

Дипломна робота на тему: Алгоритми логічного виведення фактів в OWL-онтологіях

Виконала

Студентка IV курсу
Слухай Я.О.

Науковий керівник

Булах Б.В.,
к.т.н., ст. викладач

Актуальність теми дослідження

- ▶ Онтології грають важливу роль в організації обробки знань на основі Web, а також для їх спільного використання.
- ▶ Онтології часто використовуються для формальної специфікації понять і відношень, які характеризують певну область знань (термінологічні онтології).
- ▶ Особливу увагу привертає можливість виведення нових логічних фактів, що забезпечується модулями логічного виведення фактів з онтологій (reasoners).

Мета роботи

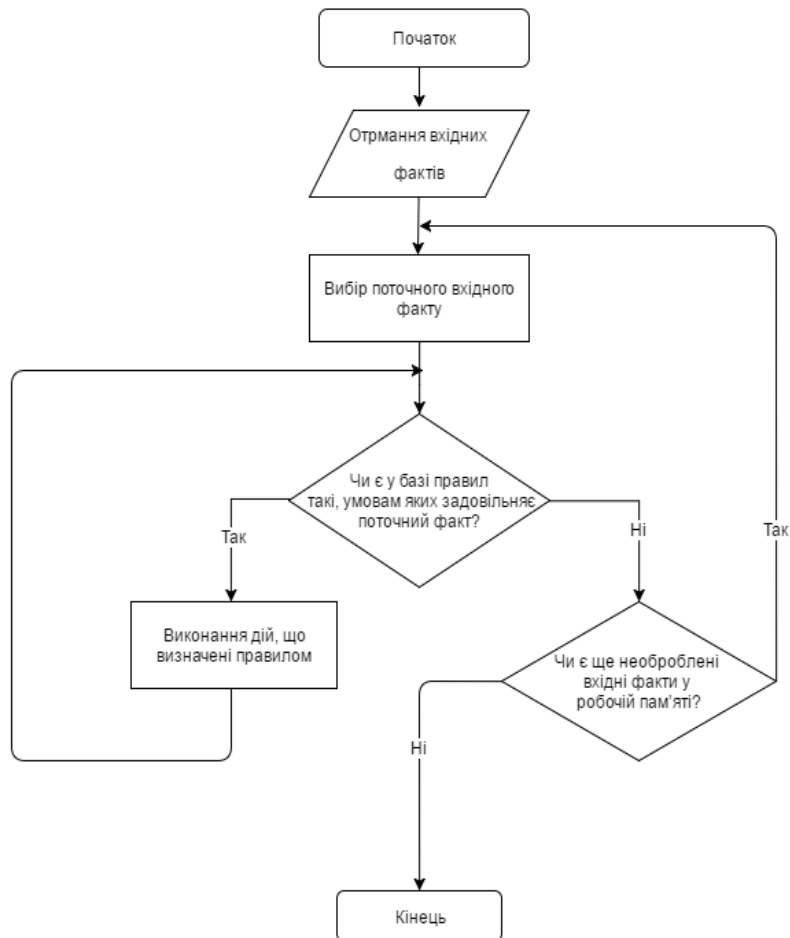
- ▶ Дана робота присвячена огляду алгоритмів, які найчастіше лягають в основу побудови ризонерів, їх аналізу та порівнянню швидкодії та точності виконання основних функцій ризонерів у контексті різних за вибраними показниками онтологій.
- ▶ Метою дипломного проекту є розробка програмного додатку, що аналізує швидкодію та якість роботи основних функцій вибраних семантичних ризонерів (Structural reasoner, Pellet, Hermit), виконуючи їх над вибраною користувачем онтологією.

