of steps.



1. Formalize the given problem as an informed search problem. Specifically, describe a state, transition model, and the goal test.
2. Propose an optimal informed search algorithm to solve the defined problem. Describe the proposed algorithm, for example by a pseudocode.
3. Propose a heuristic function for the puzzle. What properties does the heuristic function have? Does it ensure that the proposed algorithm finds an optimal solution? *Hint: Did you write a tree search or a graph search?*