

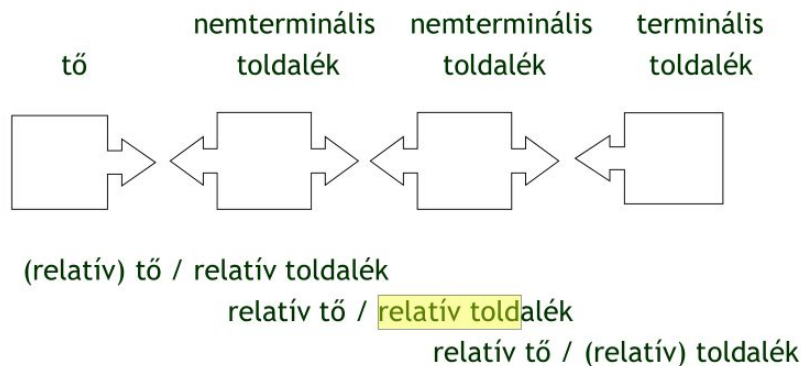
Nyelvtech ZH 2017-2018 kidolgozva

1. Definiáld a következő alapfogalmakat!

- Fonéma:** A beszélt nyelvben a **fonéma** a hangok elemi, elvont egysége, amely egy adott nyelvben önálló egységet alkot, mivel más fonémákkal szemben jelentésmegkülönböztető szerepe van.
- Betűkészlet:** Olyan táblázat, amely adott karakterkészletek megjeleníthető karaktereihez karakterképeket rendel, melyek aztán a számítógép képernyőjén vagy nyomtatásban megjelennek. Az alkalmazott betűkészlet a szöveg számítógépbeli értelmezésére - például a szöveget nyelvi szempontból feldolgozó programokra - nincs befolyással, csak annak megjelenítésekor használatos.
- Zipf-törvény:** Egy szövegben ha az egyes szavakat előfordulási gyakoriságuk szerint sorba rendezzük és ábrázoljuk rang szerint (aszerint, hogy a szó hanyadik az adott sorban), akkor közelítőleg egy hatványfüggvény szerint lecsengő görbét kapunk, -1-hez közeli kitevővel.

$$R_i \sim \frac{1}{i^\alpha}$$

- Type:**
- Token:** Szóközrel elválasztott szavak száma (?)
- XPATh:** Xml path language, formailizmus mellyel az XML fa részeire hivatkozhatunk: fán belüli navigáció, csomópontok kiválasztása
- Morféma:** a nyelv legkisebb jelentéssel bíró alkotórésze, nincs természetes határa
- Relatív toldalék:**



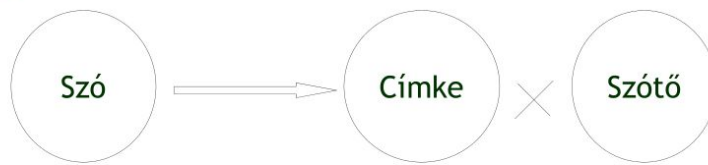
- Unifikáció:** (jegyszerkezet-unifikáció) 2 egyszerkezetben működő művelet, mely csak akkor hiúsul meg, ha van olyan ösvény mindkét szerkezetben, melyeknek eltérő atomi értékei vannak, minden más esetben a létrejövő szerkezet az eredeti két szerkezet összes ösvényét tartalmazza.
- Procedurális markup:** mit kell csinálni a benne lévő infóval. pl: <i>kata</i>
- UCS:** Universal Character Set, minden beszélt holt és egyéb nyelv karakterei.
- XSD:** XML sémanyelv - XSD. felírhatóak segítségével az olyan szabályrendszerek, melynek meg kell, hogy feleljen egy XML dokumentum ahhoz, hogy valid legyen a sémában. (névterek, adattípusok). XML dokumentum formális leírása + forráskód generáláshoz használjuk.
- Véges állapotú transzducer:** mi a fos??
- n-gram: Egymás után követő n darab karakter egy szövegben. Lehet szó, de nem feltétlenül az.

