

1. Definiáld az alábbi fogalmakat!

fonéma:

betűkészlet:

Zipf-törvény: lista: az első elemek gyakorisága $R_i \sim \frac{1}{i^\alpha}$
Szavak száma növekszik, nem a név-díj, hanem a gyakoriság, növekszik a szavak.

type: típus

token:

XPATH: formalizmus az XML fa névűre hivatkozásra való
főbbi navigáció, XPath-ek kivételével...

morféma: A Szóalkotás és jelentés viszonyát többféleképpen lehet vizsgálni
PL: bagoly, lov, megy
-höz, -nak, -t

relatív toldalék:

unifikáció: egy művelet amellyel egyetemesen nem használható szavakat
ha a tagokba gyorsan jönni kell a kiegészítéssel.

procedurális markup:



2. Válaszolj az alábbi kérdésekre!

30 pont

Mit jelent egy XML dokumentum jóformáltsága és érvényessége? Mi a kettő viszonya?
4/4 Jóformásított XML - megfelel a szabványban meghatározott szintaktikai szabványoknak, de nem kell a típusdefiníciókat megfigyelni.
0/4 Érvényesség - jól formázott, és típusdefiníciókat is tartalmazó.

Mi az unifikációs morfológia lényege?

1/4 Absztrakció művelet - nem igazán a morfológiához tartozik.

Mi a szófaji egyértelműsítés?

0/4 Hasonlítsd össze a SAX és DOM XML feldolgozó módszereket!
2/4 Dom - document object model - teljes doc elérése, technológiai elem beírása elérése (doc egészre kell)

SAX - Stream API - tartalmi a pozíciót, csak aktuális elem elérése (doc részeire kell)

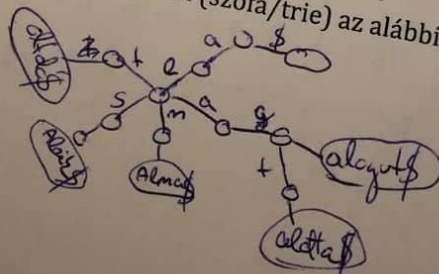
Mi az XML schema és mi jellemzi?

3/3 XSD - Full XML doc adatkészlet típusait, értelmezési tartományát definiálja
- egymással összekötött
- generátor nem kell
- absztrakció művelet elérése definíció adatkészlet

Mi a számítógépes morfológia két fő feladata?

4/4 ajánlatok és ajánlatok kibontásával foglalkozik
1 - Szóalak felismerése
2 - Szóalak generálása.
Ábrázold átmenetgráfos ábrázolással (szófa/trie) az alábbi szótárrészletet!

alszik
alma
alagút
alatta
altató



HUAWEI P10 Plus
LEICA DUAL SUMMILUX

4. Válaszd ki azt az XPath kifejezést, ami illeszkedik az „XSLT Stylesheet”-ben az ____ (1) ____ helyre úgy, hogy azt az „XML Document”-re alkalmazva a „Transformation Results” legyen az eredmény!

“XML Document”

```
<?xml version="1.0" encoding="UTF-8" ?>
<UserList>
  <UserID Password="xxxxxx" authority="administrator">User01</UserID>
  <UserID Password="zzzzzz" authority="general">User02</UserID>
  <UserID Password="xyzxyz" authority="system">User03</UserID>
  <UserID Password="yyyyyy" authority="general">User04</UserID>
</UserList>
```

“Transformation Results”

```
<?xml version="1.0" encoding="UTF-8"?>
<GeneralUserList>
  <UserID>User02</UserID>
  <authority>general</authority>
  <UserID>User04</UserID>
  <authority>general</authority>
</GeneralUserList>
```

“XSLT Stylesheet”

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <GeneralUserList>
      <xsl:apply-templates select="____ ( 1 ) ____" />
    </GeneralUserList>
  </xsl:template>
  <xsl:template match="UserID">
    <UserID><xsl:value-of select="text()" /></UserID>
    <xsl:element name="authority"><xsl:value-of select="@authority" /></xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

- a. ☐ `UserList/UserID[@authority='general']`
b. ☐ `UserList/UserID/@authority='general'`
c. ☐ `UserList/UserID/[@authority='general']`
d. ☒ Egyik sem helyes megadás, mivel az XPath kifejezések hibásak.



5. Ha az „XML Document”-re alkalmazzuk az „XSLT Stylesheet” által definiált transzformációt, akkor melyik lesz annak helyes eredménye?

“XML Document”

```
<?xml version="1.0" encoding="UTF-8" ?>
<UserList>
  <UserID Password="xxxxxx" authority="administrator">User01</UserID>
  <UserID Password="zzzzzz" authority="general">User02</UserID>
  <UserID Password="xyzxyz" authority="system">User03</UserID>
  <UserID Password="yyyyyy" authority="general">User04</UserID>
</UserList>
```

“XSLT Stylesheet”

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <SystemUserList>
      <xsl:apply-templates select="UserList/UserID" />
    </SystemUserList>
  </xsl:template>

  <xsl:template match="UserID">
    <xsl:if test="@authority='system'">
      <UserID><xsl:value-of select="text()" /></UserID>
      <xsl:element name="authority"><xsl:value-of select="@authority" /></xsl:element>
    </xsl:if>
  </xsl:template>
</xsl:stylesheet>
```

- a.

