СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички листове.
- Пишете само на предоставените листове, без да ги разкопчавате.
- Решението на една задача трябва да бъде на същия лист, на който е и нейното условие (т.е. може да пишете отпред и отзад на листа със задачата, но не и на лист на друга задача).
- Ако имате нужда от допълнителен лист, можете да поискате от квесторите.
- На един лист не може да има едновременно и чернова, и белова.
- Черновите трябва да се маркират, като най-отгоре на листа напишете “ЧЕРНОВА”.
- Ако решението на една задача не се побира на нейния лист, трябва да поискате нов бял лист от квесторите. Той трябва да се защипе с телбод към листа със задачата.
- Всеки от допълнителните листове (белова или чернова) трябва да се надпише най-отгоре с вашия факултетен номер.
- Черновите също се предават и се защипват в края на работата.
- Времето за работа по изпита е 3 часа.

Изпитната комисия ви пожелава успешна работа!

Задача 1.

Задачата да се реши на езика C++. В подточките, в които се изисква да се посочи какво ще се изведе, точки се дават само ако отговорът напълно съвпада с това, което извежда съответният код.

1) Дадена е дефиницията:

```
void print(int* arr, int size) {
  for (int i = 0; i < size; ++i)
    cout << arr[i] << ";";
  cout << "\n";
}
```

Какво ще изведе на стандартния изход даденият по-долу фрагмент? Всеки отделен ред от изхода да се попълни в съответното поле.

```
const int size = 5;
int arr[] = { 5, 10, -1, 5, 6 };
for (int i = 0; i < size; ++i) {
  for (int j = 1; j < size; ++j) {
    if (arr[j-1] < arr[j]) {
      swap(arr[j-1], arr[j]);
    }
  }
  print(arr, size);
}
```

На ред 1 извежда: _____
На ред 2 извежда: _____
На ред 3 извежда: _____
На ред 4 извежда: _____
На ред 5 извежда: _____

2) Какво трябва да се попълни на мястото на коментара в `toUpper` така, че ако на функцията се подаде малка латинска буква, тя да връща съответната ѝ главна, а всички други символи да връща непроменени?

```
char toUpper(char c) {
  return /* какво трябва да има тук? */ ;
}
```

А) `(c >= 'a' || c <= 'z') ? (c - 'a' + 'A') : c`
Б) `(c >= 'a' || c <= 'z') ? (c + 'A') : c`
В) `(c >= 'a' || c <= 'z') ? (c - 'a') : c`
Г) `(c >= 'a' && c <= 'z') ? (c - 'a' + 'A') : c`[1]
Д) `(c >= 'a' && c <= 'z') ? (c + 'A') : c`[1]
Е) `(c >= 'a' && c <= 'z') ? (c - 'a') : c`[1]

3) Да се попълнят празните места в дефинициите на функциите така, че `isOdd` да връща истина тогава и само тогава, когато подаденото ѝ число е нечетно, а `isEven` — когато то е четно.

```
bool isEven(unsigned int);
bool isOdd(unsigned int n) {
  if (n == 0) return ________;
  if (n == 1) return ________;
  return isEven(________);
}
bool isEven(unsigned int n) {
  if (n == 0) return ________;
  if (n == 1) return ________;
  return isOdd(_________);
}
```

4) Под всеки от дадените по-долу фрагменти да се посочи какво ще изведе той на стандартния изход.

```
for (int i = 0; i < 9; ++i)
  cout << (1 << i) << ";";
```

_____

```
switch(5 % 2) {
case 0:
  std::cout << "0";
case 1:
  std::cout << "1";
default:
  std::cout << "x";
}
```

_____

```
for (int i = 0; i < 5; ++i) {
  if ( ! (i - 3))
    continue;
  std::cout << i;
}
```

_____

```
int arr[] = { 1, 2, 3, 4, 0 };
*(arr + 2) *= 10;
for (int* p = arr; *p; p++)
  std::cout << *p << ";";
```

_____

[1] В дадената на изпита тема на тези отговори беше допусната печатна грешка, която тук е отстранена.

