

Protégé Tutorial

Fabio Papacchini

Protégé – What and Where

What is Protégé? (from their webpage)

A free, open-source ontology editor and framework for building intelligent systems

Protégé is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organisational modelling.

Where to get it: <http://protege.stanford.edu/>

Useful resources

- ▶ http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf

NOTE: the manual is for version 4, but the current version is 5.1

- ▶ http://protegewiki.stanford.edu/wiki/Main_Page

Protégé – What and Where (cont'd)

Specifically, Protégé is

- ▶ a java-based application (multi-platform)
- ▶ thought for a variety of people (more than 300 thousands users)
- ▶ a GUI to help the editing of ontologies
creation, modification, reasoning, debugging, . . .

Syntax – DL, OWL, Manchester

Protégé uses the Manchester syntax

DL	OWL	Manchester
\top	owl:Thing	owl:Thing
\perp	owl:Nothing	owl:Nothing
Concept name	Class	Class
Role name	Object property	Object property
$\neg C$	ObjectComplementOf(C)	not C
$C \sqcup D$	ObjectUnionOf(C D)	C or D
$C \sqcap D$	ObjectIntersectionOf(C D)	C and D
$\exists r.C$	ObjectSomeValuesFrom(r C)	r some C
$\forall r.C$	ObjectAllValuesFrom(r C)	r only C
$(\geq n \ r.C)$	ObjectMinCardinality(n r C)	r min n C
$(\leq n \ r.C)$	ObjectMaxCardinality(n r C)	r max n C
$(= n \ r.C)$	ObjectExactCardinality(n r C)	r exactly n C

<https://www.w3.org/TR/owl2-manchester-syntax/>

Syntax – DL, OWL, Manchester – Example

DL

Person $\sqcap \exists \text{hasGender.Male}$

$(= 2 \text{ hasWheel.FrontWheel}) \sqcap (= 2 \text{ hasWheel.RearWheel})$

OWL (omitting “Object” for succinctness)

IntersectionOf(Person SomeValuesFrom(hasGender Male))

IntersectionOf(ExactCardinality(2 hasWheel FrontWheel)
ExactCardinality(2 hasWheel RearWheel))

Manchester

Person and (hasGender some Male)

(hasWheel exactly 2 FrontWheel) and (hasWheel exactly 2
RearWheel)

Convention

- ▶ concept names begin with an uppercase letter
- ▶ role names begin with a lowercase letter
- ▶ CamelBack notation for both concept and role names

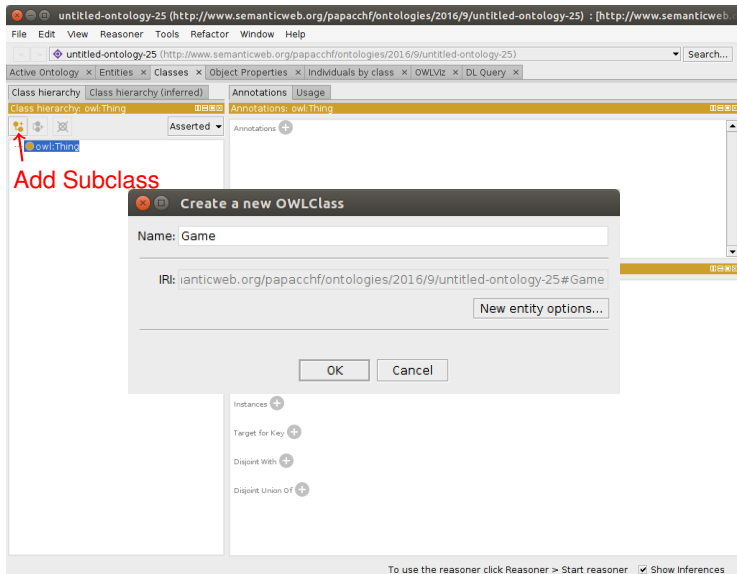
An Ontology about Video Games

Assume we want to build an ontology about video games as follows.

self-standing	modifiers	relations	definable
- Game	- Genre	hasDifficulty	MultiPlatform
- NamedGame	- SinglePlayer	hasPlatform	PuzzleGame
- LoL	- MultiPlayer	hasGenre	HardGame
- Chess	- Puzzle		NormalGame
- Sudoku	- RolePlayGame		EasyGame
- WoW	- Online		LinuxGame
- Platform	- Difficulty		WindowsGame
- Windows	- Hard		MacOSXGame
- MacOSX	- Normal		...
- Linux	- Easy		

Adding Classes

Make sure to have the “Classes” tab open
Window → Tabs → Classes



The screenshot shows the Protégé interface with the 'Classes' tab selected. A red arrow points to the 'owl:Thing' class in the 'Class hierarchy' pane, with the text 'Add Subclass' written in red. A dialog box titled 'Create a new OWLClass' is open, showing the 'Name' field set to 'Game' and the 'IRI' field set to 'http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25#Game'. The 'New entity options...' button is visible. At the bottom of the dialog, there are 'OK' and 'Cancel' buttons. Below the dialog, there are several options with plus icons: 'Instances', 'Target for Key', 'Disjoint With', and 'Disjoint Union Of'. The status bar at the bottom indicates 'To use the reasoner click Reasoner > Start reasoner' and 'Show Inferences' is checked.

untitled-ontology-25 (http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25) : [http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLViz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: owl:Thing Annotations: owl:Thing

Asserted Annotations

owl:Thing

Add Subclass

Create a new OWLClass

Name: Game

IRI: http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25#Game

New entity options...