```
<?xml version="1.0" encoding="UTF-8"?>
<UserID>User03</UserID>
<authority>system</authority>
```
- ☒ b.

```
<?xml version="1.0" encoding="UTF-8"?>
<SystemUserList>
  <UserID>User03</UserID>
</SystemUserList>
```
- c.

```
<?xml version="1.0" encoding="UTF-8"?>
<SystemUserList>
  <authority>system</authority>
  <UserID>User03</UserID>
</SystemUserList>
```
- d.

```
<?xml version="1.0" encoding="UTF-8"?>
<SystemUserList>
  <UserID>User03</UserID>
  <authority>system</authority>
</SystemUserList>
```



HUAWEI P10 Plus
LEICA DUAL SUMMILUX

6. Válaszd ki, hogy a megadott DTD definíciók közül melyik felel meg legjobban az alábbi követelményeknek:

A „BookList” elemnek gyereke a „BookInfo”
A „BookInfo”-nak gyereke a „BookName” és az „ISBN”, melyek közül az egyik pontosan egyszer fordul elő.
A „BookInfo”-nak van egy nem kötelező, „BookID” nevű attribútuma.
A „BookInfo” elem legalább egyszer előfordul.

- a.

```
<!ELEMENT BookList ( BookInfo )*>
<!ELEMENT BookInfo ( BookName | ISBN )>
<!ELEMENT BookName (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ATTLIST BookInfo BookID CDATA #IMPLIED>
```
- b.

```
<!ELEMENT BookList ( BookInfo )*>
<!ELEMENT BookInfo ( BookName , ISBN )>
<!ELEMENT BookName (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ATTLIST BookInfo BookID CDATA #IMPLIED>
```
- c.

```
<!ELEMENT BookList ( BookInfo )+>
<!ELEMENT BookInfo ( BookName | ISBN )>
<!ELEMENT BookName (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ATTLIST BookInfo BookID CDATA #IMPLIED>
```
- d.

```
<!ELEMENT BookList ( BookInfo )+>
<!ELEMENT BookInfo ( BookName , ISBN )>
<!ELEMENT BookName (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ATTLIST BookInfo BookID CDATA #IMPLIED>
```

7. Az alábbiak közül melyik felel meg annak a deklarációnak, hogy a „Név” elemnek két gyereke legyen úgy, hogy sorrendben a „Vezetéknév” legyen a „Keresztnév” előtt?

- a.

```
<!ELEMENT Név (Vezetéknév) >
<!ELEMENT Név (Keresztnév) >
```
- b.

```
<!ELEMENT Név (Vezetéknév, Keresztnév) >
```
- c.

```
<!ELEMENT Név (Vezetéknév | Keresztnév) >
```
- d.

```
<!ELEMENT Név (Vezetéknév / Keresztnév) >
```



8. Az alábbiak közül melyik jól formált XML?

- a. ☒ `<Bylaw>`
The final decision-making authority is the `<Department>`General Affairs Department
`</Department>``<Position>`Section Chief`</Position>`
`</Bylaw>`
- b. ☒ `<Article>`Article 2`</Article>`
`<Bylaw>`
The final decision-making authority is the General Affairs Department Section Chief.
`</Bylaw>`
- c. ☐ `<Bylaw>`
`<Final_decision-making_authority Department="General Affairs Department" Position="Section Chief">`
`</Bylaw>`
- d. ☐ `<Bylaw>`
`<Final_decision-making_authority>``<Department>`General Affairs Department`<Position>`Section Chief`</Department>``</Position>``</Final_decision-making_authority>`
`</Bylaw>`

4. Mit jelentenek a következő kifejezések?

10 pont

a) `<!ELEMENT nev (vezeteknev, keresztnév+)>`

2 ~~A~~ A név elemet két gyerek van benne: vezeteknev, keresztnév. kötelezően min 1 db, vagy több lehet.

b) `ul li { color : red }`

2 CSS stílusformázás: az elemnek olyan ul-li → ul-en belül van minél kisebb eltolás, hogy az így jó-e? "background-color: red"?

c) `// * [ starts - with ( @ id , 'B' ) ]`

d) `<xsd:element name="elem" type="xsd:string">`

2 xsd elem - elem neve egy string típusú elnevezés.

e) `<xsd:simpleType name="nyelvTípus">`

`<xsd:restriction base="xsd:string">`
`<xsd:enumeration value="hu"/>`
`<xsd:enumeration value="en"/>`
`<xsd:enumeration value="de"/>`
`</xsd:restriction>`

`</xsd:simpleType>`

Enumerált típus: előre meghatározott értékek (hu, en, de) lehet csak pl: újat hozni nem lehet.



HUAWEI P10 Plus
LEICA DUAL SUMMILUX