Αναφορά για την τελική εργασία στο μάθημα Αναπαράσταση Γνώσης στον Παγκόσμιο Ιστό

Στοιχεία Φοιτητή:

Ονοματεπώνυμο: Δημήτριος Κωστορρίζος

ΑΜ: 1054419

Έτος Σπουδών: 5ο

Ερώτημα 1

Α.

Το γνωστικό επίπεδο της οντολογίας που θα παρουσιάσω είναι η ιεραρχία των κλάσεων που μπορεί να χρησιμοποιήσει κάποιος παίκτης στο επιτραπέζιο παιχνίδι Dungeons and Dragons.

Β.

- Η οντολογία θα καλύψει τις βασικές κλάσεις του παιχνιδιού.
- Η οντολογία μπορεί να χρησιμοποιηθεί για να παρουσιάσει τις βασικές κλάσεις και ικανότητες που μπορεί να έχει ένας χαρακτήρας στο παιχνίδι.
- Η οντολογία θα μπορούσε να απαντήσει σε ερωτήσεις επεξήγησης μία κλάσης, αναζήτηση δυνατών συνδυασμών κλάσεων, είδη ικανοτήτων και παρουσίαση γενικών χαρακτηριστικών που έχει ένας χαρακτήρας στο παιχνίδι.
- Αν χρησιμοποιηθεί μηχανισμός συμπερασμού, η οντολογία θα μπορούσε να προβλέπει νέους συνδυασμούς κλάσεων και ικανοτήτων.

Περιγραφή Οντολογίας

Γ.

18 κλάσεις οργανωμένες σε τουλάχιστον τρία επίπεδα ιεραρχίας (υποκλάσεων)

1. PrimaryClass: Αντιπροσωπεύει τις βασικές κλάσεις του παιχνιδιού.
2. Ability: Αντιπροσωπεύει τις βασικές ικανότητες των παικτών του παιχνιδιού.
3. ClassCombinations: Αντιπροσωπεύει τους συνδυασμούς των βασικών κλάσεων του παιχνιδιού.
4. Artificer: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
5. Barbarian: Αντιπροσωπεύει μία κλάση του παιχνιδιού.

6. Bard: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
7. BloodHunter: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
8. Cleric: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
9. Druid: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
10. Fighter: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
11. Monk: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
12. Paladin: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
13. Ranger: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
14. Rogue: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
15. Sorcerer: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
16. Warlock: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
17. Wizard: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
18. Warrior: Αντιπροσωπεύει μία κλάση του παιχνιδιού.

6 κλάσεις να αποτελούν υποκλάσεις άλλων (Subsumption)

1. Expert: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
2. Spellcaster: Αντιπροσωπεύει μία κλάση του παιχνιδιού.
3. DivineSense: Αντιπροσωπεύει μία ικανότητα παίχτη.
4. WildShape: Αντιπροσωπεύει μία ικανότητα παίχτη.
5. SneakAttack: Αντιπροσωπεύει μία ικανότητα παίχτη.
6. ArcaneRecovery: Αντιπροσωπεύει μία ικανότητα παίχτη.

6 κλάσεις να είναι ξένες μεταξύ τους (Disjointness)

1. ActionSurge: Αντιπροσωπεύει μία ικανότητα παίχτη.
2. Rage: Αντιπροσωπεύει μία ικανότητα παίχτη.
3. KiStrike: Αντιπροσωπεύει μία ικανότητα παίχτη.
4. ChannelDivinity: Αντιπροσωπεύει μία ικανότητα παίχτη.
5. BloodCurse: Αντιπροσωπεύει μία ικανότητα παίχτη.
6. StunningStrike: Αντιπροσωπεύει μία ικανότητα παίχτη.

6 κλάσεις να προκύπτουν από λογική σύνθεση άλλων: Χρησιμοποιήστε τουλάχιστον δύο φορές καθεμία από τις παρακάτω πράξεις:

1. τομή (Intersection)
   1.1. SorcererWarlock: Αντιπροσωπεύει τον συνδυασμό των κλάσεων Sorcerer και Warlock του παιχνιδιού.
   1.2. PaladinWarlock: Αντιπροσωπεύει τον συνδυασμό των κλάσεων Paladin και Warlock του παιχνιδιού.

2. ένωση (Union)

2.1. HalfCaster: Αντιπροσωπεύει μία κατηγορία των spellcaster.
2.2. OneThirdCaster: Αντιπροσωπεύει μία κατηγορία των spellcaster.

3. συμπλήρωμα (Complement)
   3.1. HeavyArmorMaster: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.
   3.2. HealthPointsTank: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.