OK Cancel

Instances

Target for Key

Disjoint With

Disjoint Union Of

To use the reasoner click Reasoner > Start reasoner Show Inferences

Adding Classes

Make sure to have the “Classes” tab open
Window → Tabs → Classes

The screenshot displays the Protégé ontology editor. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main window has a tab bar with 'Active Ontology', 'Entities', 'Classes', 'Object Properties', 'Individuals by class', 'OWLviz', and 'DL Query'. The 'Classes' tab is active, showing a 'Class hierarchy' panel on the left and a 'Description' panel on the right. The 'Class hierarchy' panel shows a tree structure with 'owl:Thing' as the root and 'Game' as a child. Above the hierarchy are three icons: a plus sign for adding a sibling, a minus sign for removing a child, and an X for deleting a class. Two red arrows point to the plus and X icons, with labels 'Add Sibling' and 'Delete Class' respectively. The 'Description' panel on the right shows various properties for the 'Game' class, such as 'Equivalent To', 'SubClass Of', 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', 'Target for Key', 'Disjoint With', and 'Disjoint Union Of'. The bottom status bar indicates 'To use the reasoner click Reasoner > Start reasoner' and 'Show Inferences' is checked.

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) : (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>)

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: Game Annotations: Game

owl:Thing
Game

Add Sibling
Delete Class

Description: Game

Equivalent To +
SubClass Of +
General class axioms +
SubClass Of (Anonymous Ancestor)
Instances +
Target for Key +
Disjoint With +
Disjoint Union Of +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy...

The screenshot shows the Protégé ontology editor interface. The title bar indicates the ontology is 'untitled-ontology-25' located at 'http://www.semanticweb.org/papacch/ontologies/2016/9/untitled-ontology-25'. The 'Tools' menu is open, showing options: 'Create class hierarchy...', 'Create axioms from Excel workbook...', 'Export OWLDoc...', 'Generate java code...', and 'Usage...'. The 'Create class hierarchy...' option is highlighted. In the background, the 'Class hierarchy: owl:Thing' panel is visible, showing a tree structure with 'Game', 'Genre', and 'Platform' as subclasses of 'owl:Thing'. The 'Description: owl:Thing' panel is also visible, showing various logical constraints like 'Equivalent To', 'SubClass Of', 'General class axioms', etc.

untitled-ontology-25 (http://www.semanticweb.org/papacch/ontologies/2016/9/untitled-ontology-25) : http://www.semanticweb.org/papacch/ontologies/2016/9/untitled-ontology-25

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25

Active Ontology x Entities x

Class hierarchy: owl:Thing

Class hierarchy: owl:Thing

Found 4 uses of owl:Thing

- Game
 - Game SubClassOf owl:Thing
- Genre
 - Genre SubClassOf owl:Thing
- Platform
 - Platform SubClassOf owl:Thing

Description: owl:Thing

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

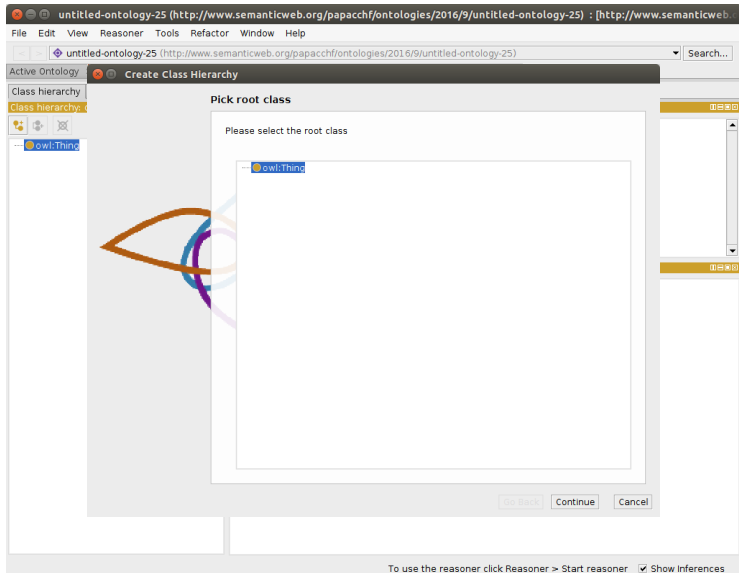
Disjoint Union Of +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Class Hierarchies

It allows us to speed up the process of adding classes.

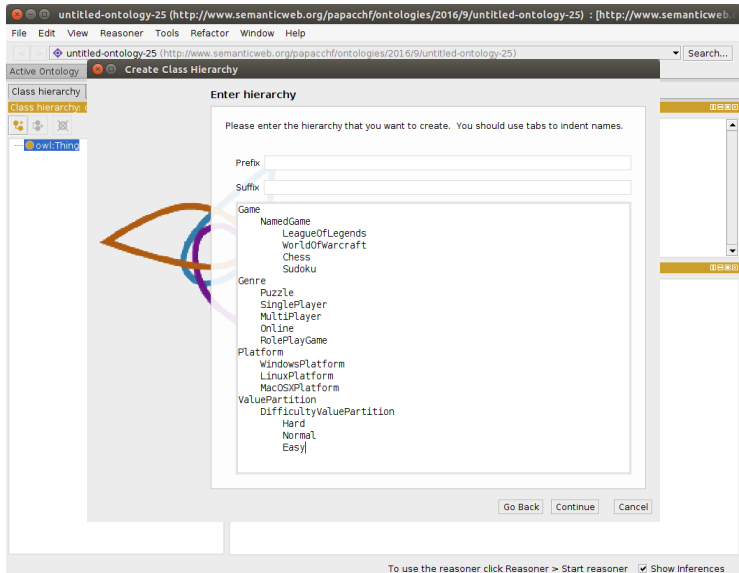
Tools → Create class hierarchy...