2. Válaszolj az alábbi kérdésekre!

- Mit jelent egy XML dokumentum jólformáltsága és érvényessége? Mi a kettő viszonya?**
Egy jól formázott XML dokumentum megfelel a szabványban rögzített összes szintaktikai szabálynak, de nem köteles megfelelni a dokumentum típus deklarációnak.
Egy érvényes XML dokumentum jól formázott ÉS a dokumentum típus deklarációban megadott struktúra követelményeinek is megfelel.

- b. Mit jelent a közvetlen összetevős elemzés, mi jellemzi?
- c. Mi az unifikációs morfológia lényege?
- d. Mi a szófaji egyértelműsítés?

Egyértelműsítő:



{tokenizált mondat, címkekészlet}
 → szófaji egyértelműsítő →
 {címkézett (és szótővezett) mondat}

- e. Hasonlítsd össze a SAX és DOM XML feldolgozó módszereket!

SAX	DOM
• Sequential acces • számon kell tartani, hogy hol tartunk, csak az aktuális elem érhető el közvetlenül • jó, ha a dokumentum kis része kell egyszerre • "push" típusú feldolgozás • XML fájlt végigolvassa, nem tárolja • memória igénye kicsi • nagy dokumentumokhoz alkalmas • mély struktúrák esetén nehéz • gyors	• random acces • a teljes dokumentum rendelkezésre áll, tetszőleges eleme bármikor elérhető • jó, ha a dokumentum nagy része kell egyszerre • Csomópontokból álló frastruktúrában tárolódik • "pull" típusú feldolgozás • memória igény: $O(\text{csomópontok} * \text{dokumentum méret})$ • bonyolult struktúrák kezelése egyszerű • lassú

- f. Mi az XML schema és mi jellemzi?

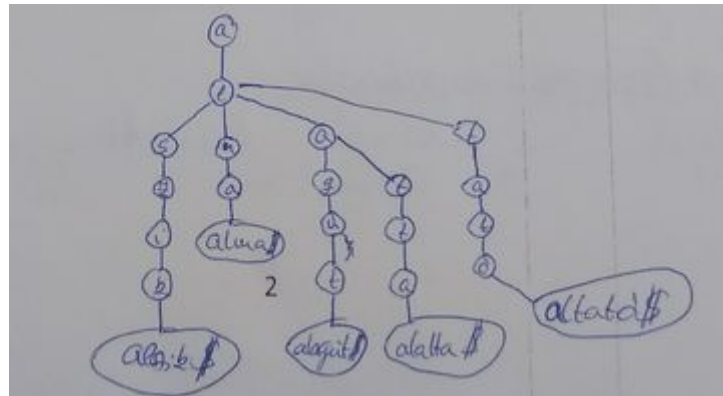
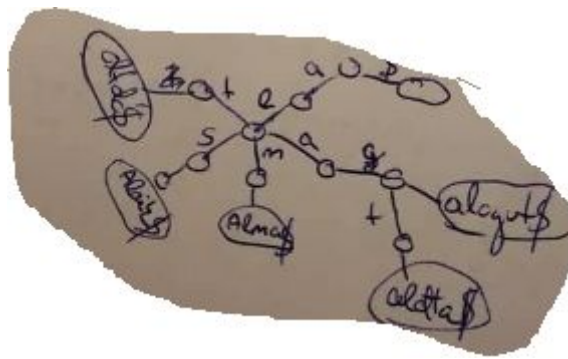
XML sémanyelv - XSD. felírhatóak segítségével az olyan szabályrendszerek, melynek meg kell, hogy feleljen egy XML dokumentum ahhoz, hogy valid legyen a sémában. (névterek, adattípusok)

- g. Mi a számítógépes morfológia két fő feladata?

Szóalak-felismerés: program visszaadja a toldalékolt szóból a szótőve, szőfaját, nyelvtatni információt

Szóalak-generálás: a helyes szóalak előállítása a szótő és morfológiailag releváns nyelvtani információ segítségével.

