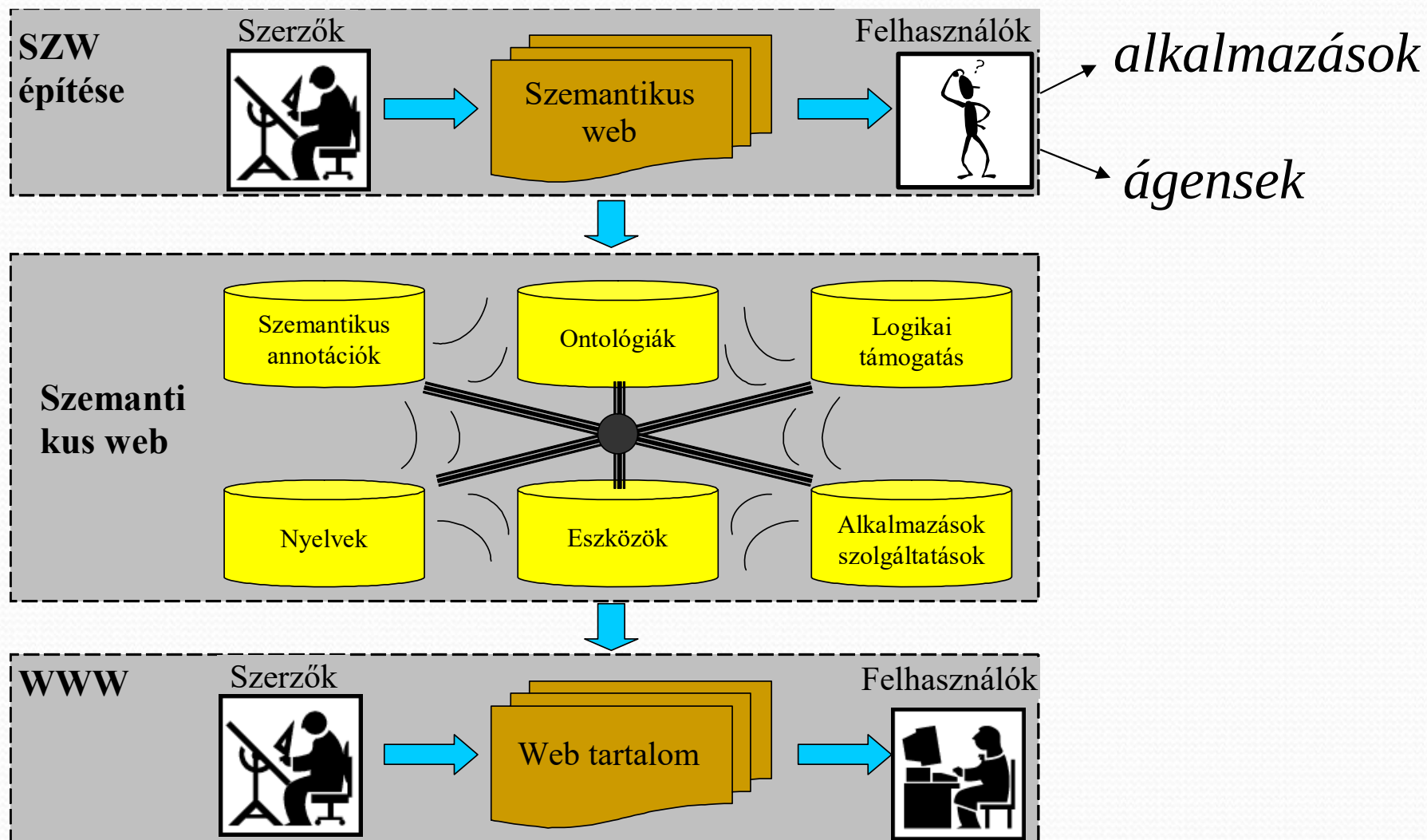


Integrációs és ellenőrzési technikák
VIMIAC04, 2019. tavasz

ONTOLÓGIÁK, OWL₂, DL 4. Előadás

Méréstechnika és Információs Rendszerek Tanszék
<https://www.mit.bme.hu/oktatas/targyak/vimiaco4>