Семантичний ризонер

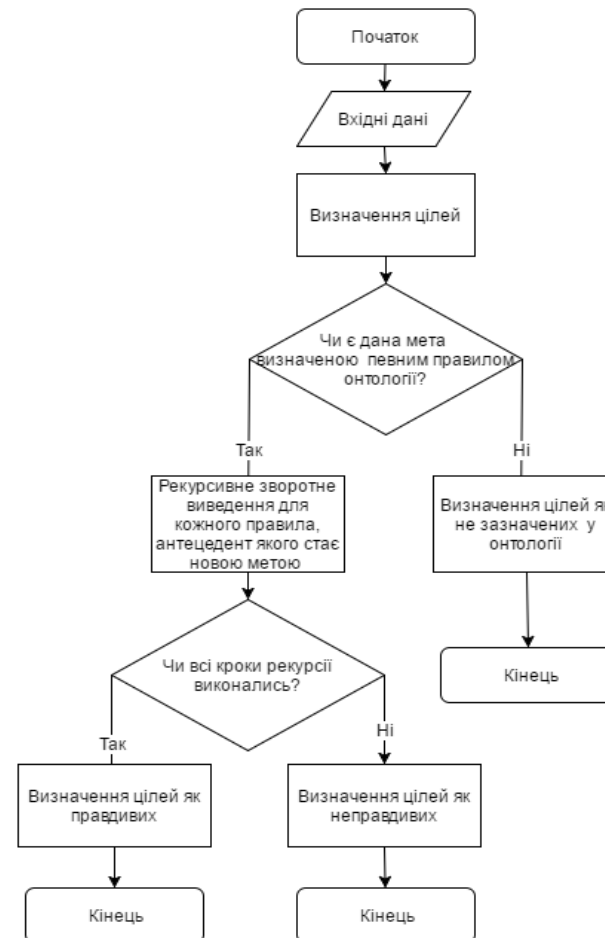
- ▶ Семантичний ризонер - частина програмного забезпечення, що здатна виводити логічні висновки з набору вибраних фактів та аксіом, а також надає можливість автоматичної підтримки таких завдань:
 - ▶ Проводити класифікацію і виводити ієрархію класів.
 - ▶ Перевіряти консистентність онтології.
 - ▶ Визначати тип індивіда - належність до певного класу.
 - ▶ Визначати класи, що не перетинаються з заданим класом.
 - ▶ Визначати підкласи вибраних класів.

Алгоритми, що найчастіше використовуються при розробці ризонерів

► Прямого виведення

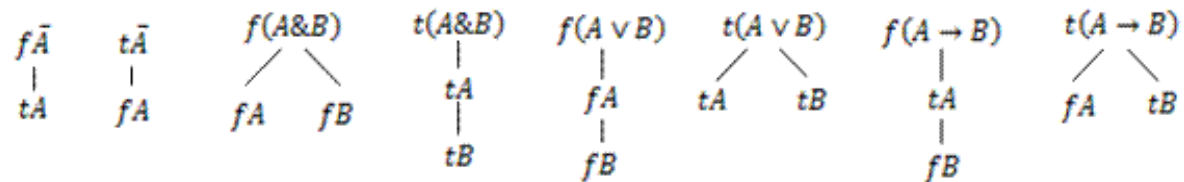


► Зворотного виведення



Алгоритми, що найчастіше використовуються при розробці різонерів

- ▶ Метод семантичних таблиць - це формальна роздільна процедура для формул логіки висловлювань і логіки предикатів, що дозволяє чисто синтаксичними засобами вирішувати семантичні проблеми формалізованих обчислень.



- ▶ Алгоритм побудови гіпертаблиць базується на особливостях побудови аналітичних таблиць, але з використанням переваг головних ідей алгоритму гіпер-резолюції.