Adding Class Hierarchies

It allows us to speed up the process of adding classes.

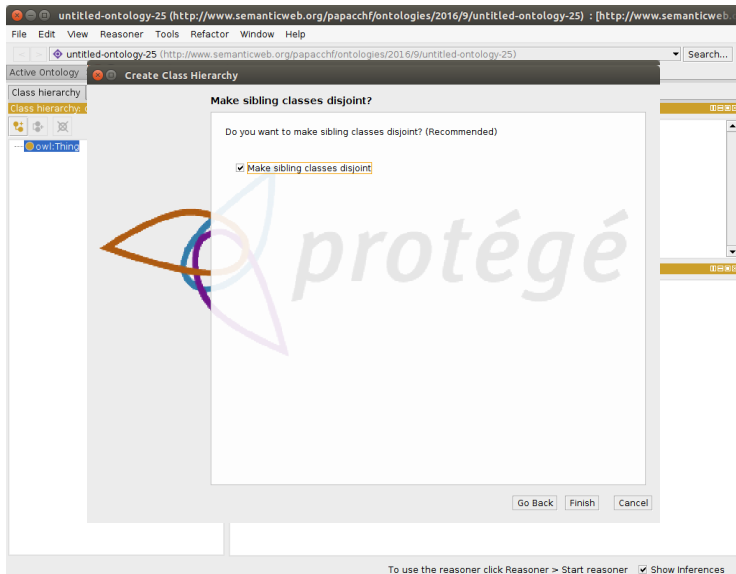
Tools → Create class hierarchy...



Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy...



Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy. . .

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The toolbar contains icons for navigating between different views: Class hierarchy, Class hierarchy (inferred), Annotations, and Usage. The 'Usage' tab is selected, showing a list of axioms involving the 'Game' class. A red arrow points to the 'Usage' tab, and another red arrow points to the 'Game' class in the left-hand class hierarchy tree.

The class hierarchy tree on the left shows the following structure:

- owl:Thing
 - Game
 - NamedGame
 - Chess
 - LeagueOfLegends
 - Sudoku
 - WorldOfWarcraft
 - Platform
 - LinuxPlatform
 - MacOSXPlatform
 - WindowsPlatform
 - Genre
 - MultiPlayer
 - Online
 - Puzzle
 - RolePlayGame
 - SinglePlayer
 - ValuePartition
 - DifficultyValuePartition
 - Easy
 - Hard
 - Normal

The 'Usage: Game' panel on the right shows the following axioms:

- Found 10 uses of Game
 - Game
 - SubClassOf owl:Thing
 - Class: Game
 - DisjointClasses: Game, Genre, Platform, ValuePartition
 - Genre
 - DisjointClasses: Game, Genre, Platform, ValuePartition
 - hasDifficulty

The 'Description: Game' panel on the right shows the following information:

- Equivalent To: +
- SubClass Of: +
 - owl:Thing
- General class axioms: +
- SubClass Of (Anonymous Ancestor):
- Instances: +
- Target for Key: +
- Disjoint With: +
 - ValuePartition, Genre, Platform
- Disjoint Union Of: +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy...

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main toolbar contains icons for navigating between different views: Class hierarchy, Class hierarchy (inferred), Annotations, and Usage. The left pane shows the 'Class hierarchy' view for the 'Game' class, with a red arrow pointing to the 'Game' class under 'owl:Thing'. The right pane shows the 'Usage' view for the 'Game' class, displaying a list of uses including 'Game SubClassOf owl:Thing', 'Class: Game', and 'DisjointClasses: Game, Genre, Platform, ValuePartition'. The bottom pane shows the 'Description' view for the 'Game' class, which is highlighted with a red border. The description includes sections for 'Equivalent To', 'SubClass Of' (listing 'owl:Thing'), 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', 'Target for Key', 'Disjoint With' (listing 'ValuePartition, Genre, Platform'), and 'Disjoint Union Of'. A red text overlay 'Description of the class "Game"' is positioned over the description pane.

untitled-ontology-25 (http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25) : [http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25)

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: Game Usage: Game

Show: ☒ this ☒ disjoint ☒ named sub/superclasses

Found 10 uses of Game

- Game
 - Game SubClassOf owl:Thing
 - Class: Game
 - DisjointClasses: Game, Genre, Platform, ValuePartition
- Genre
 - DisjointClasses: Game, Genre, Platform, ValuePartition
- hasDifficulty

Description: Game

Equivalent To +

SubClass Of +

- owl:Thing

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

- ValuePartition, Genre, Platform

Disjoint Union Of +

Description of the class "Game"

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy...

The screenshot shows the Protégé ontology editor interface. The left pane displays a class hierarchy starting from `owl:Thing`, with `Game` highlighted by a red arrow. The right pane shows the 'Usage: Game' tab, indicating that `Game` is a subclass of `owl:Thing` and is disjoint with `Genre`, `Platform`, and `ValuePartition`. The bottom pane shows the 'Description: Game' tab, which is highlighted with a red box. In this tab, the 'SubClass Of' section shows `owl:Thing` as the parent class. The text `Game ⊆ T` is written in red next to the `owl:Thing` entry. The bottom status bar indicates that the reasoner is running and inferences are shown.

Game \sqsubseteq T

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy...

The screenshot shows the Protégé ontology editor interface. The left pane displays a class hierarchy starting with `owl:Thing`, which includes `Game`, `NamedGame`, `Platform`, `Genre`, and `ValuePartition`. The `Game` class is highlighted with a red arrow. The right pane shows the 'Usage: Game' tab, which lists 10 uses of the class. The 'Description: Game' tab is also visible, showing various axioms and relationships. A red box highlights the 'Disjoint With' section, which lists `ValuePartition`, `Genre`, and `Platform`.