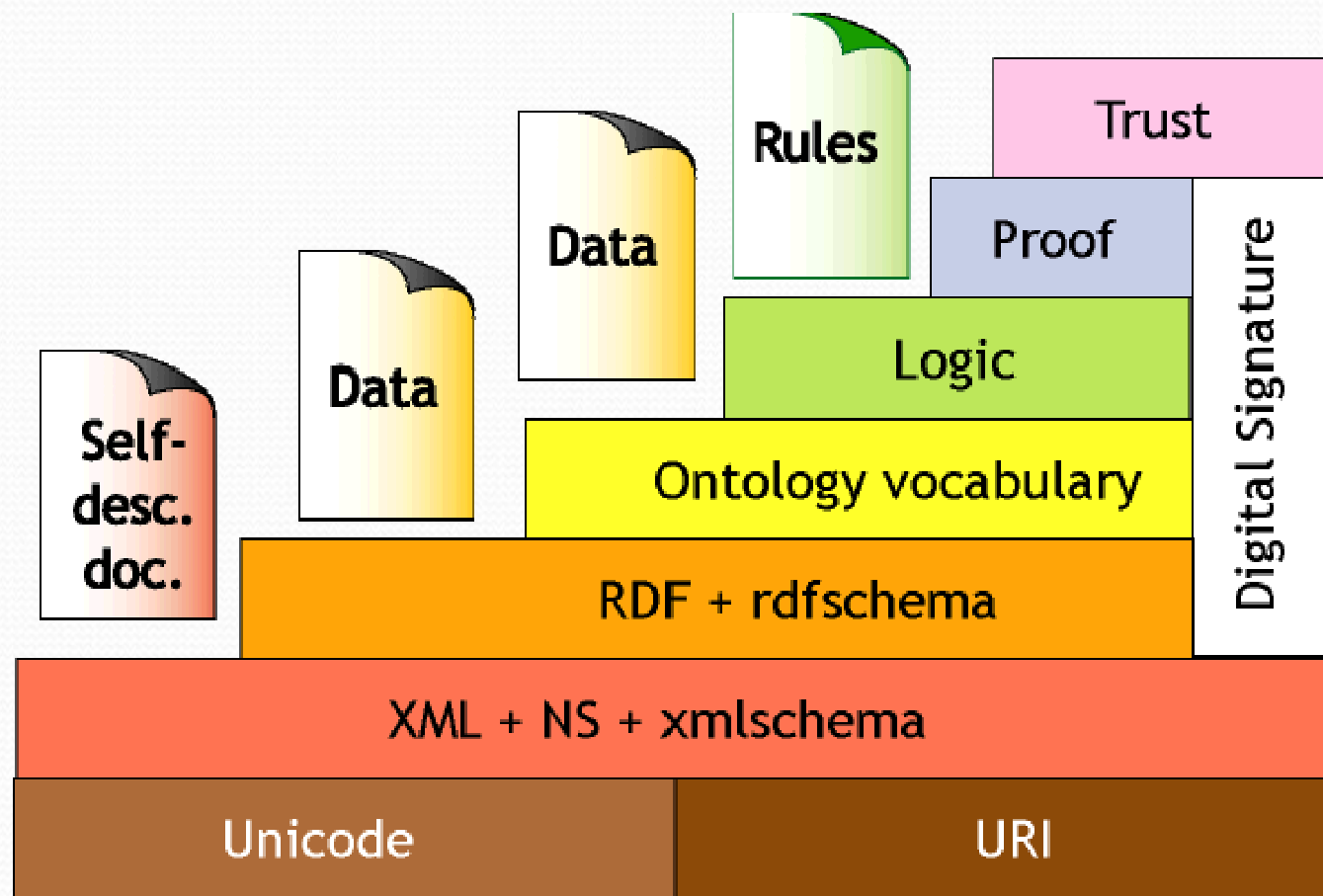
Szemantikus web – új felhasználók



A szemantikus web koncepció

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

-- Tim Berners-Lee





Ontológiák, az OWL nyelv

Szemantikus web példa

- Legyen ismert ez az állítás:
 - Budapest Magyarország fővárosa
- Háttértudásunk segítségével levezethetjük:
 - Budapest Magyarország fővárosa ✓
 - Magyarország egy állam ✓
 - Budapest egy város ✓
 - Budapest Magyarországon van ✓
 - Debrecen nem Magyarország fővárosa ✗
 - Budapest nem Ausztria fővárosa ✗
 - Budapest nem egy állam ✗
 - ...

Szemantikus web példa

- RDFS szintig nem tudjuk leírni a következőket:
 - Minden államnak pontosan egy fővárosa van
 - Tulajdonság számosság
 - Minden város csak egy ország fővárosa lehet
 - Függvény tulajdonság
 - Egy város nem lehet egyben egy állam is.
 - Diszjunkt osztályok
 - ...

OWL logika igény példa

- Állatok két csoportja: **Male** (hím) és **Female** (nőstény).

```
<rdfs:Class rdf:ID="Male">  
  <rdfs:subClassOf rdf:resource="#Animal"/>  
</rdfs:Class>
```

- A **subClassOf** kifejezés felhasználásával definiálhatjuk, hogy a **#Male** osztály alosztálya az **#Animal** osztálynak

```
<rdfs:Class rdf:ID="Female">  
  <rdfs:subClassOf rdf:resource="#Animal"/>  
  <owl:disjointWith rdf:resource="#Male"/>  
</rdfs:Class>
```

- Nőstény állatok és alosztályát alkotják az állatok osztálynak, de tudjuk hogy nincs közös elem a hím osztállyal (**disjointWith** elem használata).

OWL elemek, példák

- Osztályok
 - Person osztály
 - Man, Woman alosztály
- Tulajdonságok (egyedi)
 - isWifeOf, isHusbandOf
- Tulajdonság jellemzők, korlátok
 - inverseOf
 - domain
 - range
 - cardinality
- Osztályok közti relációk
 - disjointWith

Ontológiák

- RDFS hasznos, de nem ad megoldást a szemantika pontos leírására.
- Összetett alkalmazások további igényei:
 - Tulajdonságok leírása, jellemzése
 - Különböző URI-val rendelkező objektumok azonosságának leírása (ekvivalencia)
 - Osztályok diszjunkt vagy éppen ekvivalens jellege
 - Osztályok konstruálása (nemcsak megnevezése)
 - Következtetési igények támogatása:
 - Pl.: “Ha két «Person» erőforrás «A» és «B» azonos «**foaf:email**» tulajdonsággal rendelkeznek, akkor «A» és «B» identikus.

Ontológiák

- Az SZW világban az ontológiákat a következő értelemben használjuk:

Fogalmak és ezek relációinak definiálása egy adott tudásterület leírása céljából.

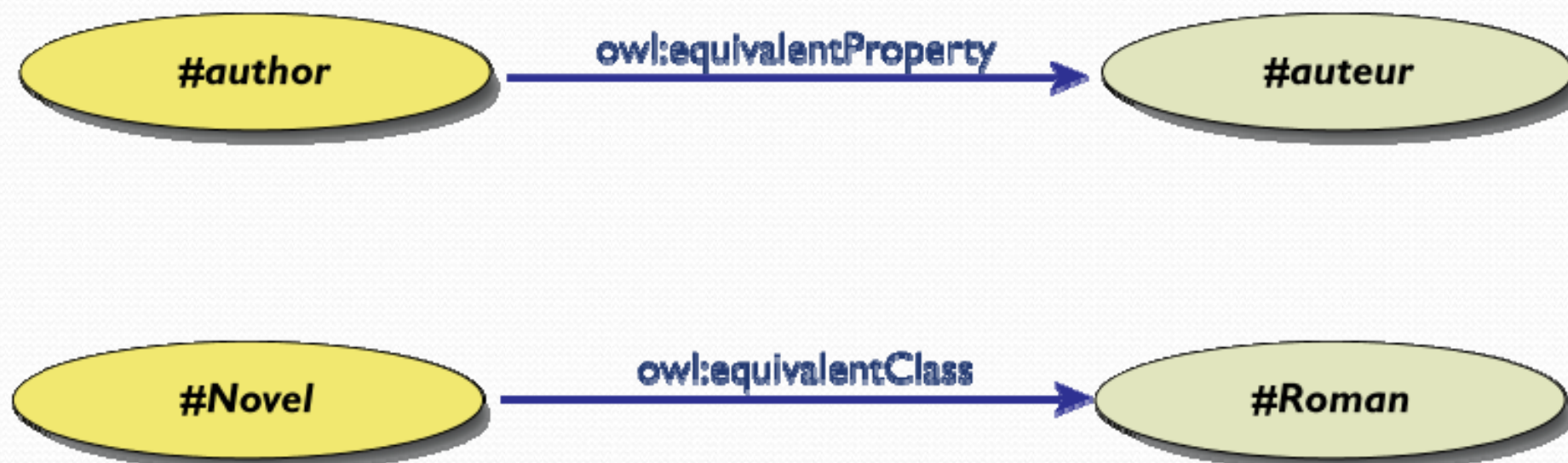
- Az RDFS is tekinthető egy egyszerű ontológia nyelvnek
- Nyelvek definiálása mindig egyfajta kompromisszum
 - gazdag szemantika tudás gazdag alkalmazásokhoz
 - ésszerűség (fizibilitás, megvalósíthatóság, fordító, következtetőgép)