Критерии за оценяване

В подточките, в които се изисква да се посочи какво ще се изведе, точки се дават само ако отговорът напълно съвпада с това, което извежда съответният код. Ако изходът от реда не съвпада с това, което е посочено в отговора, той се оценява с 0 т. Сумата от точките се закръгля до цяло число.

По-конкретно за съответните подточки:

1) Всеки напълно коректно посочен ред носи 0,4 т. (общо 2 т.). Изходът е както следва:

а) На ред 1 извежда: `10;5;5;6;-1;`
б) На ред 2 извежда: `10;5;6;5;-1;`
в) На ред 3 извежда: `10;6;5;5;-1;`
г) На ред 4 извежда: `10;6;5;5;-1;`
д) На ред 5 извежда: `10;6;5;5;-1;`

2) Носи 1 т. ако е посочен коректният отговор и 0 т. в противен случай. Коректният отговор е:

```
(c >= 'a' && c <= 'z') ? (c-'a'+'A') : c
```

3) Всяко напълно коректно попълнено място носи 0,5 т. (общо 3 т.). Ако в някой отговор вместо булевите литерали `true` и `false` са използвани 0 или 1, точките за отговора се намаляват наполовина. Примерно решение:

```
bool isEven(unsigned int);
bool isOdd(unsigned int n) {
  if (n == 0) return false;
  if (n == 1) return true;
  return isEven(n - 1);
}
bool isEven(unsigned int n) {
  if (n == 0) return true;
  if (n == 1) return false;
  return isOdd(n - 1);
}
```

4) Всеки напълно коректно отговорен фрагмент носи 1 т. (общо 4 т.). Изходът от фрагментите е посочен по-долу:

```
for (int i = 0; i < 9; ++i)
  cout << (1 << i) << ";";
```

```
1;2;4;8;16;32;64;128;256;
```

```
switch(5 % 2) {
case 0:
  std::cout << "0";
case 1:
  std::cout << "1";
default:
  std::cout << "x";
}
```

```
1x
```

```
for (int i = 0; i < 5; ++i) {
  if ( ! (i - 3))
    continue;
  std::cout << i;
}
```

```
0124
```

```
int arr[] = { 1, 2, 3, 4, 0 };
*(arr + 2) *= 10;
for (int* p = arr; *p; p++)
  std::cout << *p << ";";
```

```
1;2;30;4;
```

Задача 2.

Задачата да се реши на езика C++. Дадени са дефинициите:

```
class A {
public:
  A()   { cout << "A()\n"; }
  A(A&) { cout << "A(A&)\n"; }
  virtual ~A() { cout << "~A()\n"; }
  A& operator=(A&) {
    cout << "op=(A&)\n";
    return *this;
  }
};

class B : public A {
public:
  B()   { cout << "B()\n"; }
  B(B&) { cout << "B(B&)\n"; }
  virtual ~B() { cout << "~B()\n"; }
  B& operator=(B&) {
    cout << "op=(B&)\n";
    return *this;
  }
};

void f(A b) { cout << "f(A)\n"; }
```

Под всеки от редовете на дадения вдясно програмен фрагмент да се посочи какво ще се изведе в резултат от неговото изпълнение. (Между редовете нарочно е оставено повече място, за да може да попълните отговора си) Ако смятате, че някой ред няма да изведе нищо, напишете "не извежда нищо". Ако смятате, че някой от редовете ще предизвика грешка, напишете "грешка" и обяснете каква е тя и защо възниква.

За коректни се считат отговорите, които напълно съответстват на това, което ще се случи за съответния ред. Текст "грешка" без обяснение носи нула точки.

```
B d;


B copy = d;


A b = d;


A& ref = d;


B arr[2];


f(d);


A* p = new B(d);


delete p;


d = d;


ref = d;
```

Критерии за оценяване

- Всеки ред, за който напълно коректно е посочено какво ще се изведе/случи, носи 1 т.
- Ако изходът от реда не съвпада с това, което е посочено в отговора, той се оценява с 0 т.
- Текст “Грешка” без обяснение към него се оценява с 0 т.