Порівняльний аналіз існуючих різонерів

Різонер Критерій	Pellet	RACER	FACT++	Snorocket	SWRLIQ	Hermit	CEL	TrOWL	ELK	
Алгоритм	Семантичних таблиць	Семантичних таблиць	Семантичних таблиць	Правил виведення	SWRL правил	Гіпертабличного числення	Правил виведення	Правил виведення	Правил виведення	
Стійкість	Так	Так	Так	Так	Так	Так	Так	Так	Так	
Повність	Так	Так	Так	Так	Ні	Так	Так	Так	Так	
Виразність	SROIQ (D)	SHIQ	SROIQ (D)	EL+	-	SROIQ (D)	EL+	SHIQ	EL	
Нативний профіль	DL, EL	DL	DL	EL	-	DL	EL	DL, EL	EL	
Інкрементальна класифікація	Додавання	Так	Ні	Ні	Так	Ні	Ні	Так	Ні	Так
	Видалення	Так	Ні	Ні	Ні	Ні	Ні	Ні	Ні	Так
Підтримка правил	Так (SWRL)	Так (SWRL)	Ні	Ні	Так (SWRL)	Так (SWRL)	Ні	Ні	Ні	Так
Платформи	Всі	Всі	Всі	Всі	Всі	Всі	Linux	Всі	Всі	
Обґрунтування	Так	Так	Ні	Ні	Так	Ні	Так	Ні	Ні	
Виведення логічних висновків щодо індивідів	Так	Так	Так	Ні	Так	Так	Так	Так	Ні	
Підтримка OWL API	Так	Так	Так	Так	Ні	Так	Так	Так	Так	
Підтримка OWL Link API	Так	Так	Так	Ні	Ні	Так	Так	Ні	Ні	
Підтримка Protégé	Так	Так	Так	Так	Так	Так	Так	Так	Так	
Підтримка NeOn	Так	Ні	Ні	Ні	Ні	Так	Ні	Ні	Ні	
Ліцензія	DULI: AGPL	Власна	GLGPL	Власна	Так/Ні	GLGPL	Apache License 2.0	DULI: AGPL	Apache License 2.0	
Підтримка Jena	Так	Ні	Ні	Ні	Ні	Ні	Ні	Так	Ні	
Мова написання	Java	Lisp	C++	Java	Prolog	Java	Lisp	Java	Java	
Доступ	Відкритий	Комерційний	Відкритий	Комерційний	Комерційний	Відкритий	Відкритий	Комерційний	Відкритий	

Додаток для аналізу ризонерів

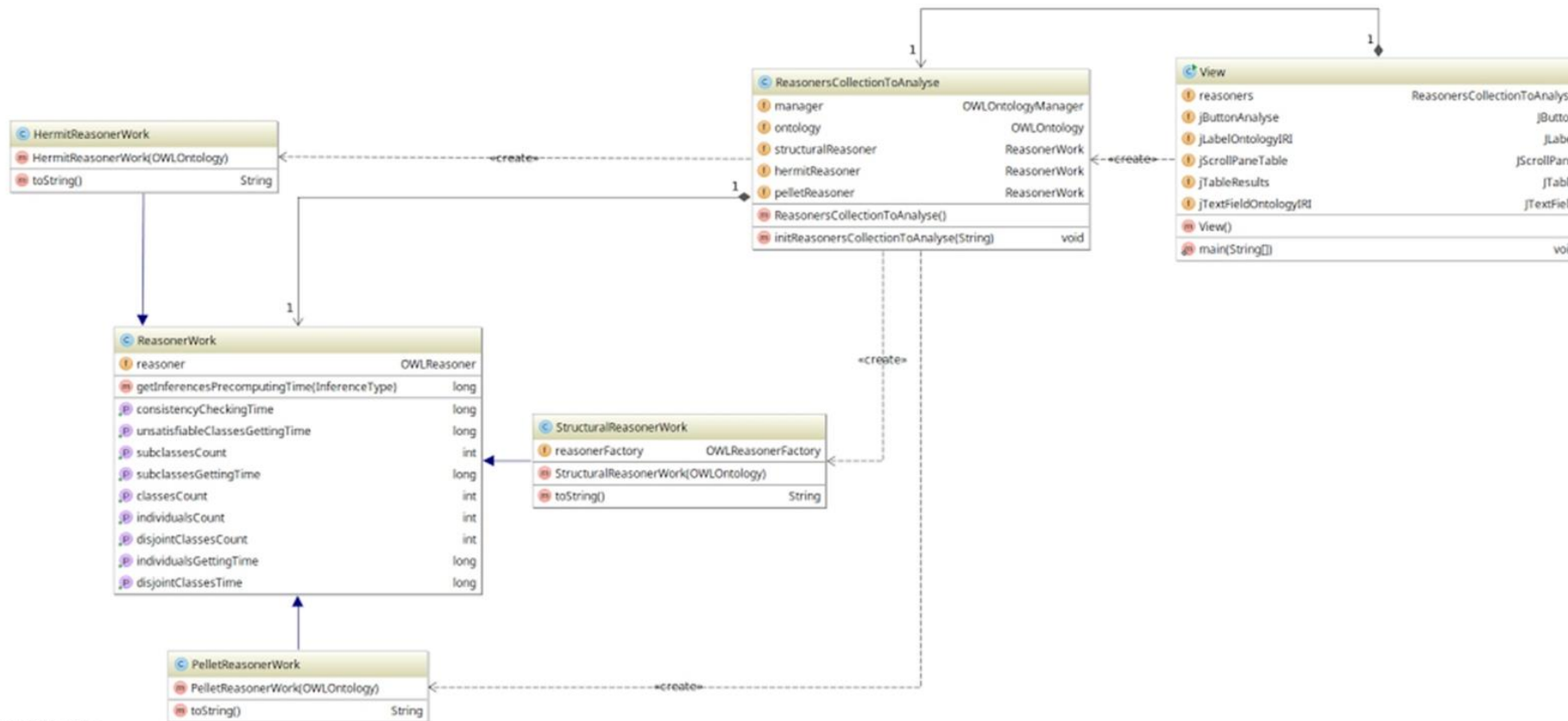
- У рамках дипломного проекту було вирішено проаналізувати швидкодію та якість виконання головних функцій трьох ризонерів, які демонструють роботу розглянутих алгоритмів.