Web Ontology Language = OWL

- OWL az SZW struktúrába egy újabb réteg, az RDFS bővítése
 - Saját névterek, saját elemek, kifejezések
 - RDFS-re épül (tartalmazza)
- Önálló SZW ajánlás
 - “OWL 2” – 2010 óta gyakorlatilag csak ezt használjuk
 - Akit a részletek érdekelnek:
<https://www.w3.org/TR/owl2-overview/>

Ekvivalencia relációk

- Osztályokra:
 - **owl:equivalentClass**: két osztálynak azonosak az elemei
 - **owl:disjointWith**: nincs közös elemük
- Tulajdonságokra:
 - **owl:equivalentProperty**
 - Példa: **a:author** vs. **f:auteur**
 - **owl:propertyDisjointWith**
- Egyedekre:
 - **owl:sameAs**: két URI ugyanazt a fogalmat vagy egyedet reprezentálja
 - **owl:differentFrom**: negált kifejezése az **owl:sameAs** kifejezésnek



Használat: owl:sameAs

- Utazási példából, Amsterdam leírása két forrásban:(Dbpedia és Geonames):

```
<http://dbpedia.org/resource/Amsterdam>  
owl:sameAs <http://sws.geonames.org/2759793>;
```

Példa logikai kapcsolatra

Ha egy tulajdonság *inverse functional*-nak lett definiálva, akkor, akkor a tulajdonság tárgya egyértelműen meghatározza a tulajdonság alanyát.

- Ha a következő igaz két állításra:

```
:email rdf:type owl:InverseFunctionalProperty.  
<A> :email "mailto:a@b.c".  
<B> :email "mailto:a@b.c".
```

Akkor ebből levezethető:

```
<A> owl:sameAs <B>.
```

- Így új relációkhoz jutunk (RDFS-ben erre nem volt lehetőség)

Osztályok az OWL-ben

- RDFS: létező osztályokat alosztály struktúrába rendezhettük – osztályhierarchia építése, semmi több...
- OWL osztályok konstruálhatók más osztályok vagy példányok alapján, :
 - Elemek felsorolásával
 - Osztályok relációinak alkalmazásával: intersection, union, complement, stb.

Osztályok OWL

- Osztályok, egyedek (classes, individuals)
 - Önálló **owl:Class** osztály van definiálva (az **rdfs:Class** specializációjaként)
 - Egyedek (individuals) definiálása külön osztályban történik: **owl:Thing**
- Pl. egy pontos definícióra:

```
ex:Person rdf:type owl:Class.
```

```
<uri-for-Amitav-Ghosh>  
  rdf:type owl:Thing;  
  rdf:type owl:Person .
```

```
:£ rdf:type owl:Thing.  
:€ rdf:type owl:Thing.  
:$ rdf:type owl:Thing.  
:Currency  
  rdf:type owl:Class;  
  owl:oneOf (:€ :£ :$).
```


Unió

```
:Novel          rdf:type owl:Class.  
:Short_Story    rdf:type owl:Class.  
:Poetry         rdf:type owl:Class.  
:Literature rdf:type owl:Class;  
    owl:unionOf (:Novel :Short_Story :Poetry).
```

- További lehetőségek: **complementOf**, **intersectionOf**, ...

Példa...

Ha:

```
:Novel          rdf:type owl:Class.  
:Short_Story    rdf:type owl:Class.  
:Poetry         rdf:type owl:Class.  
:Literature     rdf:type owl:Class;  
                owl:unionOf (:Novel :Short_Story :Poetry).  
  
<myWork> rdf:type :Novel .
```

akkor

```
<myWork> rdf:type :Literature .
```


Eddig...

- OWL: erős leíró elemeket definiáltunk
- pl., adatbázisok összeköthetőek **owl:sameAs**, vagy inverse functional tulajdonságokkal.
- Számos kapcsolatot felderíthetünk hagyományos következtetési eljárásokkal

Korlátozások formálisan is...

- Létezik **owl:Restriction** osztály
 - Hivatkozva a korlátozandó tulajdonságokat
 - Megadva a korlátozás tartalmát
- Példa: subClass

```
:Listed_Price rdfs:subClassOf [  
    rdf:type                owl:Restriction;  
    owl:onProperty         p:currency;  
    owl:allValuesFrom :Currency.  
].
```


Lehetséges felhasználás...

Ha:

```
:Listed_Price rdfs:subClassOf [  
    rdf:type                owl:Restriction;  
    owl:onProperty         p:currency;  
    owl:allValuesFrom :Currency.  
].  
  
:price rdf:type :Listed_Price .  
:price p:currency <something> .
```

Akkor a következő állítás igaz:

```
<something> rdf:type :Currency .
```

További korlátozások

- **allValuesFrom** helyettesíthető:
 - **someValuesFrom**
 - Példa: valamelyik devizában fejezzük ki az árfolyamot (legalább egyben)
 - **hasValue**, legalább egy értéke kell, hogy létezzen
- Számosság korlátozások (értékvizsgálat helyett)
 - Pl. legalább egyszer elő kell, hogy forduljon egy elem egy kategóriában

OWL 2

- Korlátozások definiáltak
 - classes, individuals, object , datatype properties - megkötésekkel
 - object properties csak individuals -re
 - *datatype* property nincs tovább specifikálva
 - ...
- Hatékony következtető algoritmus létezik!

OWL 2

- Korlátozások megadásával nagy méretű ontológiák építhetők,
és alkalmazhatóak pl. orvosi, robotika, biológia tárgyterületeken
- OWL 2 lett a formális ontológiák nyelve
 - Nem feltétlenül a weben használjuk

OWL2: Könnyítések (szintaktika)

- Diszjunkt osztályok

- OWL 1:

```
:Wine owl:equivalentClass [  
  a owl:Class ;  
  owl:unionOf (:RedWine :RoséWine :WhiteWine) ] .  
  
:RedWine owl:disjointWith :RoséWine, :WhiteWine .  
:RoséWine owl:disjointWith :WhiteWine .
```

- OWL 2:

```
:Wine owl:disjointUnionOf (:RedWine  
  :RoséWine :WhiteWine ).
```

- Megengedett formalizmus:

```
_:x a owl:AllDisjointClasses ;  
  owl:members (:RedWine :RoséWine WhiteWine ).
```

OWL2: Könnyítések (szintaktika)

- Állítások, tulajdonságok negáltja kifejezhető
- pl.: Paul nem Peter apja
- ```
_x [a owl:NegativeObjectPropertyAssertion;
 owl:sourceIndividual :Paul ;
 owl:targetIndividual :Peter ;
 owl:assertionProperty :fatherOf] .
```

VAGY

Paul a

```
[owl:complementOf
 [a owl:Restriction ;
 owl:onProperty :fatherOf ;
 owl:hasValue :Peter]
] .
```



# OWL2: Reflexiv osztály korlátozások

hasSelf reláció használata

Példa: autodidakta tanulás definíciója

```
:AutoDidact owl:equivalentClass [
 a owl:Restriction ;
 owl:onProperty :teaches ;
 owl:hasSelf "true"^^xsd:boolean] .
```