- h. Ábrázold átmenetgráfos ábrázolással (szófa/trie) az alábbi szószerkezetet! (alszik, alma, alagút, alatta, altató)



- i. Kódold az “öt szép szűz lány örült írók nyúz” szöveget az alábbi kódolások szerint!
(távírat, repülő ékezetes, számkódos)!

távírat: oet szeep szuuez laany ooerueli iiroot nyuuz

repülő ékezetes: o:t sze'p szu''t la'ny o''ru:lt i'ro't nyu'z

számkódos: o2t sze1p szu3z la1ny o3ru2lt i1ro1t nyu1z

- j. Mi a kétszintes morfológia lényege?

Több párhuzamos szabályt egy szabállyá alakít át. Egy két szintes szabály három részből áll: megfeleltetett párok (amiről valójában a szabály szól), környezet (bal/jobb), szabályoperátor. és kétszintes szabályok egyfajta megfeleltetést fejeznek ki, nem újraírást, párhuzamosan alkalmazandók és nem sorosan, az újraírással szemben nem hoznak létre semmilyen köztes reprezentációs szintet.

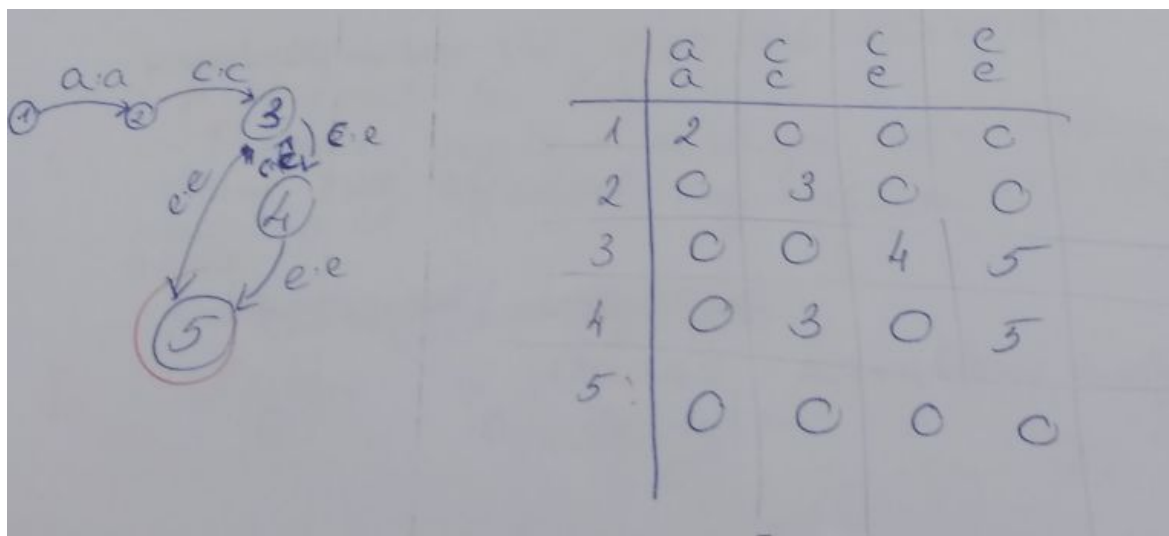
- k. Milyen lépései vannak a morfológiai elemzésnek?