5 κλάσεις να προκύπτουν από περιορισμό (Restriction) σε σχέσεις. Συγκεκριμένα να χρησιμοποιήσετε τουλάχιστον 2 φορές καθένα από τους παρακάτω περιορισμούς:

1. existential restriction (someValuesFrom)
   1.1. Brute: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.
   1.2. UnarmoredWarrior: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.

2. universal restriction (allValuesFrom)
   2.1. Commando: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.
   2.2. DivineWarrior: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.

3. hasValue
   3.1. HexLadin: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.
   3.2. DualWielder: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.

4. Minimum/Maximum Cardinality
   4.1. MultiCaster: Αντιπροσωπεύει μία κατηγορία των spellcaster.
   4.2. EldritchArcher: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.

2 κλάσεις να προκύπτουν από συνδυασμό λογικών πράξεων και περιορισμών σε σχέσεις

- WitchKnight: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.
- SkillExpert: Αντιπροσωπεύει μία κατηγορία των πολεμιστών.

Δ.

Ιδιότητες

16 ιδιότητες οργανωμένες σε τουλάχιστον δυο επίπεδα ιεραρχίας. Να υπάρχουν ιδιότητες και των δυο τύπων (datatypeκαι objectProperties) σε ποσοστό 30% τουλάχιστον από το καθένα.

Data Properties:

- Alignment: Το alignment του χαρακτήρα.
- Level: Το level του χαρακτήρα.
- IsSidekick: Αν είναι sidekick ο χαρακτήρας.
- HitPoints: Τα hitpoints του χαρακτήρα.
- InitiativeBonus: Το initiative bonus του χαρακτήρα.
- PassivePerception: Το passive perception score του χαρακτήρα.
- Name: Το όνομα του χαρακτήρα
- ArmorClass: Το armor class του χαρακτήρα.

Object Properties:

- HasUnarmoredDefence: Δηλώνει αν ο χαρακτήρας έχει Unarmored Defense.
- HasIncreasedWalkingSpeed: Δηλώνει αν ο χαρακτήρας έχει αυξημένο Walking Speed.
- HasSubclass: Δηλώνει αν ο χαρακτήρας έχει sub-κλάση.
- HasMulticlass: Δηλώνει αν ο χαρακτήρας έχει αλλάξει κλάση.

4 ιδιότητες να αποτελούν subproperties άλλων ιδιοτήτων

- AvatarAbilities: Δηλώνει της ικανότητες Avatar του χαρακτήρα.
- AuraAbilities: Δηλώνει της ικανότητες Aura του χαρακτήρα.
- MetamagicAbilities: Δηλώνει της ικανότητες Metamagic του χαρακτήρα.
- RitualCastingAbilities: Δηλώνει της ικανότητες Ritual Casting του χαρακτήρα.

4 ιδιότητες να οριστούν με τις αντίστοιχες αντίστροφες (inverse)

- HasBaseArmorClass: Δηλώνει αν ο χαρακτήρας έχει το προεπιλεγμένο Armor Class.
- HasDefaultWalkingSpeed: Δηλώνει αν ο χαρακτήρας έχει το προεπιλεγμένο Walking Speed.
- HasNotSubclass: Δηλώνει αν ο χαρακτήρας δεν έχει sub-κλάση.
- HasNotMulticlass: Δηλώνει αν ο χαρακτήρας δεν έχει αλλάξει κλάση.

2 ιδιότητες να είναι συμμετρικές (symmetric)

- HasSidekick: Δηλώνει τον βοηθό του χαρακτήρα.
- SpellcastingClass: Δηλώνει την Spell Casting κλάση του χαρακτήρα.

2 ιδιότητες να είναι μεταβατικές(transitive)

- HasAbility: Δηλώνει τις ικανότητες του χαρακτήρα.
- HasArmorProficiency: Δηλώνει τις δεξιοτεχνία του χαρακτήρα, στην χρήση πανοπλιών.

2 ιδιότητες να είναι συναρτησιακές (functional)

- ArcaneAbilities: Δηλώνει τις μαγικές ικανότητες του χαρακτήρα.
- DivineAbilities: Δηλώνει τις θεικές ικανότητες του χαρακτήρα.

2 ιδιότητες να είναι inverse functional

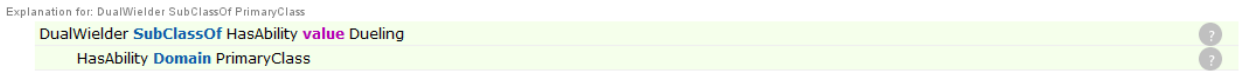
- MartialWeaponTraining: Δηλώνει τις γνώσεις του χαρακτήρα, πάνω στην χρήση απλών όπλων.
- SimpleWeaponTraining: Δηλώνει τις γνώσεις του χαρακτήρα, πάνω στην χρήση εξειδικευμένων όπλων.