# OWL alapú ontológia

- "Debrecen nem Magyarország fővárosa ✖
- Miért nem?
  - Országoknak pontosan egy fővárosa van,
  - Debrecen és Budapest nem ugyanaz a város

- OWL:

```
:capitalOf a owl:InverseFunctionalProperty .
:Budapest :capitalOf :Magyarország .
:Budapest owl:differentFrom :Debrecen .
```

Lekérdezés { :Debrecen :capitalOf :Magyarország . } → false



# OWL alapú ontológia

- Budapest nem Ausztria fővárosa ✖
- Miért nem?
  - Egy város csak egy ország fővárosa lehet
  - Ausztria és Magyarország nem azonos

- Hasonlóan:

```
:capitalOf a owl:FunctionalProperty .
:Budapest :capitalOf :Magyarország .
:Ausztria owl:differentFrom :Magyarország .
```

Lekérdezés { :Budapest :capitalOf :Ausztria . } → false

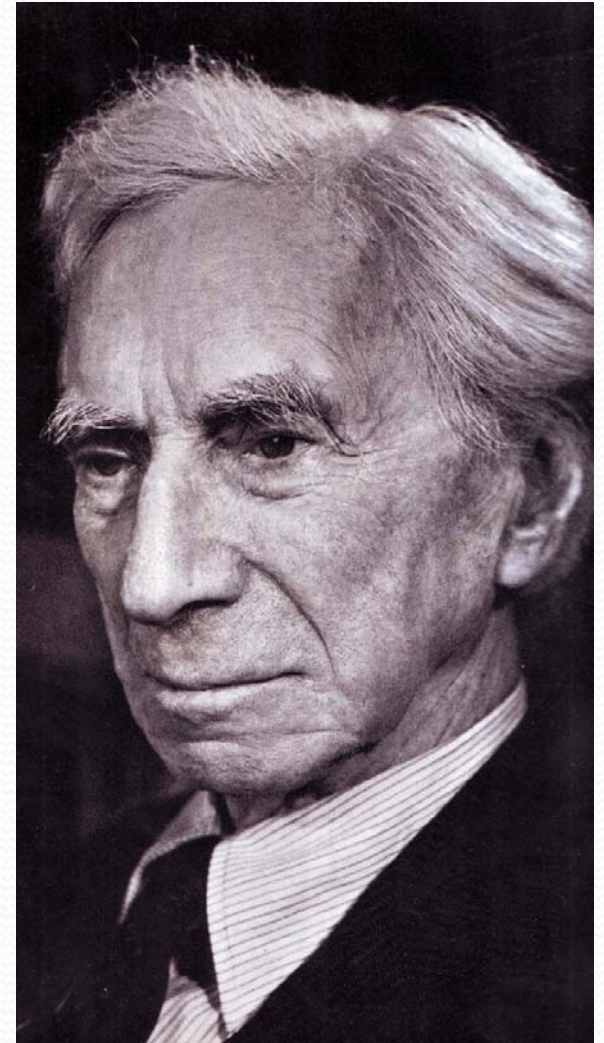
# OWL2 példa: Russell paradoxon

Paradoxon

Bertrand Russell, 1918

Egy városban, ahol pontosan egy borbély van, mindenkit megborotvál, aki nem borotválja meg önmagát.

Ki borotválja meg a borbélyt?





# OWL2 példa: Russell paradoxon

## Osztály definíciók

```
:People owl:disjointUnionOf
 (:PeopleWhoShaveThemselves
 :PeopleWhoDoNotShaveThemselves) .
```

## Relációk:

```
:shavedBy rdfs:domain :People .
:shavedBy rdfs:range :People .
:shaves owl:inverseOf :shavedBy .
```

## Szabály:

```
:People rdfs:subClassOf [
 a owl:Restriction ;
 owl:onProperty :shavedBy ;
 owl:cardinality "1"^^xsd:integer] .
```

# OWL2 példa: Russell paradoxon

- Borbély definíciója:

- - `:Barbers rdfs:subClassOf :People ;`
  - `owl:equivalentClass [`
    - `rdf:type owl:Class ;`
    - `owl:oneOf ( :theBarber )`
  - `] .`



# OWL2 példa: Russell paradoxon

Magukat borotválók osztálya:

```
:PeopleWhoShaveThemselves owl:equivalentClass [
 rdf:type owl:Class ;
 owl:intersectionOf
 (:People
 [
 a owl:Restriction ;
 owl:onProperty :shavedBy ;
 owl:hasSelf "true"^^xsd:boolean
]
)
] .
```