1. Tövesítés (lemmatizálás)
2. Szófaj
3. Morfoszintaktikai jegyek
4. Szóképzés
5. Összetett szó-elemzés

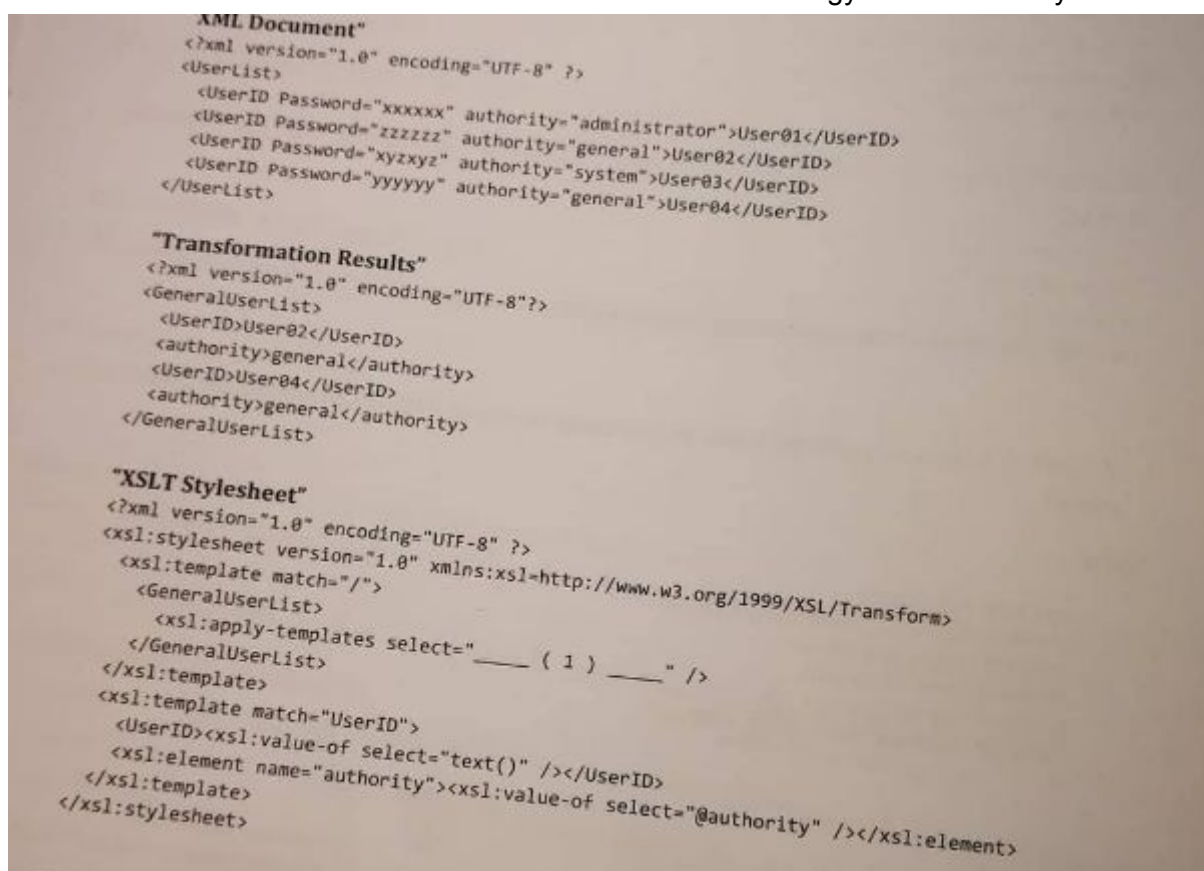
3. Rajzold fel az L1 és L2 nyelvek közötti megfeleltetést végző transzducert grafikus és táblázatos formában!

L1 = ac+e+

L2=L1-ben minden második c legyen e



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!



- 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 Documentre" 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"