Изходът от редовете е посочен по-долу:

```
Изход от "B d":
  A()
  B()

Изход от "B copy = d":
  A()
  B(B&)

Изход от "A b = d":
  A(A&)

Изход от "A& ref = d":
  не извежда нищо

Изход от "B arr[2]":
  A()
  B()
  A()
  B()

Изход от "f(d)":
  A(A&)
  f(A)
  ~A()

Изход от "A* p = new B(d)":
  A()
  B(B&)

Изход от "delete p":
  ~B()
  ~A()

Изход от "d = d":
  op=(B&)

Изход от "ref = d":
  op=(A&)
```

Задача 3. Извършват се последователни опити. При всеки опит се хвърлят четири правилни монети. Считаме, че опитът е успешен, ако при него се паднат равен брой "лица" и "гербове".

а) Да се определи вероятността един опит да е успешен.

б) Каква е вероятността да са се случили точно три неуспешни опита преди втори успешен опит?

в) Нека X е броят на успешните опити до първия неуспешен опит. Да се пресметнат математическото очакване $\mathrm{E}X$ и дисперсията $\mathrm{D}X$.

Примерно решение:

а) (3 точки) Хвърлят се четири монети. Вероятността за падане на герб на всяка от тях е $\frac{1}{2}$. Броя паднали се "гербове" да съвпадне с броя на падналите се "лица", означава да се паднат точно два герба и две лица. От четирите монети избираме двете върху който да са гербовете по $\binom{4}{2}$ начина. Вероятността да се падне герб на две фиксирани монети е $\left(\frac{1}{2}\right)^2$. Аналогично, вероятността за лице на останалите две монети е $\left(\frac{1}{2}\right)^2$. Окончателно за вероятността за успех p получаваме:

$$p = \binom{4}{2}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^2 = \frac{3}{8}$$

За намиране на тази вероятност може да се използва и биномно разпределение. Да означим с Z броя на падналите се гербове при хвърляне на четири монети. Не е трудно да се разпознае, че случайната величина Z е биномно разпределена $Z \in Bi(4, 1/2)$ и търсената вероятност е $p = P(Z = 2)$.

б) (3 точки) Съгласно предишната подточка вероятността за успех е $p = \frac{3}{8}$, съответно за неуспех $q = \frac{5}{8}$. За да има точно три неуспеха преди втория успех явно са проведени пет опита, при което на последния опит има успех. Тогава трябва да намерим вероятността на събитието — при първите четири опита има точно един успех и последния опит е успех. За първите четири опита използваме отново биномна вероятност, а при последния опит вероятността за успех е $\frac{3}{8}$. Така получаваме:

$$\binom{4}{1}pq^3\,p = \binom{4}{1}\left(\frac{3}{8}\right)^2\left(\frac{5}{8}\right)^3 = \frac{1125}{8192}$$

в) (4 точки) Случайната величина X отговаря на модела на геометричното разпределение, единствената разлика е, че в стандартния случай се търси броя на неуспехите до първия успех, а в тази задача търсим броя на успехите до първия неуспех. Тогава $X \in Ge\left(\frac{5}{8}\right)$. За намиране на очакването и дисперсията се използват директно (не е нужно да се извеждат) формулите, известни от свойствата на геометрично разпределение. При означенията въведени по-горе:

$$\mathrm{E}X = \frac{p}{q} = \frac{3/8}{5/8} = \frac{3}{5},$$

$$\mathrm{D}X = \frac{p}{q^2} = \frac{3/8}{(5/8)^2} = \frac{24}{25}.$$

Задача 4. Софтуерна система за отдаване на автомобили под наем функционира при следните правила:

1. Автомобили под наем могат да взимат само лица, навършили 23 години.
2. Автомобили под наем не могат да взимат лица с пътнотранспортни нарушения.
3. При бизнес пътуване може да се ползва отстъпка от цената.