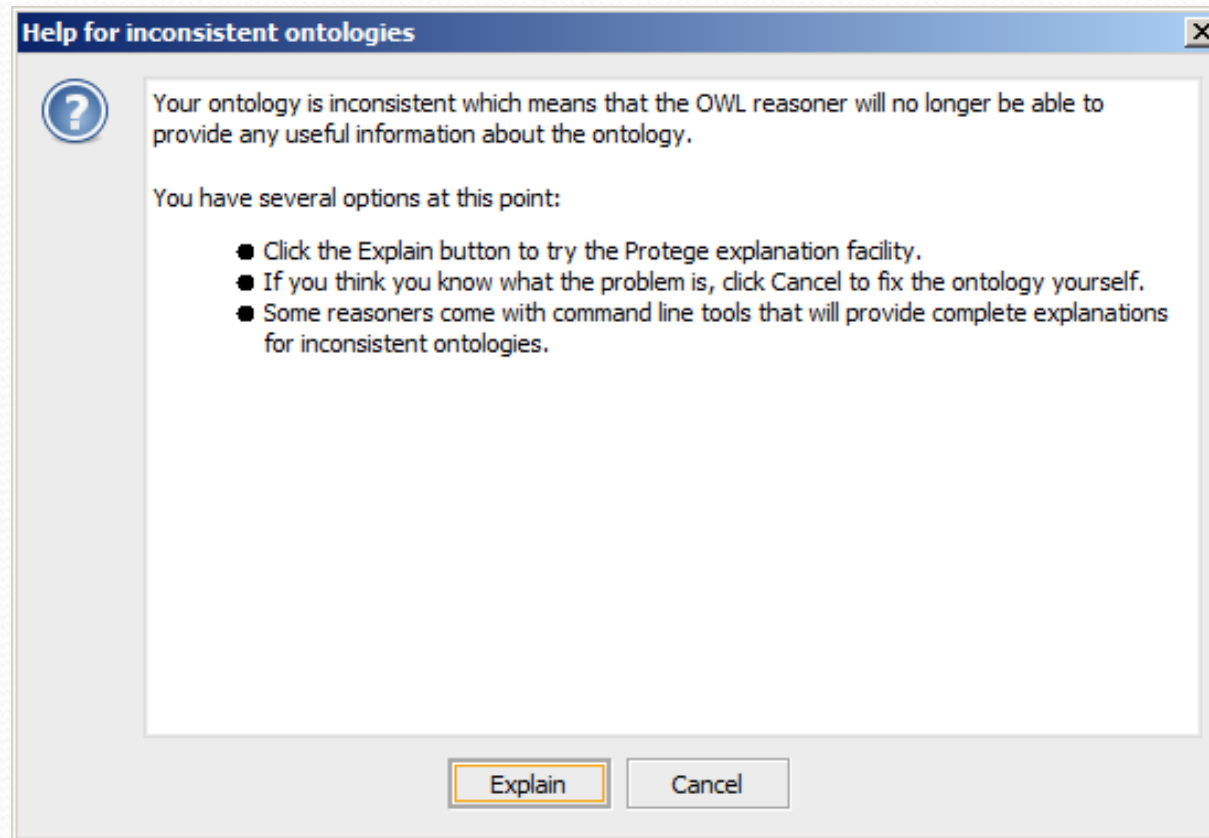
# OWL2 példa: Russell paradoxon

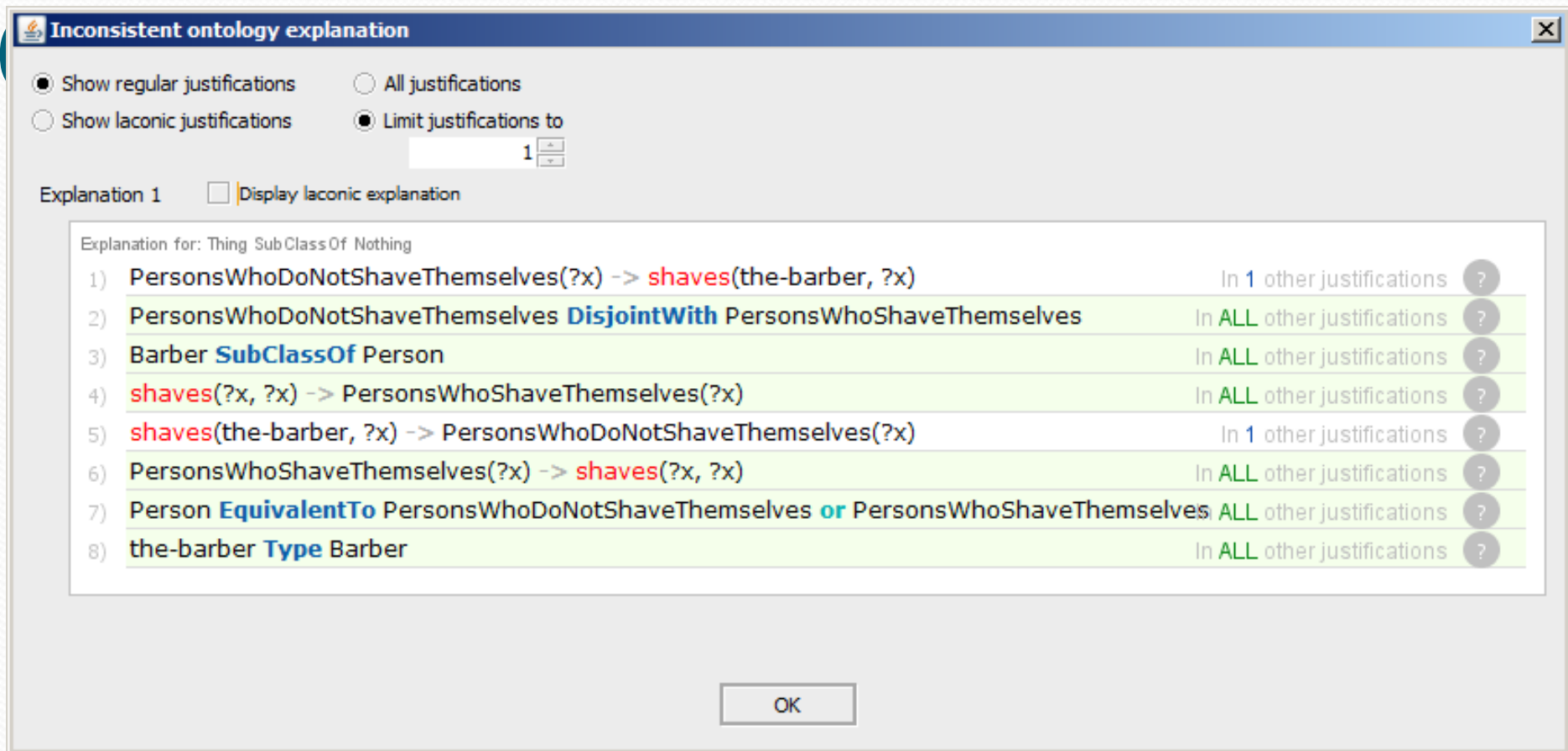
Akik nem magukat borotválják:

```
:PeopleWhoDoNotShaveThemselves owl:equivalentClass [
 a owl:Class ;
 owl:intersectionOf (
 :People
 [a owl:Restriction
 owl:onProperty :shavedBy ;
 owl:allValuesFrom :Barbers
]
)
] .
```



# OWL2 példa: Russell paradoxon







# Következtetés OWL DL nyelvben

- RDFS következtetés
  - Előrefele láncolás
  - Új axiómák levezetése ismertek alapján
- OWL DL következtetés bonyolultabb:
  - Előrefele láncolás nem hatékony nagy adatbázisokon
  - Konjunkció (pl., unionOf) nem támogatott
  - Más megközelítés: Tableau következtetés
  - Alapgondolat: találjunk ellentmondást az ontológiában
    - Egy állítás és negáltja is szerepel

# Következtetés OWL DL nyelvben

- Mit várunk a következtetőgéptől?
  - Alosztály definiálás
    - Pl. Minden madár repülő állat?
  - Ekvivalens osztályok
  - Diszjunkt osztályok
    - Pl. Lehet-e egy állat egyszerre emlős és madár is?
  - Osztály konzisztencia
    - Pl. Emlősök szaporodhatnak-e tojással?
  - Példányok kategorizálása
  - Osztályok tagjainak felsorolás



# Példa: Egy egyszerű ellentmondás

- Adott:

```
:Man a owl:Class .
:Woman a owl:Class .
:Man owl:disjointWith :Woman .
```

```
:Alex a :Man .
:Alex a :Woman .
```

# Példa: Egy egyszerű ellentmondás

- Levezethető:

- $:Man \cap :Woman = \emptyset$

- ```
owl:Nothing owl:intersectionOf (:Man :Woman) .
```

- $:Alex \in (:Man \cap :Woman)$

- ```
:Alex a [a owl:Class; owl:intersectionOf
 (:Man :Woman)] .
```

- Pl.:

- $:Alex \in \emptyset$

- ```
:Alex a owl:Nothing .
```

- ***Tehát a példány nem létezhetne, de van!***

Érvelési feladatok

- Alosztály relációk

Például: $\text{Student} \subseteq \text{Person} \Leftrightarrow$ „Every student is a person“

- Bizonyítási módszer: Reductio ad absurdum
 - "Hozzunk létre" egy i példányt
 - Definiáljuk: $\text{Student}(i)$ és $\neg \text{Person}(i)$
 - Ellenőrizzük az ellentmondás-mentességet
 - Ha létezik ilyen: $\text{Student} \subseteq \text{Person}$ igaz
 - Ha nem létezik: $\text{Student} \subseteq \text{Person}$ nem levezethető
 - (Ettől még lehet igaz)

Példa: alosztály relációk

- **Ontológia:**

```
:Student owl:subClassOf :UniversityMember .  
:UniversityMember owl:subClassOf :Person .
```

- **Bevezetett példányok:**

```
:i a :Student .  
:i a [ owl:complementOf :Person ] .
```

- **Így adott most:**

```
:i a :Student .  
:Student owl:subClassOf :UniversityMember .
```

Ezáltal:

```
:i a :UniversityMember .
```

- **.. és ez is igaz:**

```
:UniversityMember owl:subClassOf :Person .
```

- **Ebből pedig levezethető:**

```
:i a Person .
```


Példa: alosztály relációk

- Tehát ismerjük:

```
:i a :Person .
```

```
:i a [ owl:complementOf :Person ] .
```

Pl.,

```
:i a [ owl:intersectionOf ( :Person  
                               [ owl:complementOf :Person  
                               ] ) ] .
```

- Amiből következik, hogy:

```
:i a owl:Nothing .
```

Érvelési feladatok

- Osztály ekvivalencia
 - $\text{Person} \equiv \text{Human}$
- Részfeladatra bontható:
 - $\text{Person} \subseteq \text{Human}$ and
 - $\text{Human} \subseteq \text{Person}$

Tehát két alosztály definíció igazolása a feladat

- Osztályok diszjunktsága
 - C és D diszjunkt osztályok?
 - Definiáljunk példányokat: $C(i)$ és $D(i)$
 - - Ellentmondás vizsgálat: létezik-e ilyen i

Osztály konzisztencia vizsgálata

- Léteznek-e az osztályhoz tartozó példányok?

- Pl., nős aggregények:

- ```
:Bachelor owl:subClassOf :Man .
```

- ```
:Bachelor owl:subClassOf
```

- ```
 [a owl:Restriction;
```

- ```
    owl:onProperty :marriedTo;
```

- ```
 owl:cardinality 0] .
```

- ```
:MarriedPerson owl:subClassOf [
```

- ```
 a owl:Restriction;
```

- ```
    owl:onProperty :marriedTo;
```

- ```
 owl:cardinality 1] .
```

- ```
:MarriedBachelor owl:intersectionOf
```

- ```
 (:Bachelor :MarriedPerson) .
```

- Ezután definiáljunk egy példányt, és ellenőrizzük, hogy van-e ellentmondás

# Érvelési feladatok

- Példányok osztályba tartozása
  - Flipper egy delfin?
- Vizsgáljuk:
  - define  $\neg \text{Dolphin}(\text{Flipper})$
  - Vizsgálni a konzisztenciát
- Osztály felsorolás
  - Ismételni az osztályba tartozás vizsgálatot minden ismert példányra.



# Tipikus következtetési feladatok (összefoglalás)

- Mit várunk egy következtetőtől?
  - Alosztály relációk
    - pl., Minden madár repülő állat?
  - Ekvivalens osztályok
    - pl., Minden madár állat, és fordítva is igaz?
  - Diszjunkt osztályok
    - Pl., Létezik-e olyan állat, amely emlős és madár egyszerre?
  - Osztály konzisztencia
    - Pl.: Létezik olyan emlős, amely tojásokat rak?
  - Osztályok példányai
    - Pl.: Flipper egy delfin?
  - Osztályok felsorolással
    - Pl.: Adjuk meg az összes delfint

# Tipikus következtetési feladatok (összefoglalás)

- Eddig láttuk:
  - Minden következtetési feladat levezethető néhány alap következtetési feladatra, pl. szubsumáció, konzisztencia ellenőrzés
- Tehát építsünk egy következtetőt konzisztencia ellenőrzésre

# Leíró logikai notáció

- Osztályok és példányok
  - $C(x) \leftrightarrow x \text{ a } C$  .
  - $R(x,y) \leftrightarrow x R y$  .
  - $C \sqsubseteq D \leftrightarrow C \text{ rdfs:subClassOf } D$
  - $C \equiv D \leftrightarrow C \text{ owl:equivalentClass } D$
  - $C \sqsubseteq \neg D \leftrightarrow C \text{ owl:disjointWith } D$
  - $C \equiv \neg D \leftrightarrow C \text{ owl:complementOf } D$
  - $C \equiv D \sqcap E \leftrightarrow C \text{ owl:intersectionOf } (D \ E)$  .
  - $C \equiv D \sqcup E \leftrightarrow C \text{ owl:unionOf } (D \ E)$  .
  - $\top$  — `owl:Thing`
  - $\perp$  — `owl:Nothing`



# Leíró logikai notáció

- Tárgyterület, értékkészlet, egyéb korlátozások

$$- \exists R.T \sqsubseteq C \quad R \text{ rdfs:domain } C$$

Példa:  $\exists \text{Apja.leánygyermek}$  fogalom jelöli azon dolgokat, akik Apja szerepben (relációban) vannak leánygyermekükkel, tehát a  $C = \text{Apák}$  osztályából veszi elemeit, azaz a tárgyterülete az Apák osztálya

$$- \forall R.C \quad R \text{ rdfs:range } C$$

Példa:  $\forall \text{Leányapa.leánygyermek}$  fogalom jelöli azon dolgokat, akiknek LeányApa relációban vannak leánygyermekükkel, tehát a Leányapa szerep a leánygyermek osztályából veszi értékeit, azaz értékkészlete a leánygyermek osztály.

$$- C \sqsubseteq \forall R.D \leftrightarrow C \quad \begin{array}{l} \text{owl:subClassOf} \\ [ \text{a owl:Restriction;} \\ \quad \text{owl:onProperty } R; \\ \quad \text{owl:allValuesFrom } D ] . \end{array}$$

$$- C \sqsubseteq \exists R.D \leftrightarrow C \quad \begin{array}{l} \text{owl:subClassOf} \\ [ \text{a owl:Restriction;} \text{ owl:onProperty} \\ \quad R; \text{ owl:someValuesFrom } D ] . \end{array}$$

$$- C \sqsubseteq \geq n R \leftrightarrow C \quad \begin{array}{l} \text{owl:subClassOf} \\ [ \text{a owl:Restriction;} \text{ owl:onProperty} \\ \quad R; \text{ owl:minCardinality } n ] . \end{array}$$