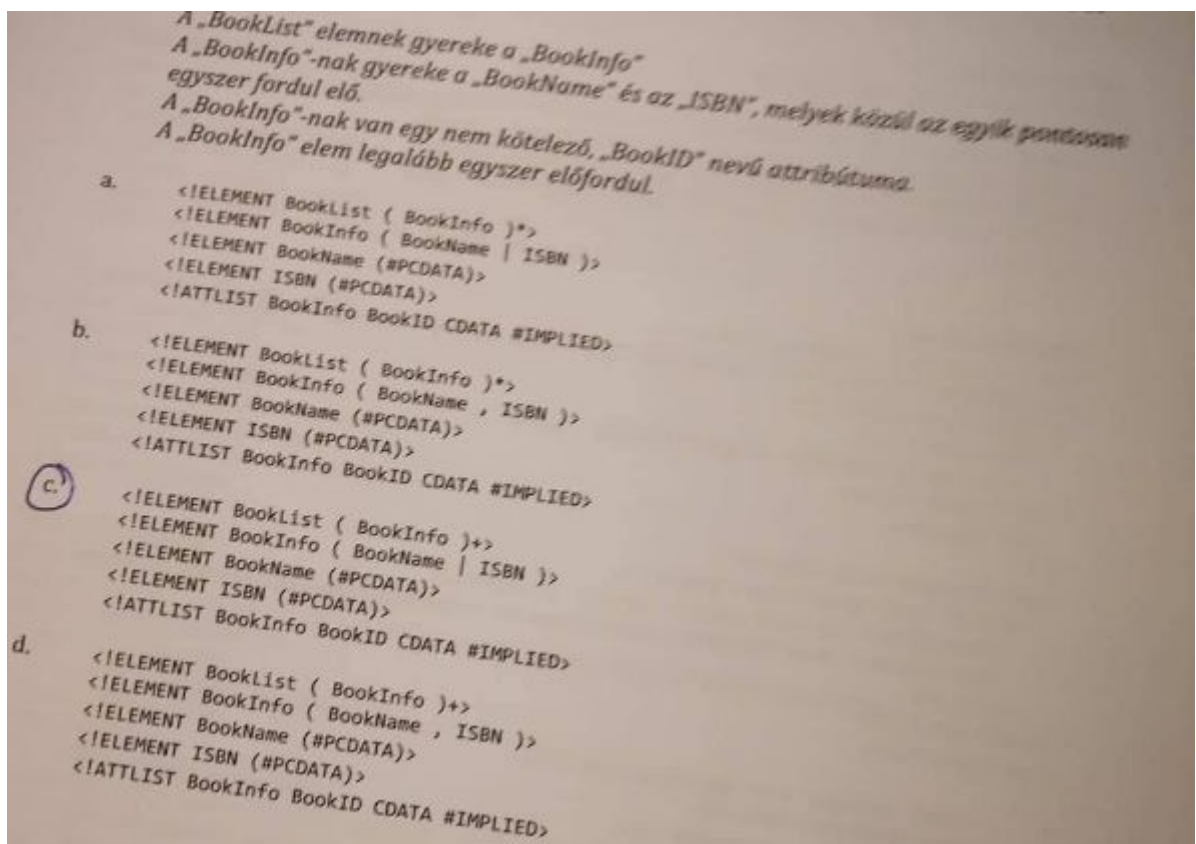
```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <SystemUserList>
      <xsl:apply-templates select="UserList/UserID" />
    </SystemUserList>
  </xsl:template>

  <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>
```

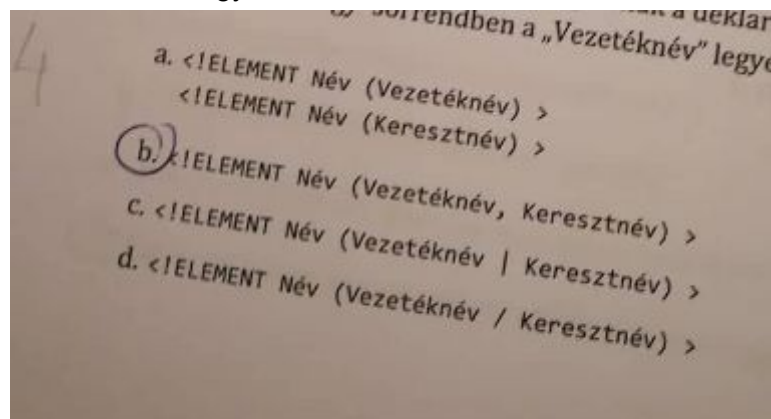
- a. `<?xml version="1.0" encoding="UTF-8"?>`
`<UserID>User03</UserID>`
`<authority>system</authority>`
- ☒ b. `<?xml version="1.0" encoding="UTF-8"?>`
`<SystemUserList>`
`<UserID>User03</UserID>`
`</SystemUserList>`
- c. `<?xml version="1.0" encoding="UTF-8"?>`
`<SystemUserList>`
`<authority>system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

