Технічні науки

УДК 004.62

Орлова Марія Миколаївна
кандидат технічних наук, доцент
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Орлова Мария Николаевна
кандидат технических наук, доцент
Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»
Orlova Mariia
Candidate of Technical Sciences, Associate Professor
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Дзицюк Євгенія Володимирівна
студент
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
Дзыцюк Евгения Владимировна
студент
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»
Dzytsiuk Yevheniia
Student of the
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

МОДИФІКАЦІЯ АЛГОРИТМУ ШЕПЛІ ДЛЯ ОБЧИСЛЕННЯ НЕВІДПОВІДНОСТІ ОНТОЛОГІЙ

МОДИФИКАЦИЯ АЛГОРИТМА ШЕПЛИ ДЛЯ ВЫЧИСЛЕНИЯ НЕСООТВЕТСТВИЯ ОНТОЛОГИЙ
MODIFICTION OF SHAPLEY ALGORITHMS FOR THE CALCULATION OF ONTOLOGIES INCONSISTENCY

Анотація. *В даній роботі описані існуючі підходи до обчислення міри неконсистентності онтологій. Запропоновано поєднання алгоритмів MIV та Шеплі та визначені переваги такого підходу.*

Ключові слова: *онтологія, консистентність, алгоритм Шеплі.*

Аннотация. *В данной работе описаны существующие подходы вычисления меры неконсистентности онтологий. Предложено объединение алгоритмов MIV и Шепли и определены преимущества такого подхода*

Ключевые слова: *онтология, консистентность, алгоритм Шепли.*

Summary. *This paper describes a study of existing approaches to the technology of measuring inconsistencies in ontology. Combining MI and Shapley algorithms was proposed and the advantages of this method were highlighted.*

Key words: *ontology, consistency, Shapley algorithm.*

Вступ. Необхідність розробки надійних, але принципових логічних методів для аналізу інформації все частіше визнається невід`ємною частиною науки про обробку даних. Онтології відіграють важливу роль в даній сфері. Онтології використовуються в різноманітних додатках, які розширюються кожного дня. Відповідно збільшується і розмір онтологій, що веде до появи неконсистентностей. На сьогодні існує декілька методів обробки неконсистентних онтологій, які засновані на наступних сценаріях

1. Існує більше одного набору аксіом, які створюють

неконсистентність в онтології і не залежать від одного.

Припустимо, K' і K'' - дві несумісні підмножини онтології O. Якщо O має неперетинаючі джерела невідповідностей, це означає, що $K' \cap K'' = \varnothing$. Невідповідність може мати місце на різних рівнях: на рівні однієї аксіоми і на рівні множин аксіом. Наприклад, нехай

$K' = \{C \cup \neg C \in C \cap \neg C\}$ та $K'' = \{T \in \exists R.B\ ,\ T \in R.\neg B\}$. Аксіома в K' є неконсистентною, тоді як обидві з двох аксіоми в K" несуть є причиною цього протиріччя.

2. Існує більше одного набору аксіом, необхідних для створення неконсистентності в онтології, і вони поєднуються з одним одним.

Онтологія O, що має перекриваючі джерела неконсистентностей, означає, що $K' \cap K'' \neq \varnothing$. Якщо два набори несумісних аксіом перетинаються, це означає, що певні аксіоми роблять більший внесок у неконсистентність. Найімовірніше, що видалення таких аксіом з O призведе до вирішення інших невідповідностей.

У даній роботі пропонується кількісна міра неконсистентностей у онтологіях, на основі алгоритмів Шеплі та MIV.

Мета даної роботи - використання раніше відомих стратегій теорії ігор в контексті онтологій в поєднанні з алгоритмом мінімальних підмножин і представлення міри систематичної оцінки неконсистентності в онтологіях. Цей підхід є незалежним від конкретного виду мови онтології або конкретної системою обробки онтології.

2. Основні поняття

У цьому розділі спочатку розглянемо деякі основні поняття та терміни, пов'язані з мовами онтології в семантичному Web та їх зв'язок з описовими логіками (DL). Також визначимо поняття неконсистентності в онтології.

2.1 Онтології в семантичному Web

Мова веб-онтології (OWL) рекомендована як стандарт мови web-онтологій Консорціумом Всесвітнього Інтернету [6]. Це мова для обміну та обґрунтування інформації в мережі Інтернеті.

OWL - це розширення структури опису ресурсів (RDF) і перегляд web-онтологічної мови DAML + OIL.

OWL являє собою представлення домену предметної області і визначає ієрархії класів та властивостей. Онтологія OWL складається з аксіом і фактів. Аксіоми визначають інтенсифікацію знання шляхом побудови зв'язків між класами та властивостями. Факти описують розширення знань про індивіди (вирази).

2.2 Описові логіки

Описова логіка - це сімейство концепцій формального представлення знань. Вона представляє собою знання про домен насамперед визначаючи відповідні поняття області. Ці поняття використовуються для визначення властивостей об'єктів у домені. Як правило, мова DL має дві частини: термінологію (*TBox*) і твердження (*ABox*). TBox включає в себе інтенсифікаційні знання у формі аксіом, тоді як ABox містить розширення знань, яке є специфічним для елементів у домені, індивідів.