Ε.

Τα στιγμιότυπα μπορούν να παρουσιαστούν στην αντίστοιχη καρτέλα στο Protégé.

Ερώτημα 3

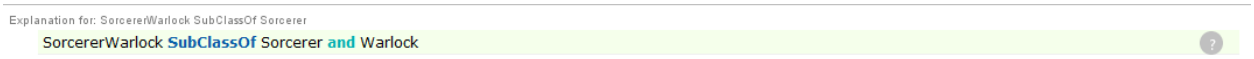
1. Περιγραφή: Η κλάση DualWielder είναι υποκλάση των βασικών κλάσεων του παιχνιδιού.



2. Περιγραφή: Η κλάση PaladinWarlock είναι υποκλάση της κλάσης Paladin.

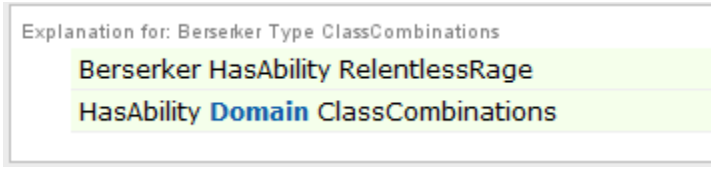
3. Περιγραφή: Η κλάση PaladinWarlock είναι υποκλάση της κλάσης Warlock.

4. Περιγραφή: Η κλάση SorcererWarlock είναι υποκλάση της κλάσης Warlock.

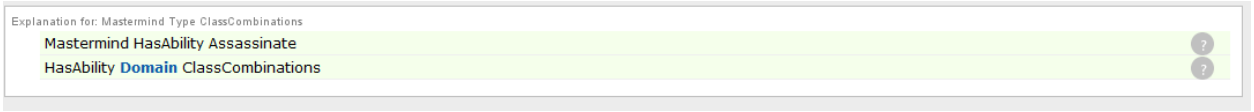


5. Περιγραφή: Η κλάση SorcererWarlock είναι υποκλάση της κλάσης Sorcerer.

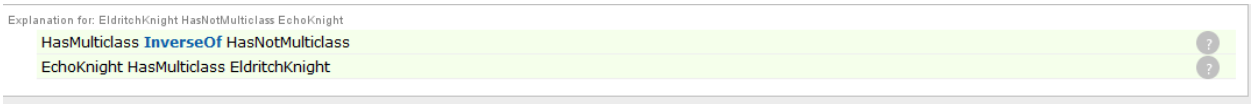
6. Ο Berserker είναι στιγμιότυπο της κλάσης ClassCombinations καθώς πληροί τα κριτήρια του restriction.



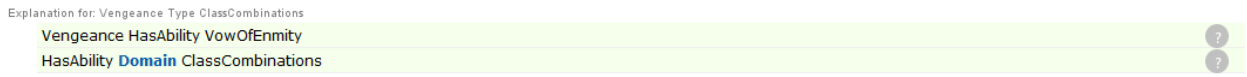
7. Ο Mastermind είναι στιγμιότυπο της κλάσης ClassCombinations και της κλάσης Commando καθώς πληροί τα κριτήρια του restriction.



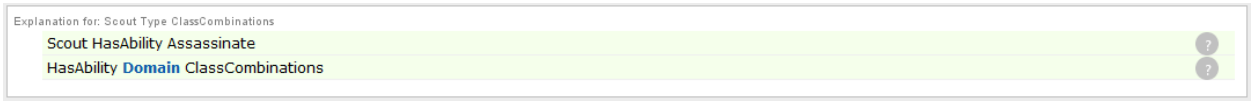
8. Ο EldritchKnight δεν είναι multiclass του EchoKnight, καθώς ο EchoKnight είναι multiclass του EldritchKnight.



9. Ο Vengeance Paladin είναι της κλάσης ClassCombinations και της κλάσης DivineWarrior καθώς ικανοποιεί τα κριτήρια του restriction.