Да се дефинира таблица за вземане на решение, въз основа на която могат да се генерират тестови сценарии. Таблицата трябва да показва условията, следствията от тях и правилата, от които се генерират тестовите сценарии. Въз основа на таблицата за вземане на решение определете резултата от изпълнението на следните тестови сценарии:

TC1: Лице на 26 години с пътнотранспортни нарушения и пътуване с бизнес цел.

TC2: 62-годишен турист без пътнотранспортни нарушения.

Примерно решение:

Стъпка 1: Идентифицират се следните условия и следствия:

- C_1: Лице над 23 години
- C_2: Липса на пътнотранспортни нарушения
- C_3: Бизнес пътуване
- E_1: Наемане на автомобил
- E_2: Ползване на отстъпка

Стъпка 2: Условията и следствията се изброяват в таблица на решенията.
Стъпка 3: Броят на комбинациите е $2^3 = 8$.
Стъпка 4: Колоните на таблицата се попълват като се има предвид, че RF1 = RF2 = 8/2 = 4, RF3 = 4/2 = 2, RF1 = 2/2 = 1.
Стъпка 5: Определяне на следствията от всяко правило.
Стъпка 6: Определя се очаквания резултат от изпълнение на всяко следствие.
Стъпка 7: Редуцират се правилата

Условия	**Стойности**	**1**	**2**	**3**	**4**
C_1	T, F, N/A	F	T	T	T
C_2	T, F, N/A	N/A	F	T	T
C_3	T, F, N/A	N/A	N/A	F	T
Действия					
E_2	T, F, N/A	F	F	T	T
E_2	T, F, N/A	F	F	F	T

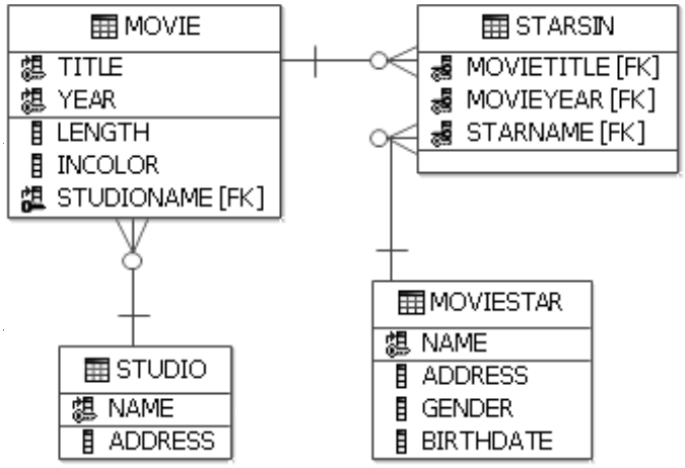
Стъпка 8: Генерират се тестови сценарии, съответстващи на правилата в таблицата.

Резултат TC1: Не наема кола.
Резултат TC2: Наема кола без отстъпка.

Критерии за оценяване:

- Дефинирани условия и следствия: 1 т.
- Дефинирани стъпки за създаване на таблица за вземане на решения: 3 т.
- Създадена таблица за вземане на решения с редуцирани правила: 4 т.
- Определяне на резултати от примерните тестови сценарии: 2 т.

Задача 5. Дадена е базата от данни `Movies`, в която се съхранява информация за филми, филмови студиа, които ги произвеждат, както и актьорите, които участват в тях.



Таблицата `Studio` съдържа информация за филмови студиа:

- `name` — име, първичен ключ
- `address` — адрес;

Таблицата `Movie` съдържа информация за филми. Атрибутите `title` и `year` заедно формират първичния ключ.

- `title` — заглавие
- `year` — година, в която е заснет филмът
- `length` — дължина в минути
- `incolor` — `'Y'` за цветен филм и `'N'` за чернобял
- `studioname` — име на студио, външен ключ към `Studio.name`;

Таблицата `MovieStar` съдържа информация за филмови звезди:

- `name` — име, първичен ключ
- `address` — адрес
- `gender` — пол, `'M'` за мъж (актьор) и `'F'` за жена (актриса)
- `birthdate` — рождена дата.