Ризонер	Алгоритм
Structural Reasoner	Прямого та зворотнього виводу
Pellet	Семантичних таблиць
Hermit	Гіпертаблиць

Критерії аналізу ризонерів

- ▶ Час логічного виведення класової ієрархії.
- ▶ Час логічного виведення ієрархії властивостей даних.
- ▶ Час логічного виведення ієрархії властивостей об'єктів.
- ▶ Кількість всіх зв'язків «клас - підклас» онтології.
- ▶ Час отримання всіх зв'язків «клас - підклас» онтології.
- ▶ Кількість всіх зв'язків «клас - непересічний клас» онтології.
- ▶ Час отримання всіх зв'язків «клас - непересічний клас» онтології.
- ▶ Кількість всіх зв'язків «індивідуал - клас» онтології.
- ▶ Час отримання всіх зв'язків «індивідуал - клас» онтології.
- ▶ Час перевірки онтології на консистентність.

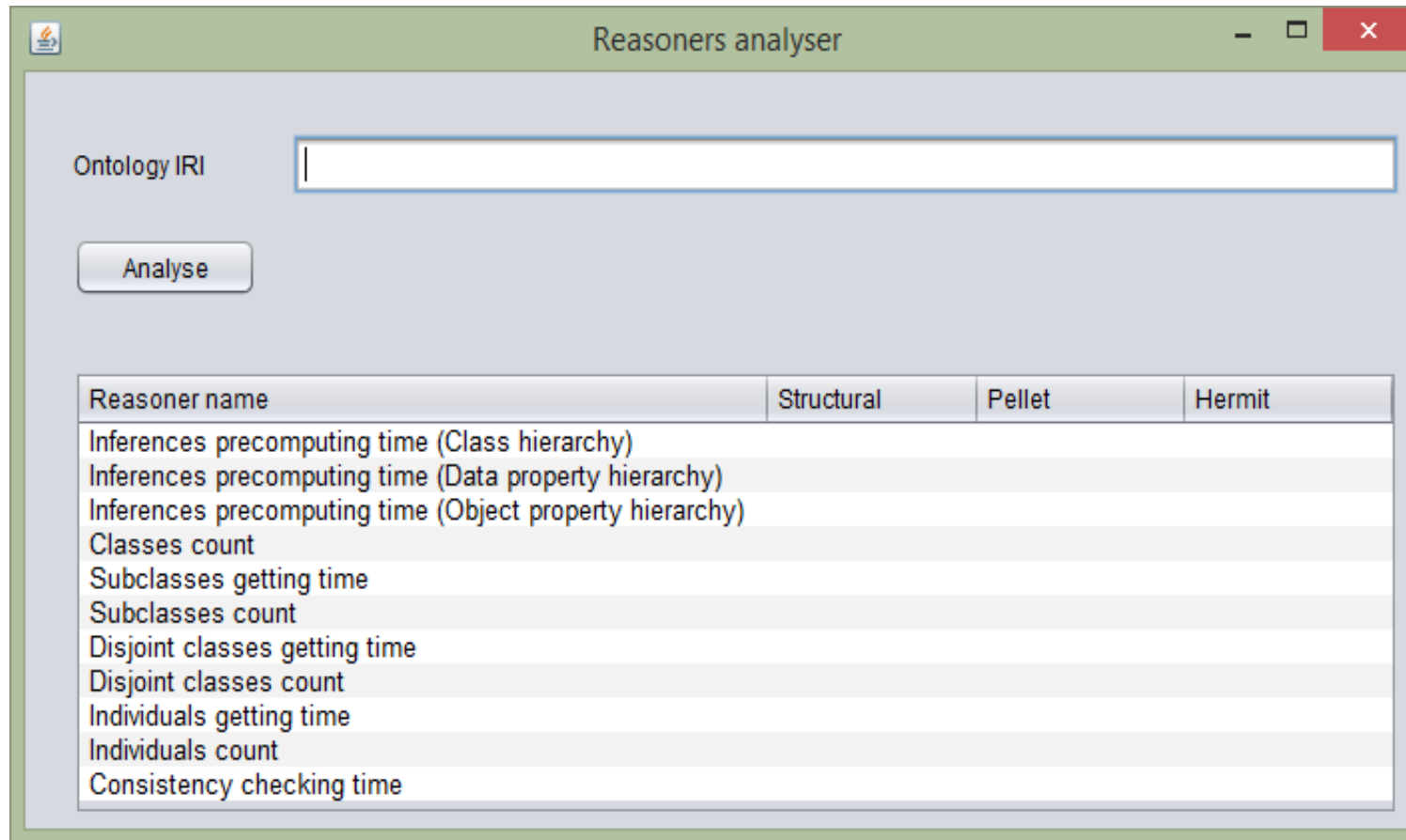
UML-діаграма класів розробленого додатку