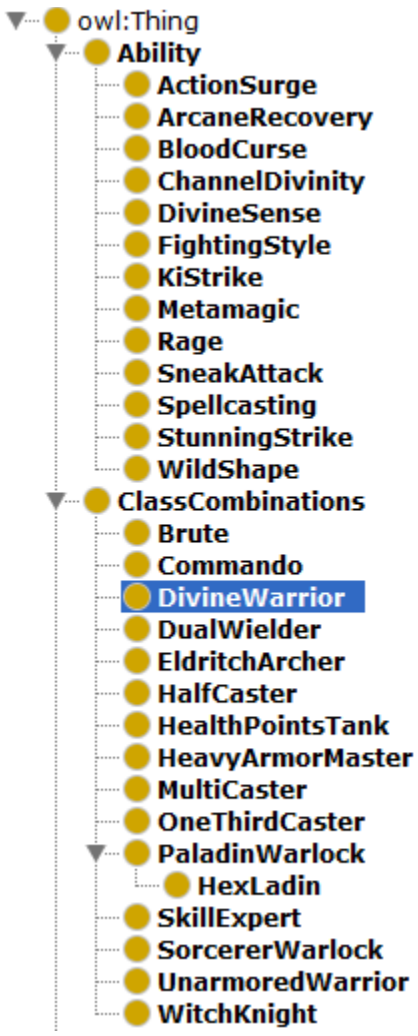
10. Ο Scout είναι στιγμιότυπο της κλάσης ClassCombinations και της κλάσης Commando καθώς πληροί τα κριτήρια του restriction.



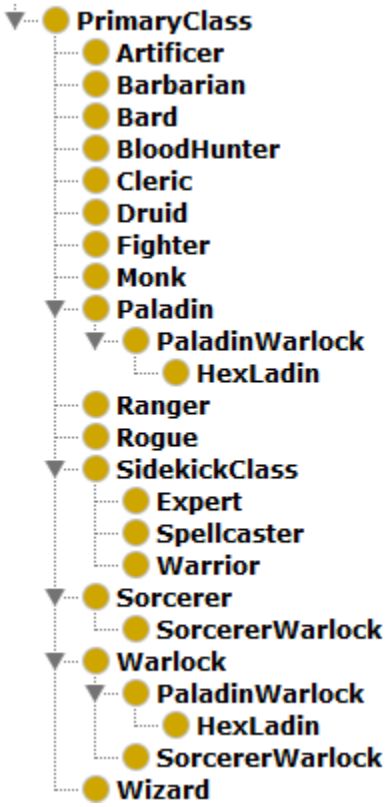
Ερώτημα 4

A.