Class hierarchy: Game

- owl:Thing
 - Game
 - NamedGame
 - Chess
 - LeagueOfLegends
 - Sudoku
 - WorldOfWarcraft
 - Platform
 - LinuxPlatform
 - MacOSXPlatform
 - WindowsPlatform
 - Genre
 - MultiPlayer
 - Online
 - Puzzle
 - RolePlayGame
 - SinglePlayer
 - ValuePartition
 - DifficultyValuePartition
 - Easy
 - Hard
 - Normal

Usage: Game

Show: ☒ this ☒ disjoint ☒ named sub/superclasses

Found 10 uses of Game

- Game
 - SubClassOf owl:Thing
 - Class: Game
 - DisjointClasses: Game, Genre, Platform, ValuePartition
- Genre
 - DisjointClasses: Game, Genre, Platform, ValuePartition
- hasDifficulty

Description: Game

Equivalent To: +

SubClass Of: +
owl:Thing

General class axioms: +

SubClass Of (Anonymous Ancestor)

Instances: +

Target for Key: +

Disjoint With: +
ValuePartition, Genre, Platform

Disjoint Union Of: +

$Game \sqcap ValuePartition \sqsubseteq \perp$
 $Game \sqcap Genre \sqsubseteq \perp$
 $Game \sqcap Platform \sqsubseteq \perp$

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Class Hierarchies

It allows us to speed up the process of adding classes.

Tools → Create class hierarchy...

The screenshot shows the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main window has tabs for Active Ontology, Entities, Classes, Object Properties, Individuals by class, OWLviz, and DL Query. The 'Classes' tab is active, showing a class hierarchy for 'Game'. A red arrow points to the 'Game' class in the hierarchy. The 'Description: Game' panel is open, showing various axioms. A red box highlights the 'Delete axiom' and 'Edit axiom' buttons in the bottom right corner of the description panel.

untitled-ontology-25 (http://www.semanticweb.org/papacch/ontologies/2016/9/untitled-ontology-25) : [http://www.semanticweb.org/papacch/ontologies/2016/9/untitled-ontology-25]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (http://www.semanticweb.org/papacch/ontologies/2016/9/untitled-ontology-25)

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: Game Usage: Game

Show: ☒ this ☒ disjoint ☒ named sub/superclasses

Found 10 uses of Game

- Game
 - Game SubClassOf owl:Thing
 - Class: Game
 - DisjointClasses: Game, Genre, Platform, ValuePartition
- Genre
 - DisjointClasses: Game, Genre, Platform, ValuePartition
- hasDifficulty

Description: Game

Equivalent To +

SubClass Of +

- owl:Thing

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

- ValuePartition, Genre, Platform

Disjoint Union Of +

Delete axiom

Edit axiom

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

What Now?

What we have...

- ▶ all non-definable classes
- ▶ an initial class hierarchy
- ▶ basic (among siblings) disjoint axioms

What we need to add...

- ▶ object properties
- ▶ relations between classes
- ▶ definable classes

Object Properties (Domain and Range)

Make sure to have the “Object Properties” tab open
Window → Tabs → Object Properties

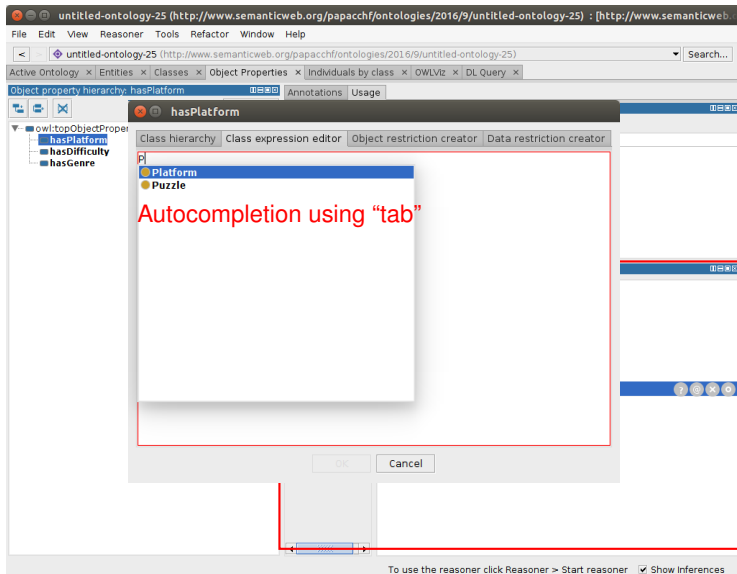
The screenshot shows the Semantic Web editor interface. The 'Object Properties' tab is active, displaying the 'Usage: hasPlatform' section. The 'Object property hierarchy: hasPlatform' is visible on the left, showing a tree structure with 'hasPlatform' as the root, and 'hasDifficulty' and 'hasGenre' as sub-properties. The 'Usage: hasPlatform' section shows 'Found 1 uses of hasPlatform' and a list of 'ObjectProperty: hasPlatform'. The 'Characteristics' section is expanded, showing various properties like 'Functional', 'Inverse function', 'Transitive', etc. The 'Description: hasPlatform' section is also expanded, showing 'Equivalent To', 'SubProperty Of', 'Inverse Of', 'Domains (intersection)', 'Ranges (intersection)', 'Disjoint With', and 'SuperProperty Of (Chain)'. A red arrow points to the 'Domains (intersection)' button, which is currently set to 'Game'. Below the screenshot, the text $\exists hasPlatform.T \sqsubseteq Game$ is displayed in red.