Таблицата `StarsIn` съдържа информация за участието на филмовите звезди във филмите. Трите атрибута заедно формират първичния ключ. Атрибутите `movietitle` и `movieyear` образуват външен ключ към `Movie`.

- `movietitle` — заглавие на филма
- `movieyear` — година на заснемане на филма
- `starname` — име на филмовата звезда, външен ключ към `MovieStar.name`.

1) Да се попълнят празните места в следната заявка така, че тя да извежда името на студиото на филма 'The Usual Suspects' и заглавията на всички филми на същото студио:

```
SELECT s.name, m.title
FROM movie ____ JOIN studio s ON m.studioname ____________________
WHERE s.name ____ (SELECT _________________
                   FROM   _________________
                   WHERE title = 'The Usual Suspects' AND year = 1995);
```

2) Да се посочи коя от следните заявки извежда имената на филмовите звезди, за които няма информация в кои филми са играли:

```
А) SELECT DISTINCT starname
   FROM starsin
   GROUP BY starname
   HAVING COUNT(*) = 0;
```

```
Б) SELECT ms.name, si.movietitle
    FROM moviestar ms
    LEFT JOIN starsin si
           ON ms.name=si.starname
    WHERE si.movietitle IS NULL;
```

```
В) SELECT name
   FROM starsin
   JOIN moviestar ON starname = name
   GROUP BY name
   HAVING COUNT(name) = 0;
```

```
Г) SELECT name
   FROM moviestar
   WHERE NOT EXISTS (SELECT starname
                     FROM starsin);
```

Критерии за оценяване

а) Общо 5 т., по ред на празните слотове:
- `m` или `AS m` — **1 т.**
- `=s.name` или `= name` — **1 т.**
- `=` или `IN` — **1 т.**
- `studioname` — **1 т.**
- `movie` — **1 т.**

б) Единственият верен отговор е Б). Посочването на този отговор носи 5 т., а посочването на грешен отговор или комбинация от отговори носи 0 т.

Заявката, описана в този отговор, коректно извежда името на филмовите звезди, за които няма информация в кои филми са играли, съгласно условието на тази подточка. Фактът, че в допълнение на исканата в условието информация заявката извежда също и втора колона с име `movietitle` и стойности `NULL`, не променя коректността на заявката относно зададеното така условие.

Заявките, посочени във всички останали отговори, не отговарят на условието на подточката. В частност, заявките в отговор А) и отговор В) винаги връщат празно множество, независимо дали има или не филмови звезди, за които няма информация в кои филми са играли, както би трябвало да е по условие. Заявката в отговор Г) не отговаря на условието, понеже в случая, в който таблицата `starsin` не е празна, не се извежда името на нито една филмова звезда, вместо да се изведат имената на тези, за които няма информация в кои филми са играли, както би трябвало да е по условие. Това поведение е съгласно стандарта за езика SQL и не зависи от конкретната реализация в система за управление на бази от данни.

Задача 6. Да се създаде XSL трансформация, която да преобразува началния XML документ в лявата колона на долната таблица в резултатен XML документ, при което:

1. кодирането на резултатния документ е UTF-8, форматът му е XML, а името на корена му е `INVENTORY2`;
2. всеки един елемент `ITEM` от началния XML документ със стойност на атрибута `PRICE` по-голяма от 100 се копира в резултатния документ с атрибутите и поделементите си без изменение;
3. всеки един елемент `ITEM` от началния XML документ със стойност на атрибута PRICE по-малка или равна на 100 се копира в резултатния документ с име `ITEM_CONVERTED`, при което:
   а) атрибутите на този елемент `ITEM` се преобразуват до поделементи на елемента `ITEM_CONVERTED`, при което имената и съдържанието им се запазват;
   б) поделементите на този елемент `ITEM` се преобразуват до атрибути на елемента `ITEM_CONVERTED`, при което имената и съдържанието им се запазват.