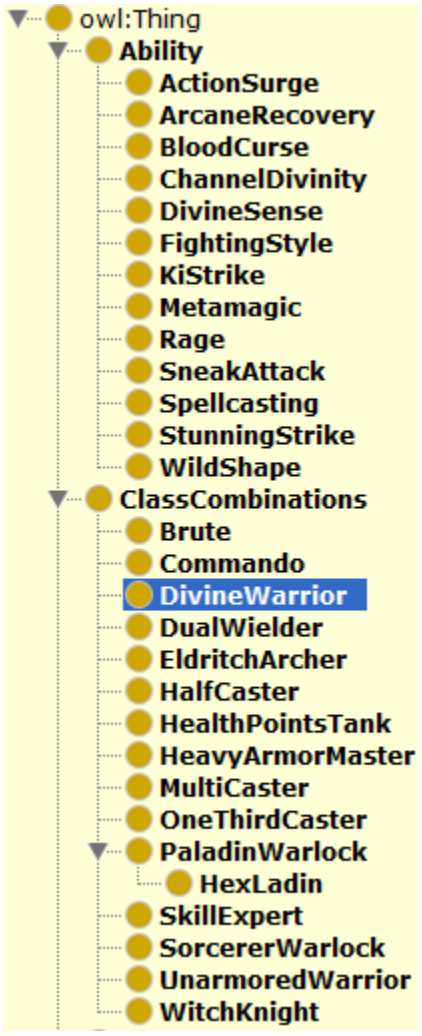
Class Hierarchy Asserted, Μέρος 1



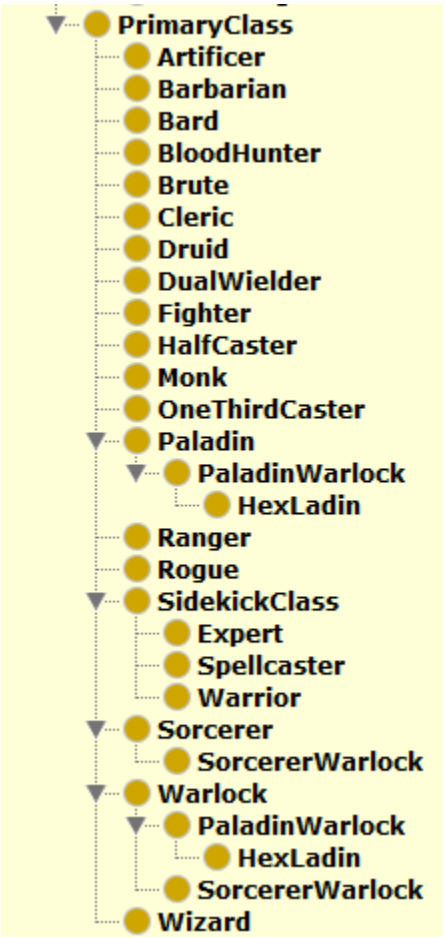
Class Hierarchy, Μέρος 2



Class Hierarchy Inferred, Μέρος 1

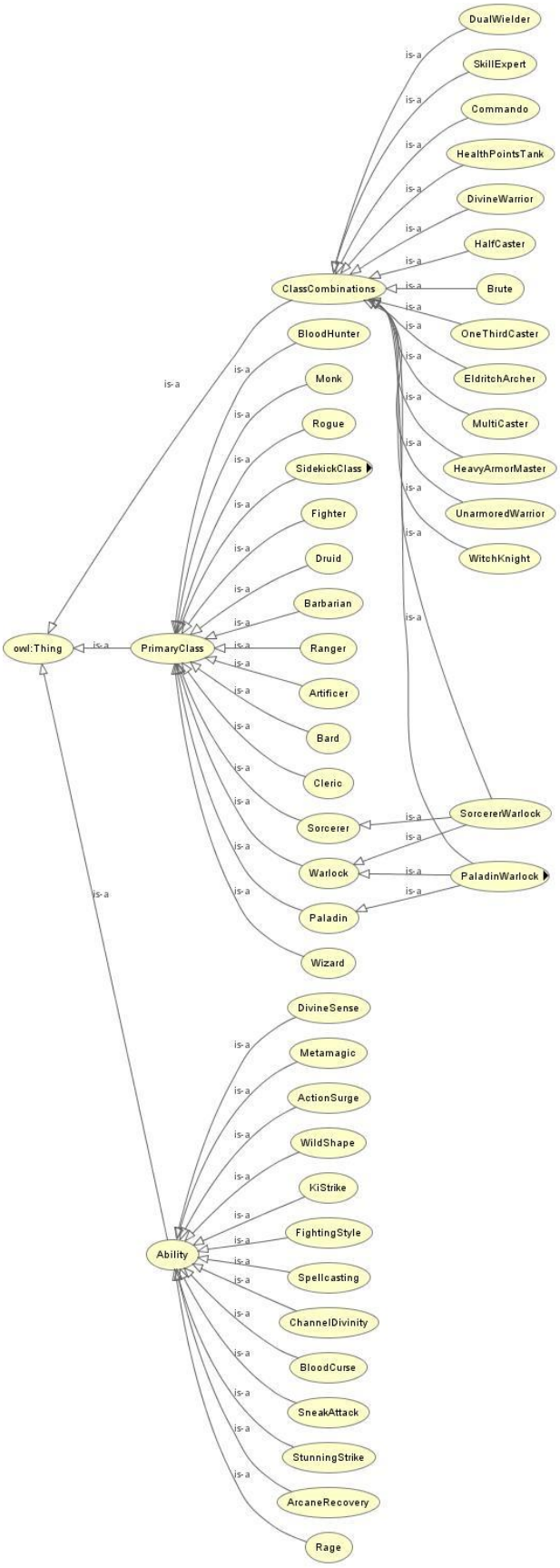


Class Hierarchy Inferred, Μέρος 2

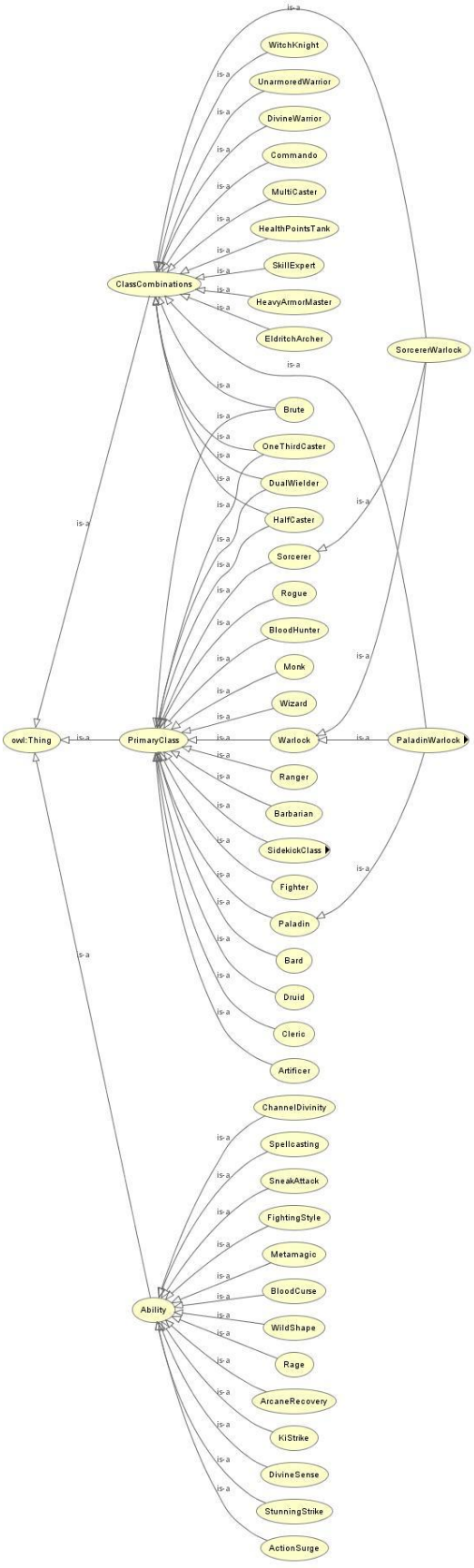


B.

Class Hierarchy Graph Asserted


owl:Thing
is-a
PrimaryClass
ClassCombinations
DualWielder
SkillExpert
Commando
HealthPointsTank
DivineWarrior
HalfCaster
Brute
OneThirdCaster
EldritchArcher
MultiCaster
HeavyArmorMaster
UnarmoredWarrior
WitchKnight
SorcererWarlock
PaladinWarlock
BloodHunter
Monk
Rogue
SidekickClass
Fighter
Druid
Barbarian
Ranger
Artificer
Bard
Cleric
Sorcerer
Warlock
Paladin
Wizard
Ability
DivineSense
Metamagic
ActionSurge
WildShape
KiStrike
FightingStyle
Spellcasting
ChannelDivinity
BloodCurse
SneakAttack
StunningStrike
ArcaneRecovery
Rage

Class Hierarchy Graph Inferred


is-a
WitchKnight
UnarmoredWarrior
DivineWarrior
Commando
MultiCaster
HealthPointsTank
SkillExpert
ClassCombinations
HeavyArmorMaster
EldritchArcher
SorcererWarlock
Brute
OneThirdCaster
DualWielder
HalfCaster
Sorcerer
Rogue
BloodHunter
Monk
Wizard
owl:Thing
PrimaryClass
Warlock
PaladinWarlock
Ranger
Barbarian
SidekickClass
Fighter
Paladin
Bard
Druid
Cleric
Artificer
ChannelDivinity
Spellcasting
SneakAttack
FightingStyle
Metamagic
BloodCurse
Ability
WildShape
Rage
ArcaneRecovery
KiStrike
DivineSense
StunningStrike
ActionSurge

C.

Όπως διαφαίνεται και από τις παραπάνω εικόνες, δεν υπάρχουν διαφορές μεταξύ των Asserted και Inferred ιεραρχιών των κλάσεων.