$\exists hasPlatform.T \sqsubseteq Game$

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Object Properties (Domain and Range)

Make sure to have the “Object Properties” tab open
Window → Tabs → Object Properties



Object Properties (Domain and Range)

Make sure to have the “Object Properties” tab open
Window → Tabs → Object Properties

The screenshot shows the Semantic Web editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL: <http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>. The main window has tabs for Active Ontology, Entities, Classes, Object Properties, Individuals by class, OWLViz, and DL Query. The 'Object Properties' tab is active, showing the 'Object property hierarchy: hasPlatform'. The left pane displays the hierarchy: owl:topObjectProperty > hasPlatform > hasDifficulty > hasGenre. The right pane shows the 'Usage: hasPlatform' section, indicating 'Found 1 uses of hasPlatform' and listing 'ObjectProperty: hasPlatform'. A red box highlights the 'Characteristic' and 'Description' sections. The 'Characteristic' section includes checkboxes for Functional, Inverse function, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive. The 'Description' section includes 'Equivalent To', 'SubProperty Of', 'Inverse Of', 'Domains (intersection)' (with 'Game' selected), 'Ranges (intersection)' (with 'Platform' selected), 'Disjoint With', and 'SuperProperty Of (Chain)'. Below the red box, the logical expression $T \sqsubseteq \forall \text{hasPlatform}. \text{Platform}$ is displayed. At the bottom, a footer bar contains the text 'To use the reasoner click Reasoner > Start reasoner' and a checked checkbox for 'Show Inferences'.

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) : (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>)

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLViz x DL Query x

Object property hierarchy: hasPlatform

Annotations Usage

Asserted

owl:topObjectProperty

- hasPlatform
 - hasDifficulty
 - hasGenre

Show: ☒ this ☒ disjoints

Found 1 uses of hasPlatform

- hasPlatform
 - ObjectProperty: hasPlatform

Characteristic

- ☐ Functional
- ☐ Inverse function
- ☐ Transitive
- ☐ Symmetric
- ☐ Asymmetric
- ☐ Reflexive
- ☐ Irreflexive

Description: hasPlatform

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

- Game

Ranges (intersection) +

- Platform

Disjoint With +

SuperProperty Of (Chain) +

$T \sqsubseteq \forall \text{hasPlatform}. \text{Platform}$

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Axioms

Which axioms?

- ▶ only axioms of the following forms
 - ▶ $A \sqsubseteq C$ (necessary condition for A)
 - ▶ $A \equiv C$ (sufficient and necessary condition for A – definition)
- ▶ for each subclass of NamedGame we need to insert axioms expressing something like
 - ▶ Chess can be installed on any platform
 - ▶ League of Legends is an online game
- ▶ DifficultyValuePartition need to be properly defined
(i.e., its values can only be Hard, Normal, or Easy)
- ▶ adding definable classes

$A \sqsubseteq C$ – Example

- ▶ Natural language specification

Chess can be installed on any platform

- ▶ Rephrase the specification using the ontology vocabulary

Chess has platform Windows, has platform MacOSX, and has platform Linux

- ▶ Write it in description logic syntax (optional)

$Chess \sqsubseteq \exists hasPlatform.WindowsPlatform$

$Chess \sqsubseteq \exists hasPlatform.MacOSXPlatform$

$Chess \sqsubseteq \exists hasPlatform.LinuxPlatform$

- ▶ Write it in Manchester syntax (the right-hand side is enough)

hasPlatform some WindowsPlatform

hasPlatform some MacOSXPlatform

hasPlatform some LinuxPlatform

Adding Axioms to the Class “Chess”

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main window is titled 'untitled-ontology-25' and shows a class hierarchy on the left and a description panel on the right.

Class hierarchy: Chess

- owl:Thing
 - Game
 - NamedGame
 - Chess** (highlighted with a red arrow)
 - LeagueOfLegends
 - Sudoku
 - WorldOfWarcraft
 - Platform
 - LinuxPlatform
 - MacOSXPlatform
 - WindowsPlatform
 - Genre
 - MultiPlayer
 - Online
 - Puzzle
 - RolePlayGame
 - SinglePlayer
 - ValuePartition
 - DifficultyValuePartition
 - Easy
 - Hard
 - Normal

Usage: Chess

Show: ☒ this ☒ disjoints ☒ named sub/superclasses

Found 6 uses of **Chess**

- Chess**
 - Class: **Chess**
 - Chess** SubClassOf NamedGame
 - DisjointClasses: **Chess**, LeagueOfLegends, Sudoku, WorldOfWarcraft
- LeagueOfLegends**
 - DisjointClasses: **Chess**, LeagueOfLegends, Sudoku, WorldOfWarcraft
- Sudoku**

Description: Chess

Equivalent To: +

SubClass Of: + (highlighted with a red arrow)
NamedGame

General class axioms: +

SubClass Of (Anonymous Ancestor):

Instances: +

Target for Key: +

Disjoint With: +
Sudoku, WorldOfWarcraft, LeagueOfLegends

Disjoint Union Of: +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Axioms to the Class “Chess”

The screenshot shows the Protégé interface for an ontology named "untitled-ontology-25". The "Class hierarchy" tab is active, displaying a tree structure of classes. A red arrow points to the "Chess" class, which is highlighted in blue. The "Chess" class is a subclass of "NamedGame", which is a subclass of "Game". The "Game" class is a subclass of "owl:Thing". The "Chess" class has three subclasses: "LeagueOf", "Sudoku", and "WorldOfWar". The "Platform" class has three subclasses: "LinuxPlatform", "MacOSPlatform", and "WindowsPlatform". The "Genre" class has five subclasses: "MultiPlayer", "Online", "Puzzle", "RolePlayGame", and "SinglePlayer". The "ValuePartition" class has three subclasses: "Easy", "Hard", and "Normal".