Використані технології

- ▶ OWL API - прикладний програмний інтерфейс для створення, аналізу і обробки OWL онтологій. Включає в себе інтерфейс OWLReasoner для доступу до OWL ризонерів, який імплементують всі вибрані для дослідження ризонери.
- ▶ Java Maven - це засіб автоматизації роботи з програмними проектами, який спочатку використовувався для Java проектів. Використовується для управління (management) та складання (build) програм.
- ▶ Java Swing - платформи-незалежна бібліотека, що означає, що програму з використанням Swing можна запустити на всіх платформах, які підтримують JVM.

Графічний інтерфейс розробленого додатку



Тестові онтології

	IRI онтології	Тип онтології
Тестова онтологія №1	http://ontology.dumontierlab.com/molecule-complex	Термінологічна, одномовна, безекземплярна онтологія предметної області (молекулярна хімія)
Тестова онтологія №2	http://ontology.dumontierlab.com/unit-individuals	Термінологічна, одномовна, екземплярна онтологія предметної області (фізика)
Тестова онтологія №3	http://ontology.dumontierlab.com/time-interval-primitive	Термінологічна, одномовна, безекземплярна онтологія предметної області (одиниці виміру інтервалів часу)
Тестова онтологія №4	http://owl.man.ac.uk/2006/07/sssw/people.owl	«Справжня», одномовна, екземплярна онтологія для опису навколишнього світу
Тестова онтологія №5	http://www.biopax.org/release/biopax-level1.owl	Термінологічна, одномовна, безекземплярна онтологія предметної області (біохімія)
Тестова онтологія №6	http://ontology.dumontierlab.com/physics-complex-1.0.owl	Термінологічна, одномовна, безекземплярна онтологія предметної області (фізика)

Висновки

- ▶ Отже, бачимо, що оптимальним варіантом для виведення логічних фактів на прикладі вибраних тестових онтологій є Pellet - ризонер, що реалізовує метод семантичних таблиць. Але перевірку на консистентність системи варто проводити, використовуючи Hermit. Structural reasoner виявився менш ефективним у виведенні зв'язків «індивідуал - клас», «клас - підклас», «клас - непересічний клас», адже у багатьох випадках кількість таких зв'язків була суттєво меншою за кількість зв'язків, яку вивели Pellet та Hermit.
- ▶ Виконано порівняльний аналіз характеристик існуючих ризонерів, досліджено принципи їх роботи та сформульовано вимоги до програмного забезпечення, що тестуватиме їх ефективність та реалізовано таке програмне забезпечення на мові Java.

Висновки

- ▶ Розроблений додаток дає змогу користувачеві вибрати оптимальний ризонер для виконання кожної з функцій над конкретною онтологією в залежності від її цілей, архітектури, кількості явних і неявних фактів т.д. Це забезпечує кращу швидкодію роботи з базою знань.
- ▶ Розроблене програмне забезпечення дозволить тестувати нові ризонери, порівнюючи їх з тими, що вже існують, на наборі тестових онтологій. Тобто це корисний інструмент з набору як розробників онтологій так і програмістів-розробників ризонерів.