Начален XML документ	**Примерен резултат след XSL трансформация**

`<?xml version="1.0"?>`
`<INVENTORY>`
`  <ITEM ID="123" ORDERED="22" PRICE="99">`
`  <SERIALNUMBER>123456</SERIALNUMBER>`
`  <LOCATION>Sofia</LOCATION>`
`  <NAME>Valve</NAME>`
`  </ITEM>`
`  <ITEM ID="321" ORDERED="33" PRICE="101">`
`  <SERIALNUMBER>654321</SERIALNUMBER>`
`  <LOCATION>Plovdiv</LOCATION>`
`  <NAME>Piston</NAME>`
`  </ITEM>`
`</INVENTORY>`
	`<INVENTORY2>`
	`  <ITEM_CONVERTED SERIALNUMBER="123456"`
	`  LOCATION="Sofia" NAME="Valve">`
	`  <ID>123</ID>`
	`  <ORDERED>22</ORDERED>`
	`  <PRICE>99</PRICE>`
	`  </ITEM_CONVERTED>`
	`  <ITEM ID="321" ORDERED="33" PRICE="101">`
	`  <SERIALNUMBER>654321</SERIALNUMBER>`
	`  <LOCATION>Plovdiv</LOCATION>`
	`  <NAME>Piston</NAME>`
	`  </ITEM>`
	`</INVENTORY2>`