(ez itt azt jelenti, hogy a d) a helyes)

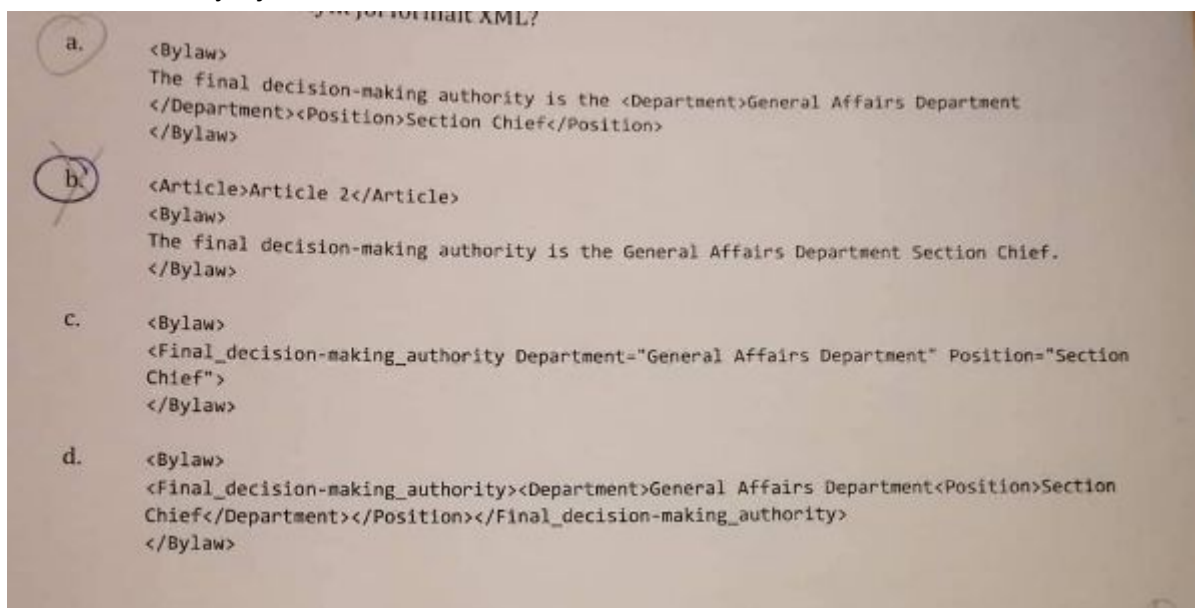
6. Válaszd ki, hogy a megadott DTD definíciók közül melyik felel meg legjobban az alábbi követelményeknek?



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



8. Az alábbiak közül melyik jól formált XML?



(itt az a) a jó válasz)

9. Mit jelentenek a következő kifejezések?

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

A név elemnek van vezetéknév és keresztnév altagja, ebben a sorrendben. Keresztnévből minimum egy van.

b. ul li {color: red}

CSS-es kód, pirosra csinálja a felsorolás színét / ul-nak van egy li gyereke, melynek színe piros (xml-nél)

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

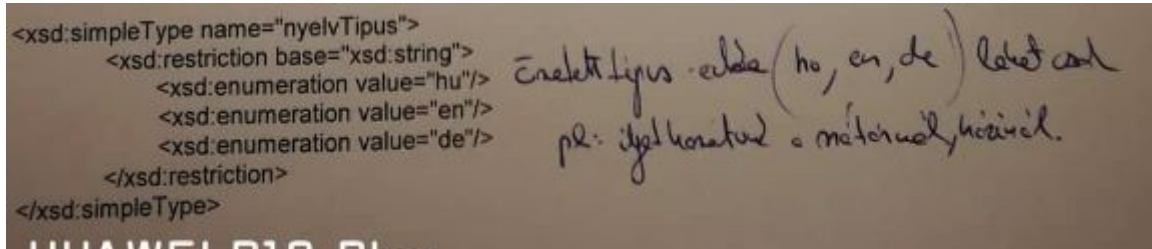
d. /ido[1849>ev]

kihagyja az 1849-nél kisebb évű elemeket

e. <xsd:element name="elem" type="xsd:string">

xsd elem, aminek a neve elem, és string típusú

f.