Ερώτημα 5

A.

1. **Query:**

```
SELECT ?wizard
 WHERE
{
        ?wizard rdf:type <http://www.semanticweb.org/DnD_Classes_Ontology#Wizard>.
        OPTIONAL
        {
            ?wizard <http://www.semanticweb.org/DnD_Classes_Ontology#Level> 5
        }
}
ORDER BY ?wizard
```

Περιγραφή: Δείξε όλους τους wizard, είτε έχουν είτε δεν έχουν ως level την τιμή 5, ταξινομημένα κατά όνομα.

Αποτελέσματα:

<http://www.semanticweb.org/DnD_Classes_Ontology#Bladesinger>
<http://www.semanticweb.org/DnD_Classes_Ontology#Necromancer>

2. **Query:**

```
SELECT ?ability WHERE
{
```

```
        {
            ?ability rdf:type
<http://www.semanticweb.org/DnD_Classes_Ontology#ChannelDivinity>.
        }
        UNION
        {
            ?ability rdf:type
<http://www.semanticweb.org/DnD_Classes_Ontology#DivineSense>
        }
}

ORDER BY ?ability
```

Περιγραφή: Δείξε όλα τα Ability τύπου ChannelDivinity και DivineSense, ταξινομημένα κατά όνομα.

Αποτελέσματα:

<http://www.semanticweb.org/DnD_Classes_Ontology#FiendSense>
<http://www.semanticweb.org/DnD_Classes_Ontology#VowOfEnmity>

3. Query:

```
SELECT (MAX(?level) AS ?MaxLevel) WHERE
{
        {
            ?instance rdf:type
<http://www.semanticweb.org/DnD_Classes_Ontology#PrimaryClass>.
            ?instance <http://www.semanticweb.org/DnD_Classes_Ontology#Level> ?level
        }
}
```

Περιγραφή: Δείξε το μέγιστο level των χαρακτήρων.

Αποτελέσματα:

20.0

4. Query:

SELECT **?instance** WHERE

{

?instance rdf:type <http://www.semanticweb.org/DnD_Classes_Ontology#PrimaryClass>.