Примерно решение

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
    <xsl:template match="/">
        <INVENTORY2>
        <xsl:for-each select="INVENTORY/ITEM">
            <xsl:choose>
                <xsl:when test="@PRICE &gt; 100">
                    <xsl:copy-of select="." />
                </xsl:when>
                <xsl:otherwise>
                    <ITEM_CONVERTED>
                    <xsl:for-each select="*">
                        <xsl:attribute name="{name()}">
                            <xsl:value-of select="text()"/>
                        </xsl:attribute>
                    </xsl:for-each>
                    <xsl:for-each select="@*">
                        <xsl:element name="{name()}">
                            <xsl:value-of select="."/>
                        </xsl:element>
                    </xsl:for-each>
                    </ITEM_CONVERTED>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:for-each>
        </INVENTORY2>
    </xsl:template>
</xsl:stylesheet>
```

Критерии за оценяване

- Изпълнено е Условие 1: 2 точки;
- Изпълнено е Условие 2: 2 точки;
- Изпълнено е Условие 3.а: 3 точки;
- Изпълнено е Условие 3.б: 3 точки.

Задача 7. За думи α и β над азбука $\Sigma = \{a, b, c\}$, означаваме

$$\alpha \preceq \beta \Leftrightarrow (\exists \gamma \in \Sigma^\star)[\alpha \cdot \gamma = \beta].$$

За произволен език $L \subseteq \Sigma^\star$, означаваме

$$\texttt{Pref}(L) = \{\alpha \in \Sigma^\star \mid (\exists \beta \in L)[\alpha \preceq \beta]\}.$$

Вярно ли е, че за всеки език $L \subseteq \Sigma^\star$:

а) ако L е регулярен, то $\texttt{Pref}(L)$ е регулярен ?

б) ако L не е регулярен, то $\texttt{Pref}(L)$ не е регулярен ?

Обосновете отговорите си като приложите доказателства!

Примерно решение:

Доказателство.

а) Нека L е произволен регулярен език. Ще покажем, че $\texttt{Pref}(L)$ също е регулярен.

Да фиксираме детерминиран краен автомат $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ с език $L(\mathcal{A}) = L$. Нека $Q_{co-acc} = \{q \in Q \mid \exists \gamma \in \Sigma^*(\delta^*(q, \gamma) \in F)\}$ и

$$\mathcal{A}_{pref} = (\Sigma, Q, \delta, q_0, Q_{co-acc}).$$

От теоремата на Клини е достатъчно да докажем, че $L(\mathcal{A}_{pref}) = \texttt{Pref}(L)$.

Първо, нека $\alpha \in \texttt{Pref}(L)$. Това означава, че $\alpha \preceq \beta$, за някоя дума $\beta = \alpha \cdot \gamma \in \mathcal{L}(\mathcal{A})$. Тогава $\delta^\star(q_0, \alpha \cdot \gamma) \in F$, откъдето $p = \delta^\star(q_0, \alpha)$ е добре дефинирано и $\delta^\star(p, \gamma) \in F$. Последното показва, че $p \in Q_{co-acc}$, откъдето $\delta^\star(q_0, \alpha) \in Q_{co-acc}$ и $\alpha \in L(\mathcal{A}_{pref})$.

Второ, нека $\alpha \in \mathcal{L}(\mathcal{A}_{pref})$. Тогава $\delta^\star(q_0, \alpha) = p \in Q_{co-acc}$ и по дефиницията на Q_{co-acc} получаваме, че има дума γ, за която $\delta^\star(p, \gamma) \in F$. Сега е ясно, че $\delta^\star(q_0, \alpha \cdot \gamma) = \delta^\star(p, \gamma) \in F$, тоест $\alpha \cdot \gamma \in L$. Следователно $\alpha \in \texttt{Pref}(L)$.

б) Ще докажем, че условието б) не е изпълнено. За тази цел, ще посочим нерегулярен език L, за който $\texttt{Pref}(L)$ е регулярен.

Да изберем езика $L = \{a^n b^m \mid n \leq m\}$. Ще видим, че L не е регулярен като покажем, че съществуват безкрайно много класове на еквивалентност на релацията на Майхил-Нероуд $\approx_L$, където

$$\alpha \approx_L \beta \Leftrightarrow (\forall \gamma \in \Sigma^\star)[\alpha\gamma \in L \iff \beta\gamma \in L].$$

Достатъчно е да докажем, че за всеки две естествени числа ℓ и k

$$\ell < k \implies a^\ell \not\approx_L a^k.$$

Това лесно се съобразява, защото за $\gamma = b^\ell$, $a^\ell\gamma \in L$, но $a^k\gamma \notin L$.

Сега ще докажем, че $\texttt{Pref}(L) = \{a\}^\star \cdot \{b\}^\star$, който е регулярен език.

Нека $\alpha \in \texttt{Pref}(L)$. Това означава, че $\alpha \preceq a^n b^k$ за някои $k \geq n$. Ясно е, че тогава $\alpha \in \{a\}^\star \cdot \{b\}^\star$.