# Negációs normál forma (NNF)

- Ontológiák átalakítása NNF-re
  - $\sqsubseteq$  és  $\equiv$  nem használható
  - Negációt csak atomi formulákra és osztályokra alkalmazható
- Egyszerűbb notáció
- Tabló következtetés formalizmusa

# Negációs normál forma (NNF)

- $\sqsubseteq$  eliminálás:
  - Helyettesítsük  $C \sqsubseteq D$  -t:  $\neg C \sqcup D$
  - Rövidített notáció az ismert formára:  $\forall x: \neg C(x) \vee D(x)$
- Miért
  - $C \sqsubseteq D$  ekvivalens  $C(x) \rightarrow D(x)$

| $C(x)$ | $D(x)$ | $C(x) \rightarrow D(x)$ | $\neg C(x) \vee D(x)$ |
|--------|--------|-------------------------|-----------------------|
| true   | true   | true                    | true                  |
| true   | false  | false                   | false                 |
| false  | true   | true                    | true                  |
| false  | false  | true                    | true                  |



# Negációs normál forma (NNF)

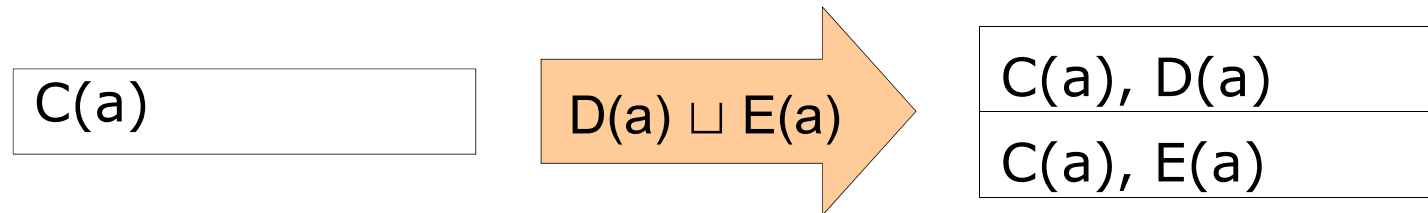
- Elimináljuk  $\equiv$ :
  - Helyettesítsük  $C \equiv D$  -t:  $C \sqsubseteq D$  és  $D \sqsubseteq C$
  - Innen használható a korábbi megoldás
- Pl.:  $C \equiv D$  átírva:
  - $C \sqsubseteq D$
  - $D \sqsubseteq C$
- És így:
  - $\neg C \sqcup D$
  - $\neg D \sqcup C$

# Negációs normál forma (NNF)

- További transzformációs szabályok:
  - $\text{NNF}(C) = C$  (for atomic  $C$ )
  - $\text{NNF}(\neg C) = \neg C$  (for atomic  $C$ )
  - $\text{NNF}(\neg \neg C) = C$
  - $\text{NNF}(C \sqcup D) = \text{NNF}(C) \sqcup \text{NNF}(D)$
  - $\text{NNF}(C \sqcap D) = \text{NNF}(C) \sqcap \text{NNF}(D)$
  - $\text{NNF}(\neg(C \sqcap D)) = \text{NNF}(\neg C) \sqcup \text{NNF}(\neg D)$
  - $\text{NNF}(\neg(C \sqcup D)) = \text{NNF}(\neg C) \sqcap \text{NNF}(\neg D)$
  - $\text{NNF}(\forall R.C) = \forall R.\text{NNF}(C)$
  - $\text{NNF}(\exists R.C) = \exists R.\text{NNF}(C)$
  - $\text{NNF}(\neg \forall R.C) = \exists R.\text{NNF}(\neg C)$
  - $\text{NNF}(\neg \exists R.C) = \forall R.\text{NNF}(\neg C)$

# Tabló következtető (Tableau Algorithm)

- Tábló: Levezetett axiómák gyűjteménye
  - Folyamatosan bővítjük
  - Előrefele láncoláshoz hasonló eljárással
- Például konjunkcióra:
  - Bontsuk ketté a táblót:





# Mikor ellentmondásmentes egy ontológia?

- Tablót bővítjük és bontjuk
- Nincs ellentmondás, ha
  - További axióma nem hozható létre
  - Legalább egy rész tabló ellentmondás mentes
  - Egy rész tabló tartalmaz ellentmondást, ha egy axióma és ellentettje is eleme:
    - Pl..  $\text{Person}(\text{Peter})$  és  $\neg \text{Person}(\text{Peter})$
    - Ilyenkor a rész tablót lezárjuk.

# Tabló következtető

- Adott: egy ontológia NNF formában

Amíg nem minden rész tábló lezárt

\* Válasz egy nyitott táblót  $T$  és egy  $A \in O \cup T$

Ha  $A$ -t nem tartalmazza  $T$ , akkor

Ha  $A$  atomi formula, akkor

adjuk  $A$ -t

$T$ -hez és

vissza \*

Ha  $A$  nem atomi formula, akkor

Válasszunk egy  $i \in O \cup T$

példányt

Adjuk  $A(i)$ -t  $T$ -hez

Vissza \*

egyenként

Bővítsük a táblót  $A$  következményeivel és

vissza \*

# Tabló következtető

- Bővítsük a tablót a következőkkel:

| Nr | Axiom              | Action                                                          |
|----|--------------------|-----------------------------------------------------------------|
| 1  | $C(a)$             | Add $C(a)$                                                      |
| 2  | $R(a,b)$           | Add $R(a,b)$                                                    |
| 3  | $C$                | Choose an individual $a$ , add $C(a)$                           |
| 4  | $(C \sqcap D)(a)$  | Add $C(a)$ and $D(a)$                                           |
| 5  | $(C \sqcup D)(a)$  | Split tableau into T1 and T2.<br>Add $C(a)$ to T1, $D(a)$ to T2 |
| 6  | $(\exists R.C)(a)$ | Add $R(a,b)$ and $C(b)$ for a <i>new</i> Individual $b$         |
| 7  | $(\forall R.C)(a)$ | For all $b$ with $R(a,b) \in T$ : add $C(b)$                    |



# Példa

- Adott egy ontológia:
  - :Animal owl:unionOf (:Mammal :Bird :Fish :Insect :Reptile) .
  - :Animal owl:disjointWith :Human .
  - :Seth a :Human .
  - :Seth a :Insect .
- Konzisztens-e a tudásbázis?

# Példa

- Given the following ontology:
  - :Animal owl:unionOf (:Mammal :Bird :Fish :Insect :Reptile) .
  - :Animal owl:disjointWith :Human .
  - :Seth a :Human .
  - :Seth a :Insect .
- A fenti ontológia DL-NNF alakban:
  - $\neg \text{Animal} \sqcup \neg \text{Human}$
  - $\text{Animal} \sqcup (\neg \text{Mammal} \sqcap \neg \text{Bird} \sqcap \neg \text{Fish} \sqcap \neg \text{Insect} \sqcap \neg \text{Reptile})$
  - $\neg \text{Animal} \sqcup (\text{Mammal} \sqcup \text{Bird} \sqcup \text{Fish} \sqcup \text{Insect} \sqcup \text{Reptile})$
  - Human(Seth)
  - Insect(Seth)
- Végezzük el a következtetést!