10. Készítsd el az XML file-t ami megfelel az alábbi DTD-nek és XSD-nek! Rajzold fel az XML fát!

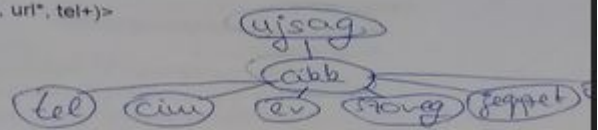
3. XML

Készíts XML file-t, ami megfelel az alábbi DTD-nek és XSD-nek! (3p)

22/ 25 pont

utolsó lap katal

```
<!DOCTYPE ujsag [
  <!ELEMENT ujsag (cikk+)>
  <!ELEMENT cikk (cim+, ev?, szoveg+, jegyzet*, url*, tel+)>
  <!ELEMENT cim (#PCDATA)>
  <!ELEMENT ev (#PCDATA)>
  <!ELEMENT szoveg (#PCDATA)>
  <!ELEMENT jegyzet (#PCDATA)>
  <!ELEMENT tel (#PCDATA)>
  <!ELEMENT url (#PCDATA)>
  <!ATTLIST cikk szerzo CDATA #REQUIRED>
  <!ATTLIST url href CDATA #REQUIRED>
]
```



```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="cim" type="xsd:string"/>
  <xsd:element name="ev" type="xsd:positiveInteger"/>
  <xsd:element name="szoveg" type="xsd:string"/>
  <xsd:element name="jegyzet" type="xsd:string"/>
  <xsd:element name="tel" type="xsd:positiveInteger"/>
  <xsd:attribute name="szerzo" type="xsd:string"/>
  <xsd:attribute name="href" type="xsd:string"/>

  <xsd:element name="cikk">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="cim"/>
        <xsd:element ref="ev"/>
        <xsd:element ref="szoveg"/>
        <xsd:element ref="jegyzet"/>
        <xsd:element ref="url"/>
        <xsd:element ref="tel" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute ref="szerzo" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="url">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:string">
          <xsd:attribute ref="href"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="ujsag">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="cikk"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

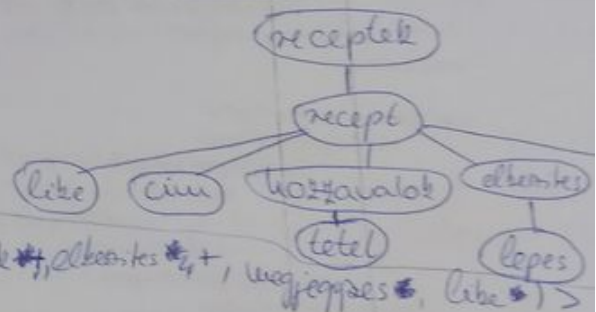
3, <?xml version = "1.0" ~~2.0~~ encoding = "UTF-8" >
 <ujisag>
 <abb> ~~21. encl usagob~~ ~~/abb~~ → <cih ranzo = "0x11">
 <~~cih ranzo = "0x11"~~>
 <aim> 21. encl menyek </aim>
 <ev> 2018 </ev>
 Valami.
 <jegget> </jegget>
 <url> </url>
 <tel> 0611234567 </tel>
 </abb>
</ujisag>

11. Készíts DTD-t az alábbi XML file-hoz! Rajzold fel az XML fát!

Készíts DTD-t az alábbi XML file-hoz! (3p)
Rajzold fel az XML fát! (2p)