?instance <http://www.semanticweb.org/DnD_Classes_Ontology#Level> ?level

FILTER(?level > 6)

}

Περιγραφή: Δείξε τους χαρακτήρες με ελάχιστο level την τιμή 6.

Αποτελέσματα:

<http://www.semanticweb.org/DnD_Classes_Ontology#Necromancer>
<http://www.semanticweb.org/DnD_Classes_Ontology#EchoKnight>
<http://www.semanticweb.org/DnD_Classes_Ontology#Berserker>

5. Query:

SELECT DISTINCT **?rogue** WHERE

{

?rogue rdf:type <http://www.semanticweb.org/DnD_Classes_Ontology#Rogue>.

?rogue <http://www.semanticweb.org/DnD_Classes_Ontology#Level> ?level

FILTER(?level >= 3)

OPTIONAL{**?rogue** <http://www.semanticweb.org/DnD_Classes_Ontology#HasAbility> <http://www.semanticweb.org/DnD_Classes_Ontology#Assassinate>}

}

Περιγραφή: Δείξε τους μοναδικούς χαρακτήρες, με ελάχιστο level την τιμή 3, που μπορεί να έχουν ή και όχι, το Ability Assasinate.

Αποτελέσματα:

<http://www.semanticweb.org/DnD_Classes_Ontology#Scout>
<http://www.semanticweb.org/DnD_Classes_Ontology#Mastermind>

Β.

1. **Query:**

Barbarian(?b) -> ArmorClass(?b, 16)

Περιγραφή: Θέτει την τιμή του property ArmorClass των στιγμιότυπων της κλάσης Barbarian στην τιμή 16.

Αποτελέσματα:

Berserker: ArmorClass = 16

2. **Query:**

Cleric(?c) -> Alignment(?c, "Lawful")

Περιγραφή: Θέτει την τιμή του property Alignment των στιγμιότυπων της κλάσης Cleric στην τιμή “Lawful”.

Αποτελέσματα:

Life: Alignment = “Lawful”

3. **Query:**

Ranger(?r) -> Level(?r, 9)

Περιγραφή: Θέτει την τιμή του property Level των στιγμιότυπων της κλάσης Ranger στην τιμή 9.

Αποτελέσματα:

GloomStalker: Level = 9

4. **Query:**

Warlock(?w) ^ SpellcastingClass(?w, ?sc) -> HasSubclass(?w, ?sc)

Περιγραφή: Θέτει την τιμή του property HasSubclass των στιγμιότυπων της κλάσης Warlock στην τιμή του SpellcastingClass.

Αποτελέσματα:

HexBlade: HasSubclass = Warlock

5. **Query:**

Monk(?m) -> HasAbility(?m, KiFist)

Περιγραφή: Θέτει την τιμή του property HasAbility των στιγμιότυπων της κλάσης Monk στην τιμή του KiFist.

Αποτελέσματα:

Kensei: HasAbility = KiFist

Ερώτημα 6

Ως **open-world assumption** θεωρούμε την υπόθεση, κατά την οποία για την διεξαγωγή του συμπεράσματος της υπόθεσης, θεωρείται ότι η αλήθεια μίας δήλωσης μπορεί να ισχύει, ανεξαρτήτως του αν είναι ευρέως γνωστή η δήλωση.

Παράδειγμα στην οντολογία: Το στιγμιότυπο HexOathbreakerPaladin ανήκει στην Paladin και στην κλάση Warlock, επειδή ανήκει στην κλάση PaladinWarlock.

Ως **non-unique-name assumption** θεωρούμε την υπόθεση, κατά την οποία στιγμιότυπα με διαφορετικά ονόματα-ετικέτες ανήκουν στην ίδια κλάση.

Αυτού του είδους την υπόθεση εφαρμόζει το Protégé, όταν υπάρχει κάποια δήλωση στο πεδίο EquivalentTo.

Παράδειγμα στην οντολογία: Η κλάση OneThirdCaster είναι αντίστοιχη με την κλάση MultiCaster.

Ερώτημα 7

Η εφαρμογή συντάχθηκε σε C# 8, ως μία .NET Command Line εφαρμογή, χρησιμοποιώντας το NuGet package donnetrdf για τις δυνατότητες διαχείρισης του αρχείου οντολογίας .owl.