# Példa

Human(Seth), Insect(Seth)

| Nr | Axiom  | Action     |
|----|--------|------------|
| 1  | $C(a)$ | Add $C(a)$ |



# Példa

Human(Seth), Insect(Seth),  
( $\neg$ Animal  $\sqcup$   $\neg$ Human)(Seth)

| Nr | Axiom | Action                           |
|----|-------|----------------------------------|
| 3  | C     | Choose an individual a, add C(a) |

# Példa

Human(Seth), Insect(Seth),  
 $\neg$ Animal(Seth)

Human(Seth), Insect(Seth),  
 $\neg$ Human(Seth)

| Nr | Axiom             | Action                                                          |
|----|-------------------|-----------------------------------------------------------------|
| 5  | $(C \sqcup D)(a)$ | Split the tableau into T1 and T2.<br>Add C(a) to T1, D(a) to T2 |

# Példa

Human(Seth), Insect(Seth),  
 $\neg$ Animal(Seth)  
 $\text{Animal} \sqcup (\neg\text{Mammal} \sqcap \neg\text{Bird} \sqcap \neg\text{Fish} \sqcap \neg\text{Insect})(\text{Seth})$

Human(Seth), Insect(Seth),  
 $\neg$ Human(Seth)

| Nr | Axiom | Action                           |
|----|-------|----------------------------------|
| 3  | C     | Choose an individual a, add C(a) |



# Példa

Human(Seth), Insect(Seth),

$\neg$ Animal(Seth)

$\neg$ Animal(Seth)

Human(Seth), Insect(Seth),

$\neg$ Animal(Seth)

$(\neg\text{Mammal} \sqcap \neg\text{Bird} \sqcap \neg\text{Fish} \sqcap \neg\text{Insect} \sqcap \neg\text{Reptile})(\text{Seth})$

Human(Seth), Insect(Seth),

$\neg$ Human(Seth)

| Nr | Axiom | Action |
|----|-------|--------|
|----|-------|--------|

|   |                   |                                                                 |
|---|-------------------|-----------------------------------------------------------------|
| 5 | $(C \sqcup D)(a)$ | Split the tableau into T1 and T2.<br>Add C(a) to T1, D(a) to T2 |
|---|-------------------|-----------------------------------------------------------------|

# Példa

Human(Seth), Insect(Seth),  
 $\neg$ Animal(Seth)  
Animal(Seth)

Human(Seth), Insect(Seth),  
 $\neg$ Animal(Seth)  
 $(\neg\text{Mammal} \sqcap \neg\text{Bird} \sqcap \neg\text{Fish} \sqcap \neg\text{Insect} \sqcap \neg\text{Reptile})(\text{Seth})$   
 $\neg\text{Mammal}(\text{Seth}) \sqcap \neg\text{Bird}(\text{Seth}) \sqcap \neg\text{Fish}(\text{Seth}) \sqcap$   
 $\neg\text{Insect}(\text{Seth}) \sqcap \neg\text{Reptile}(\text{Seth})$

Human(Seth), Insect(Seth),  
 $\neg$ Human(Seth)

| Nr | Aussage           | Aktion            |
|----|-------------------|-------------------|
| 4  | $(C \sqcap D)(a)$ | Add C(a) and D(a) |

# Példa 2

- Egy másik egyszerű ontológia:

```
:Woman rdfs:subClassOf :Person .
:Man rdfs:subClassOf :Person .
:hasChild rdfs:domain :Person .
:hasChild rdfs:range :Person .
:Peter :hasChild :Julia .
:Julia a :Woman .
:Peter a :Man .
```

DL NNF alakban:

```
¬Man ⊑ Person
¬Woman ⊑ Person
¬∃hasChild.T ⊑ Person
∀hasChild.Person
hasChild(Peter,Julia)
Woman(Julia)
Man(Peter)
```

## Példa 2

hasChild(Peter,Julia)

| Nr | Axiom    | Action       |
|----|----------|--------------|
| 2  | $R(a,b)$ | Add $R(a,b)$ |



# Another Example

hasChild(Peter,Julia), Woman(Julia)

| Nr | Axiom | Action   |
|----|-------|----------|
| 1  | C(a)  | Add C(a) |

## Példa 2

hasChild(Peter,Julia), Woman(Julia),  
 $(\neg \exists \text{hasChild.T} \sqcup \text{Person})(\text{Peter})$

| Nr | Axiom | Action                           |
|----|-------|----------------------------------|
| 3  | C     | Choose an individual a, add C(a) |

## Példa 2

hasChild(Peter,Julia), Woman(Julia),  
( $\neg\exists\text{hasChild.T} \sqcup \text{Person}$ )(Peter),  
 $\neg\exists\text{hasChild.T}(\text{Peter})$

hasChild(Peter,Julia), Woman(Julia),  
( $\neg\exists\text{hasChild.T}$ )(Peter), Person(Peter)

| Nr | Axiom               | Action                                                          |
|----|---------------------|-----------------------------------------------------------------|
| 5  | ( $C \sqcup D$ )(a) | Split the tableau into T1 and T2.<br>Add C(a) to T1, D(a) to T2 |

## Példa 2

hasChild(Peter,Julia), Woman(Julia),  
( $\neg\exists\text{hasChild.T}$ )(Peter),  
 $\neg\text{ParentsOfSons(Peter)}$

hasChild(Peter,Julia), Woman(Julia),  
( $\neg\exists\text{hasChild.T}$ )(Peter),  
Person(Peter),  
 $\neg\text{hasChild(Peter,b0),T(b0)}$

| Nr | Axiom              | Action                                                  |
|----|--------------------|---------------------------------------------------------|
| 6  | $(\exists R.C)(a)$ | Add $R(a,b)$ und $C(b)$ for a <i>new</i> Individual $b$ |



## Példa 2

hasChild(Peter,Julia), Woman(Julia),  
( $\neg$ ParentsOfSons  $\sqcup$   $\exists$ hasChild.Man)(Peter),  
 $\neg$ ParentsOfSons(Peter)

hasChild(Peter,Julia), Woman(Julia),  
( $\neg \exists$ hasChild.T)(Peter),  
Person(Peter),  
 $\neg$ hasChild(Peter,b0),T(b0),  
 $\neg$ hasChild(Peter,b1),T(b1),  
...

| Nr | Axiom              | Action                                                  |
|----|--------------------|---------------------------------------------------------|
| 6  | $(\exists R.C)(a)$ | Add $R(a,b)$ und $C(b)$ for a <i>new</i> Individual $b$ |