```
<?xml version="1.0" encoding="UTF-8"?>
<receptek>
  <recept id="1">
    <cim>Csülök pékné módra</cim>
    <hozzavalok>
      <tetel mennyiség="2" meritekegyseg="db" szukseges="true">csülök</tetel>
      <tetel mennyiség="1.5" meritekegyseg="kg" szukseges="true">krumpli</tetel>
      <tetel mennyiség="4" meritekegyseg="fej" szukseges="true">hagyma</tetel>
      <tetel mennyiség="1" meritekegyseg="evokanal" szukseges="true">só</tetel>
      <tetel mennyiség="1" szukseges="true">fokhagyma</tetel>
      <tetel mennyiség="1" meritekegyseg="evokanal" szukseges="true">örölt bors</tetel>
      <tetel mennyiség="1" meritekegyseg="evokanal" szukseges="true">majoranna</tetel>
      <tetel mennyiség="500" meritekegyseg="ml" szukseges="false">sőr</tetel>
    </hozzavalok>
    <elkeszites>
      <lepes>A krumplit meghámozzuk, megmossuk, hasábokra vágjuk, és egy tepsibe tesszük.</lepes>
      <lepes>A hagymát először félbe vágjuk, majd felkarikázzuk, a felét rászórjuk a krumplira, és megsózzuk.</lepes>
      <lepes>A csülköt besózzuk, bedörzsöljük a borssal és az összezúzott fokhagymával, majd a krumplira fektetjük.</lepes>
      <lepes>Az egészet megszórjuk majorannával, majd a maradék hagymakarikát a csülökre szórjuk.</lepes>
      <lepes>Aláöntünk egy nagy pohár vizet, majd letakarjuk alufóliával, és előmelegített sütőben közepes lángon egy órát sütjük.</lepes>
      <lepes>Ezután levesszük a fóliát, és még további fél órát sütjük, hogy megpiruljon az étel.</lepes>
    </elkeszites>
    <megjegyzes>Jó munkát! :)</megjegyzes>
    <like>19</like>
  </recept>
</receptek>
```

XML fa:



DTD:

```
<!DOCTYPE receptek [
  <!ELEMENT recept (cim+, hozzavalok*, elkeszites*, megjegyzes*, like*)>
  <!ELEMENT cim (#PCDATA)>
  <!ELEMENT hozzavalok (tetel)>
  <!ELEMENT elkeszites (#PCDATA)>
  <!ELEMENT megjegyzes (#PCDATA)>
  <!ELEMENT like (#PCDATA)>
  <!ELEMENT tetel (#PCDATA)>
  <!-- ATTENTION: The following are not valid DTD syntax -->
  <!-- <!ATTLIST recept id CDATA #REQUIRED -->
  <!-- <!ATTLIST tetel mennyiseg CDATA #REQUIRED -->
  <!-- <!ATTLIST tetel meritekegyseg CDATA #IMPLIED -->
  <!-- <!ATTLIST tetel szukseges CDATA #REQUIRED -->
--]
```

12. Egészítsd ki az alábbi XSD-t a fenti XML alapján!

A nyelvtechnológia alapjai - 1. ZH
2018. november 6.

Név: Szabó Katalin Dóra

A

Egészítsd ki az alábbi XSD-t a fenti XML alapján! (10p)

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="cim" type="xsd:string"/>
  <xsd:element name="megjegyzes" type="xsd:string"/>
  <xsd:element name="like" type="xsd:int"/>
  <xsd:attribute name="id" type="xsd:int"/>
  <xsd:attribute name="ertekezes" type="xsd:boolean"/>
  <xsd:attribute name="lepes" type="xsd:string"/>
  <xsd:element name="tetei">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="megjegyzes" use="required"/>
        <xsd:element ref="ertekezes" use="optional"/>
        <xsd:element ref="lepes" use="required"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="hozzavalok">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="elkeszites"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="elkeszites">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="lepes"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="recept">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="cim"/>
        <xsd:element ref="hozzavalok"/>
        <xsd:element ref="elkeszites"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="receptek">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="recept"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```