The "Class expression editor" tab is also visible, showing the expression "hasPlatform some WindowsPlatform".

The "Sudoku, WorldOfWarcraft, LeagueOfLegends" class is highlighted in the hierarchy, and the "Disjoint Union Of" button is visible below it.

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Axioms to the Class “Chess”

The screenshot shows the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL: <http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>. The main window is divided into several panes:

- Class hierarchy:** Shows a tree structure of classes. The 'Chess' class is highlighted under the 'NamedGame' class. A red arrow points to the 'Chess' class in this hierarchy.
- Usage:** Shows the usage of the 'Chess' class. It lists 6 uses of 'Chess', including 'Class: Chess', 'Chess SubClassOf NamedGame', and 'DisjointClasses: Chess, LeagueOfLegends, Sudoku, WorldOfWarcraft'.
- Description:** Shows the description of the 'Chess' class. It includes a list of axioms and a table of properties.

The **Description** pane is highlighted with a red border. It contains the following information:

- Equivalent To:** A button with a plus sign (+).
- SubClass Of:** A button with a plus sign (+).
- Axioms:** A list of axioms for the 'Chess' class:
 - hasDifficulty some Normal
 - hasGenre some MultiPlayer
 - hasGenre some SinglePlayer
 - hasPlatform some LinuxPlatform
 - hasPlatform some MacOSXPlatform
 - hasPlatform some WindowsPlatform
 - NamedGame
- General class axioms:** A button with a plus sign (+).
- SubClass Of (Anonymous Ancestor):** A button with a plus sign (+).
- Instances:** A button with a plus sign (+).
- Target for Key:** A button with a plus sign (+).

The table of properties is as follows:

Property	Value
hasDifficulty	some Normal
hasGenre	some MultiPlayer
hasGenre	some SinglePlayer
hasPlatform	some LinuxPlatform
hasPlatform	some MacOSXPlatform
hasPlatform	some WindowsPlatform
NamedGame	

At the bottom of the window, there is a status bar that reads: "To use the reasoner click Reasoner > Start reasoner" and a checkbox labeled "Show Inferences" which is checked.

Improving DifficultyValuePartition Definition

What needs to be done?

- ▶ add $\text{DifficultyValuePartition} \equiv \text{Hard} \sqcup \text{Normal} \sqcup \text{Easy}$

Note that Hard, Normal and Easy are already disjoint

- ▶ add domain and range of hasDifficulty
- ▶ make hasDifficulty functional

Improving DifficultyValuePartition Definition (cont'd)

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) : (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>)

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: DifficultyValuePartition Usage: DifficultyValuePartition

owl:Thing

- Game
 - NamedGame
 - Chess
 - LeagueOfLegends
 - Sudoku
 - WorldOfWarcraft
 - Platform
 - LinuxPlatform
 - MacOSXPlatform
 - WindowsPlatform
 - Genre
 - MultiPlayer
 - Online
 - Puzzle
 - RolePlayGame
 - SinglePlayer
 - ValuePartition
 - DifficultyValuePartition
 - Easy
 - Hard
 - Normal

Found 6 uses of DifficultyValuePartition

- DifficultyValuePartition
 - DifficultyValuePartition SubClassOf ValuePartition
 - Class: DifficultyValuePartition
- Easy
 - Easy SubClassOf DifficultyValuePartition
- Hard
 - Hard SubClassOf DifficultyValuePartition

Description: DifficultyValuePartition

Equivalent To +

SubClass Of +

- ValuePartition

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Disjoint Union Of +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Improving DifficultyValuePartition Definition (cont'd)

The screenshot displays the Protégé ontology editor interface. The main window shows the class hierarchy for 'untitled-ontology-25'. The 'DifficultyValuePartition' class is highlighted in the hierarchy, and a red arrow points to it. The 'Object restriction creator' dialog is open, showing the text 'Hard or Normal or Easy'. The dialog has tabs for 'Object restriction creator', 'Class expression editor', 'Class hierarchy', and 'Data restriction creator'. The 'Object restriction creator' tab is active, and the text 'Hard or Normal or Easy' is entered. The dialog also has 'OK' and 'Cancel' buttons. Below the dialog, there is a red box containing the text 'Disjoint Union Of' followed by a plus sign icon. At the bottom of the window, there is a status bar with the text 'To use the reasoner click Reasoner > Start reasoner' and a checked checkbox for 'Show Inferences'.

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) : [<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (<http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25>) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWL Viz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: DifficultyValuePartition

owl:Thing

- Game
 - NamedGame
 - Chess
 - LeagueOfLegends
 - Sudoku
 - WorldOfWarcraft
 - Platform
 - LinuxPlatform
 - MacOSPlatform
 - WindowsPlatform
 - Genre
 - MultiPlayer
 - Online
 - Puzzle
 - RolePlayingGame
 - SinglePlayer
 - ValuePartition
 - DifficultyValuePartition
 - Easy
 - Hard
 - Normal

Object restriction creator Class expression editor Class hierarchy Data restriction creator

Hard or Normal or Easy

OK Cancel

Disjoint Union Of +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Improving DifficultyValuePartition Definition (cont'd)

The screenshot shows the Protégé interface with the 'untitled-ontology-25' loaded. The left pane displays the class hierarchy, where 'DifficultyValuePartition' is selected under 'ValuePartition'. The right pane shows the 'Usage' tab for 'DifficultyValuePartition', listing 7 uses. A red box highlights the 'Description' tab for 'DifficultyValuePartition', which contains the following information:

- Equivalent To: **Hard or Normal or Easy**
- SubClass Of: **ValuePartition**
- General class axioms: (empty)
- SubClass Of (Anonymous Ancestor): (empty)
- Instances: (empty)
- Target for Key: (empty)
- Disjoint With: (empty)
- Disjoint Union Of: (empty)

At the bottom of the interface, a status bar indicates: 'To use the reasoner click Reasoner > Start reasoner' and a checked checkbox for 'Show Inferences'.