Η εφαρμογή μεταφέρει το περιεχόμενο του owl αρχείου στην μνήμη και εφαρμόζει τον Static Rdfs Reasoner. Έπειτα, εμφανίζει στην οθόνη του χρήστη τις επιλογές που έχει. Αν ο χρήστης επιλέξει τον αριθμό 1, τότε εμφανίζονται όλες οι ιδιότητες που υπάρχουν στο αρχείο. Έπειτα, ο χρήστης πρέπει να εισάγει το URI της ιδιότητας που επιθυμεί και έπειτα την τιμή της ιδιότητας, ώστε να εμφανιστούν τα στιγμιότυπα που έχουν την ιδιότητα με την συγκεκριμένη τιμή. Αν ο χρήστης επιλέξει τον αριθμό 2 και εισάγει το Uri του νέου στιγμιότυπου, το στιγμιότυπο αυτό προστίθεται στο αρχείο. Αν ο χρήστης επιλέξει τον αριθμό 3, τότε εμφανίζονται όλες οι κλάσεις που υπάρχουν στο αρχείο. Έπειτα, ο χρήστης πρέπει να εισάγει το URI της κλάσης που επιθυμεί, ώστε να εμφανιστούν τα στιγμιότυπα που ανήκουν σε αυτήν την κλάση.

Η οντολογία χρησιμοποιηθεί τον Reasoner όπως ακριβώς χρησιμοποιεί το Protégé τον Pellet Reasoner, για να καταλήγει σε πορίσματα και συμπεράσματα. Ο Reasoner χρησιμοποιεί τις τριπλέτες, εφαρμόζοντας διάφορες μορφές συλλογισμού, ώστε να δημιουργήσει νέες τριπλέτες γνώσης.

Κώδικας για αναζήτηση στιγμιότυπων ανά κλάση:

```
1 reference
public static void PrintInstances(OntologyGraph ontologyGraph)
{
    // Get all classes
    var classes = ontologyGraph.AllClasses;

    // For every class...
    foreach (var owlClass in classes)
    {
        // Print the uri
        Console.WriteLine(owlClass.Resource);
    }

    Console.WriteLine("Enter the Uri of the class.\n");

    // Read the user choice
    var selectedUri = new Uri(Console.ReadLine());

    Console.WriteLine("\nClass Instances:\n");

    // Get the selected class
    var selectedClass = classes.First(x => x.Resource.ToString() == selectedUri.ToString());

    // For every class...
    foreach (var instance in selectedClass.Instances)
    {
        // Print the uri
        Console.WriteLine(instance.Resource);
    }
}
```

Κώδικας για αναζήτηση στιγμιότυπων ανά ιδιότητα, Μέρος 1:

```
/// <summary>
/// The function for printing the instances
/// </summary>
/// <param name="ontologyGraph">The owl graph</param>
1 reference
public static void SearchInstances(OntologyGraph ontologyGraph)
{
    // Get all defined properties
    var properties = ontologyGraph.AllProperties;

    // For every property...
    foreach (var property in properties)
    {
        // Print the property name
        Console.WriteLine(property.Resource);
    }

    Console.WriteLine("\nEnter the Uri of the property.\n");

    // Read the user choice
    var selectedUri = new Uri(Console.ReadLine());

    // Get the selected property
    var selectedProperty = properties.First(x => x.Resource.ToString() == selectedUri.ToString());

    Console.WriteLine("\nEnter the property value:\n");

    // Read the user choice
    var input = Console.ReadLine();

    // Get all the triples
    var triples = selectedProperty.TriplesWithPredicate.Where(x =>
```

Κώδικας για αναζήτηση στιγμιότυπων ανά ιδιότητα, Μέρος 2:

```
// Get all the triples
var triples = selectedProperty.TriplesWithPredicate.Where(x =>
{
    // If the object node is literal...
    if (x.Object.NodeType == NodeType.Literal)
    {
        // Typecast the node
        var node = x.Object as LiteralNode;

        // Check the value
        return node.Value == input;
    }
    else
    {
        // Typecast the node
        var node = x.Object as UriNode;

        // Check the uri
        return node.Uri.ToString() == input;
    }
});

Console.WriteLine("\nInstances:\n");

// For every triple...
foreach (var triple in triples)
{
    // Print the triple subject
    Console.WriteLine(triple.Subject);
}
```

Κώδικας για δημιουργία στιγμιότυπων:

```
/// <summary>
/// The function for adding instances
/// </summary>
/// <param name="ontologyGraph">The owl graph</param>
/// <param name="owlFilePath">The owl file path</param>
1 reference
public static void AddInstances(OntologyGraph ontologyGraph, string owlFilePath)
{
    // Declare an empty uri node
    var uriNode = ontologyGraph.CreateUriNode();

    // Set the uri of the node
    uriNode.GraphUri = new Uri(Console.ReadLine());

    // Add the uri node as an individual
    ontologyGraph.CreateIndividual(uriNode);

    // Save the graph to the file
    ontologyGraph.SaveToFile(owlFilePath);

    Console.WriteLine("Updated was successful.");
}
```