Обратно, ако $\alpha \in \{a\}^\star \cdot \{b\}^\star$, то $\alpha = a^n b^k$ за някои $n, k \in \mathbb{N}$, откъдето $\alpha \preceq (a^n b^k) \cdot b^n = a^n b^{n+k} \in L$, тоест $\alpha \in \texttt{Pref}(L)$.

□

Критерии за оценяване:

а) 3 точки, от които:

- 1 точка – за конструкцията на $\mathcal{A}_{pref}$;
- 1 точка – за доказателство, че $\texttt{Pref}(L) \subseteq L(\mathcal{A}_{pref})$;
- 1 точка – за доказателство, че $L(\mathcal{A}_{pref}) \subseteq \texttt{Pref}(L)$.

б) 7 точки, от които:

- 1 точка – за избора на езика L. Тази точка се дава, само ако решението заслужава поне още една точка от долните (в частност алтернативната схема);
- 3 точки – за доказателство, че L не е регулярен, от които:
  - 1 точка – за деклариране на безброй много две по две нееквивалентни думи;
  - 2 точки – за доказателство, че те наистина са две по две нееквивалентни.
- 3 точки – за доказателство, че $\texttt{Pref}(L)$ е регулярен, от които:
  - 1 точка – за деклариране, че $\texttt{Pref}(L) = \{a\}^\star \cdot \{b\}^\star$;
  - 1 точка – за доказателство, че $\texttt{Pref}(L) \subseteq \{a\}^\star \cdot \{b\}^\star$;
  - 1 точка – за доказателство, че $\{a\}^\star \cdot \{b\}^\star \subseteq \texttt{Pref}(L)$;

Алтернативна схема за а).: При доказателства на а). използващи рекурсивната дефиниция за регулярен език:

- 1 точка – за доказателство $\texttt{Pref}(L_1 \cup L_2) = \texttt{Pref}(L_1) \cup \texttt{Pref}(L_2)$;
- 1 точка – за доказателство $\texttt{Pref}(L_1 \cdot L_2) = \texttt{Pref}(L_1) \cup L_1 \cdot \texttt{Pref}(L_2)$;
- 1 точка – за доказателство $\texttt{Pref}(L_1^\star) = L_1^\star \texttt{Pref}(L_1)$.
- -1 точка – ако решението е изпълнило горните, но липсва някой от базовите случаи $\texttt{Pref}(\{a\}) = \{\varepsilon, a\}$ и/или $\texttt{Pref}(\emptyset) = \emptyset$. (Доколкото, $\emptyset^\star = \{\varepsilon\}$, тези два случая са достатъчни.)

Алтернативна схема за б).: При доказателства в б)., че L не е регулярен, използващи лема за разрастването:

- 1 точка – за коректен избор на w с проверка, че $w \in L$ и $|w| \geq n_0$, където предварително n_0 е въведено като константата от лемата за разрастването за езика L, например $w = a^{n_0} b^{n_0}$;
- 2 точки – за достигането до противоречие.
- Решения, които не формулират и/или използват по грешен начин лемата не получават точки.

Забележка: Точките от алтернативната и основната схема за оценяване не са адитивни. Ако са приложими повече от една схема, работата се оценява с максимума по съответната схема.

Задача 8. Да се пресметне определеният интеграл

$$\int_0^1 \sqrt{x}\,\ln\sqrt{1+x}\,dx.$$

Примерно решение

В интеграла правим субституцията $x = t^2$, $t \geq 0$ Използваме, че $\ln \sqrt{1+x} = \frac{1}{2} \ln(1+x)$. Така получаваме

$$\int_0^1 \sqrt{x} \ln \sqrt{1+x}\, dx = \int_0^1 t^2 \ln(1+t^2)\, dt.$$

След това внасяме t^2 под знака на диференциала и интегрираме по части:

$$\begin{aligned}\int_0^1 t^2 \ln(1+t^2)\, dt &= \frac{1}{3} \int_0^1 \ln(1+t^2)\, d(t^3) \\ &= \frac{1}{3} \left(t^3 \ln(1+t^2) \Big|_0^1 - \int_0^1 t^3 d \ln(1+t^2) \right) \\ &= \frac{\ln 2}{3} - \frac{2}{3} \int_0^1 \frac{t^4}{1+t^2}\, dt.\end{aligned}$$

Използваме разлагането

$$\frac{t^4}{1+t^2} = \frac{t^4-1}{t^2+1} + \frac{1}{1+t^2} = t^2 - 1 + \frac{1}{1+t^2}.$$

Така получаваме

$$\begin{aligned}\int_0^1 t^2 \ln(1+t^2)\, dt &= \frac{\ln 2}{3} - \frac{2}{3} \int_0^1 t^2\, dt + \frac{2}{3} \int_0^1 dt - \frac{2}{3} \int_0^1 \frac{dt}{1+t^2} \\ &= \frac{\ln 2}{3} - \frac{2}{3} \frac{t^3}{3} \Big|_0^1 + \frac{2}{3} - \frac{2}{3} \operatorname{arctg} t \Big|_0^1 \\ &= \frac{\ln 2}{3} - \frac{2}{9} + \frac{2}{3} - \frac{2}{3} \left(\frac{\pi}{4} - 0 \right) \\ &= \frac{\ln 2}{3} + \frac{4}{9} - \frac{\pi}{6}.\end{aligned}$$

Критерии за оценяване

- субституция $x = t^2$: 2 т.,
- интегриране по части: 3 т.,
- разлагане на $\frac{t^4}{1+t^2}$ във вид удобен за интегриране (в сума на елементарни дроби): 2 т.
- пресмятане на $\int_0^1 (t^2 - 1)\, dt$: 1 т.,
- пресмятане на $\int_0^1 \frac{1}{1+t^2}\, dt$: 1 т.,
- получаване на окончателния отговор: 1 т.

Чернова