Improving DifficultyValuePartition Definition (cont'd)

The screenshot shows the Protégé ontology editor interface. The top bar indicates the ontology is 'untitled-ontology-25'. The left pane shows the 'Object property hierarchy: hasDifficulty' with a tree structure: owl:topObjectProperty, hasPlatform, hasDifficulty (highlighted with a red arrow), and hasGenre. The right pane shows the 'Usage: hasDifficulty' section, which includes a 'Show: this disjoints' section and a 'Found 4 uses of hasDifficulty' section. The 'hasDifficulty' property is expanded, showing its characteristics: 'hasDifficulty Range DifficultyValuePartition', 'ObjectProperty: hasDifficulty', and 'hasDifficulty Domain Game'. Below this, the 'Characteristics' section is expanded, showing a list of checkboxes: 'Functional' (checked, highlighted with a red arrow), 'Inverse function', 'Transitive', 'Symmetric', 'Asymmetric', 'Reflexive', and 'Irreflexive'. The 'Description: hasDifficulty' section is also visible, showing 'Equivalent To', 'SubProperty Of', 'Inverse Of', 'Domains (intersection)' (with 'Game' as a domain), 'Ranges (intersection)' (with 'DifficultyValuePartition' as a range), 'Disjoint With', and 'SuperProperty Of (Chain)'. At the bottom, a status bar indicates 'To use the reasoner click Reasoner > Start reasoner' and 'Show Inferences' is checked.

untitled-ontology-25 (http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25) : [http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (http://www.semanticweb.org/papacchf/ontologies/2016/9/untitled-ontology-25) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Object property hierarchy: hasDifficulty

Annotations Usage

Usage: hasDifficulty

Show: ☒ this ☒ disjoints

Found 4 uses of hasDifficulty

Chess

Chess SubClassOf hasDifficulty some Normal

hasDifficulty

hasDifficulty Range DifficultyValuePartition

ObjectProperty: hasDifficulty

hasDifficulty Domain Game

Characteristics

☒ Functional

☐ Inverse function

☐ Transitive

☐ Symmetric

☐ Asymmetric

☐ Reflexive

☐ Irreflexive

Description: hasDifficulty

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Game

Ranges (intersection) +

DifficultyValuePartition

Disjoint With +

SuperProperty Of (Chain) +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Adding Definable Class “MultiPlayerGame”

The screenshot shows the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main window displays the ontology "untitled-ontology-25" with a search bar and tabs for Active Ontology, Entities, Classes, Object Properties, Individuals by class, OWLviz, and DL Query.

The left pane shows the "Class hierarchy: MultiPlayerGame" with a tree structure. A red arrow points to the "MultiPlayerGame" class under the "Game" class. The tree structure is as follows:

- owl:Thing
 - Game
 - MultiPlayerGame**
 - NamedGame
 - Chess
 - LeagueOfLegends
 - Sudoku
 - WorldOfWarcraft
 - Platform
 - LinuxPlatform
 - MacOSXPlatform
 - WindowsPlatform
 - Genre
 - MultiPlayer
 - Online
 - Puzzle
 - RolePlayGame
 - SinglePlayer
 - ValuePartition
 - DifficultyValuePartition
 - Easy
 - Hard
 - Normal

The right pane shows the "Usage: MultiPlayerGame" tab, displaying "Found 2 uses of MultiPlayerGame". The first use is "MultiPlayerGame" (Class: MultiPlayerGame) and the second use is "MultiPlayerGame SubClassOf Game".

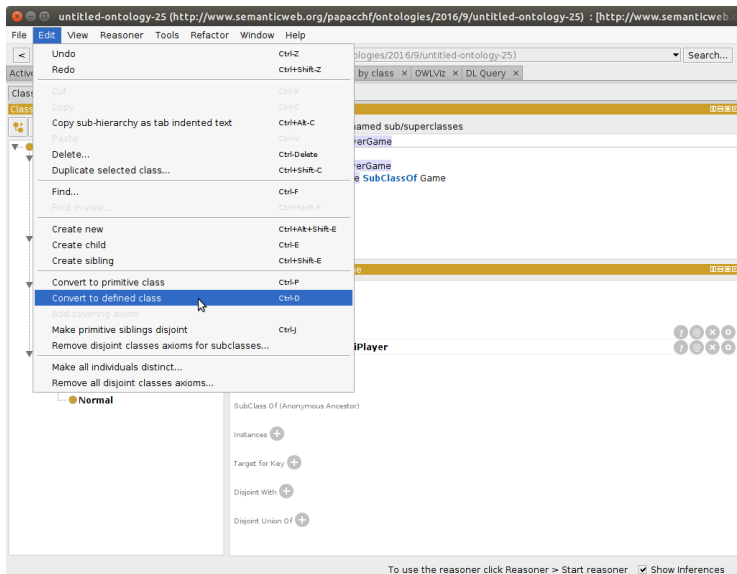
The bottom pane shows the "Description: MultiPlayerGame" tab, which is highlighted with a red box. It contains the following information:

- Equivalent To: +
- SubClass Of: +
 - Game
 - hasGenre some MultiPlayer
- General class axioms: +
- SubClass Of (Anonymous Ancestor):
- Instances: +
- Target for Key: +
- Disjoint With: +
- Disjoint Union Of: +

A red arrow points to the "hasGenre some MultiPlayer" axiom in the "SubClass Of" list.

At the bottom of the window, a status bar indicates: "To use the reasoner click Reasoner > Start reasoner" and a checkbox for "Show Inferences" is checked.

Adding Definable Class “MultiPlayerGame”



Adding Definable Class “MultiPlayerGame”

The screenshot shows the Protégé ontology editor interface. The left pane displays the class hierarchy, with **MultiPlayerGame** highlighted under the **Game** class. A red arrow points to this class in the hierarchy. The right pane shows the **Usage: MultiPlayerGame** tab, which displays the class definition: **MultiPlayerGame** *EquivalentTo* **Game** *and* (*hasGenre* *some* **MultiPlayer**). A red arrow points to this definition. Below the definition, the **Description: MultiPlayerGame** tab is visible, showing the same definition. A red box highlights this tab, and a red arrow points to the definition. The bottom status bar indicates: To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences.

untitled-ontology-25 (<http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>) : <http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (<http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: MultiPlayerGame Usage: MultiPlayerGame

Show: ☒ this ☒ disjoints ☒ named sub/superclasses

Found 2 uses of MultiPlayerGame

MultiPlayerGame

- Class: MultiPlayerGame
- MultiPlayerGame *EquivalentTo* Game *and* (*hasGenre* *some* MultiPlayer)

MultiPlayerGame \equiv *Game* \sqcap \exists *hasGenre*. *MultiPlayer*

Description: MultiPlayerGame

Equivalent To: **Game** *and* (*hasGenre* *some* **MultiPlayer**)

SubClass Of

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

Disjoint Union Of

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Reasoning

Protégé can be used for reasoning tasks such as classification

- ▶ configure the reasoner

Reasoner → Configure... (for this tutorial, check everything under Class inferences and Object property inferences)

- ▶ select a reasoner

for example, Reasoner → HermiT (other reasoners can be added, which one to use depends on several factors such as the expressivity of the ontology)

- ▶ finally, Reasoner → Start reasoner

Reasoning Example

untitled-ontology-25 (<http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>) : (<http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>)

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-25 (<http://www.semanticweb.org/papachf/ontologies/2016/9/untitled-ontology-25>) Search...

Active Ontology x Entities x Classes x Object Properties x Individuals by class x OWLviz x DL Query x

Class hierarchy Class hierarchy (inferred) Annotations Usage

Class hierarchy: Game Usage: Game

Asserted

Show: ☒ this ☒ disjoints ☒ named sub/superclasses

Found 20 uses of Game

EasyGame

- EasyGame **EquivalentTo** Game and (hasDifficulty **some** Easy)

Game

- Game **SubClassOf** owl:Thing
- Class: Game
- DisjointClasses:** Game, Genre, Platform, ValuePartition

Description: Game

Equivalent To +

SubClass Of +

- owl:Thing

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

- ValuePartition, Genre, Platform

Disjoint Union Of +

To use the reasoner click Reasoner > Start reasoner ☒ Show Inferences

Reasoning Example

The screenshot displays a Semantic Web browser interface for an ontology named 'untitled-ontology-25'. The interface is divided into several panes:

- Class hierarchy: Chess**: A tree view on the left showing the hierarchy of classes. The 'Chess' class is highlighted under 'MultiPlatformGame', which is under 'LinuxGame', which is under 'Game', which is under 'owl:Thing'. A red arrow points from the 'Inferred' tab to this hierarchy.
- Usage: Chess**: A pane on the right showing the usage of the 'Chess' class. It lists 12 uses of 'Chess', including 'Chess SubClassOf hasDifficulty some Normal', 'Chess SubClassOf hasGenre some MultiPlayer', 'Chess SubClassOf hasGenre some SinglePlayer', and 'Chess SubClassOf hasGenre some MultiPlayer'. A red arrow points from the 'Inferred' tab to this pane.
- Description: Chess**: A pane on the right showing the description of the 'Chess' class. It lists the following properties and values:
 - hasDifficulty some Normal
 - hasGenre some MultiPlayer
 - hasGenre some SinglePlayer
 - hasPlatform some LinuxPlatform
 - hasPlatform some MacOSXPlatform
 - hasPlatform some WindowsPlatform
 - NamedGame
 - MultiPlatformGame
 - MultiPlayerGame
 - NormalGame
 - SinglePlayerGame
- General class axioms**: A pane at the bottom showing the general class axioms for 'Chess'. It lists the following axioms:
 - Game and (hasGenre some MultiPlayer)
 - Game and (hasDifficulty some Normal)
 - Game and (hasPlatform some LinuxPlatform) and (hasPlatform some MacOSXPlatform) and (hasPlatform some WindowsPlatform)
 - Game and (hasGenre some SinglePlayer)

The 'Inferred' tab is selected, and the 'Reasoner active' checkbox is checked at the bottom right.

Reasoning – Visually (Asserted)



Reasoning – Visually (Inferred)



Resources Summary

Download from <http://protege.stanford.edu/>

Guide, Slides, Wiki

- ▶ Official Tutorial (for version 4, small differences here and there)
http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf
- ▶ Wiki
http://protegewiki.stanford.edu/wiki/Main_Page
- ▶ extended version of this presentation at the module site
<http://cgi.csc.liv.ac.uk/~frank/teaching/comp08/comp321.html>

Game Ontology <http://cgi.csc.liv.ac.uk/~frank/teaching/comp08/videogame.owl>