TBox разом з ABox називається базою знань в DL. В TBox, основні описи - атомарні поняття, позначені універсальними предикатами, і атомарні ролі, позначені бінарними предикатами для вираження зв'язків між індивідами. Поняття можуть бути побудовані на атомарних концепціях за допомогою перетину, об'єднання, заперечення, обмеження. Аксіоми виражають як поняття так і ролі пов'язані один з одним. Як правило, це твердження форми $C \in D$: "Концепція C підпадає під поняття D"; або $C \equiv D$, що означає $C \in D$ та $D \in C$. де C та D є концентами описів. ABox - це набір тверджень $C(a)$ та $R(a, b)$, де R - роль, а a, b - індивіди.

Інтерпретація I визначає формальну семантику понять, ролей та

індивідів. Вона складається з непустої множини Δ^I, яку називають доменом. Функція інтерпретації I ставить у відповідність кожен концепт A підмножині A^I з множини Δ^I та кожну атомарну роль R до бінарного відношення $R^I \epsilon \Delta^I x \Delta^I$. До того ж I переводить кожне ім'я індивіда a в елемент $a^I \epsilon \Delta^I$. В інтерпретації I $C \epsilon D$, якщо $C^I \epsilon D^I$; $C \equiv D$, якщо $C^I \equiv D^I$; $C(a)$, якщо $a^I \epsilon C^I$; $R(a,b)$, якщо $(a^I,b^I) \epsilon R^I$.

Основні властивості концептів в Tbox: здійсненність, категоризація, еквівалентність і диз'юнкція. Концепція C в Tbox T називається здійсненною по відношенню до T, якщо існує непуста модель T (інтерпретація I, яка відповідає аксіомам T) C^I. Інші три властивості можуть бути зведені до (не)здійсненності. Ще одним важливим фактором обробки Tbox є перевірка його консистентності, тобто перевірка існування моделі T. Обробка Abox включає перевірку екземплярів, реалізацію, і вибірку даних. Перевірка екземпляра перевіряє, чи конкретний індивід є екземпляром визначеної концепції. Реалізація знаходить найбільш конкретну концепцію, екземпляром якої є індивід. Вирка даних знаходить індивіди у базі знань, які є екземплярами даної концепції. ABox A є консистентним відносно до TBox T, якщо існує інтерпретація, яка є моделлю як A, так і T (тобто інтерпретація I, яка одночасно задовольняє аксіомам T і твердженням A). Подібно до властивостей TBox, властивості ABox також можна звести до проблеми узгодженості ABox. В даній роботі термін "консистентність" використовується для послання на проблеми узгодженості і в Tbox, і в Abox.

Таким чином, концепт в DL називається класом в OWL. Роль в DL - це властивість у OWL. Терміни "аксіома" та "індивід" мають однакове значення в DL і OWL. OWL DL частково заснований на DL SHOIN (D), який включає в себе спеціальні конструктори, такі як транзитивні властивості, зворотні властивості та властивості типу даних, а також його підмножину OWL Lite, яка базується на менш виразному DL SHIF (D).

Враховуючи те, що зв'язок між OWL і DL є тісним, в даній роботі будемо вважати їх синонімами, приклади будуть наведені в синтаксисі DL.

Вільна формула бази знань K - це формула K, що не належить до жодної мінімальної неконсистентної підмножини K. Це означає, що для ця формули не має конфліктів з базою.

3. **Вимірювання неконсистентності**

У цьому розділі вивчаються методи вимірювання невідповідностей у базах знань DL. Ідея полягає в тому, щоб спочатку визначити значення невідповідності, а потім прийняти його як характеристичну функцію для обчислення значення Шеплі.

Наступний приклад будемо використовувати як робочий. У прикладі A, B, C, D, A1 - A6 позначають концепти, a та b — індивіди.

1. $A1 \sqsubseteq A2 \sqcap \neg A \sqcap A3$
2. $A2 \sqsubseteq A \sqcap A4$
3. $3 \sqsubseteq A5 \sqcap A4$
4. $A4 \sqsubseteq C \sqcap \forall S.B$
5. $A5 \sqsubseteq \exists S.\neg B$
6. $D \sqcup \neg D \sqsubseteq D \sqcap \neg D$
7. A1(a)
8. A3(b)
9. $A6 \equiv D$

Повна DL-обробка, така як FaCT ++ [8], RACER [9], повідомляє, що ця база знань є непослідовною. Однак вони не дають інформацію про те, що, наприклад, існує чотири несумісні підмножини (3, 4, 5 та 8; 1, 2 і 7; 1, 3, 4, 5 і 7; 6) в цій базі знань і , що одна аксіома (аксіома 6) є по суті неконсистентною.

Визначення 1. Враховуючи набір аксіом і тверджень в базі знань K, характеристична функція v: $2^K \rightarrow R$ присвоює значення кожній підмножині K', де $K' \sqsubseteq K$.

Прикладом характеристичної функції є значення невідповідності, яке присвоює 1 до набору аксіом, якщо він неконсистентний, а 0 - до набору, якщо він є послідовним.

Визначення 2. Для набору аксіом і тверджень K' належить K,

неконсистентне значення *K'* визначається як:

I_d *(K')* = { 0, якщо *K'* консистентне або пусте; 1 — в іншому випадку}.

Приклад 1. Деякі значення невідповідності робочого прикладу є наступними. Зображуються лише деякі них, з коефіцієнтом невідповідності 1, і деякі послідовні (коефіцієнт невідповідності 0).

I_d ({1}) = 0 I_d ({1, 2}) = 0

I_d ({3,4,5,8}) = 1 I_d ({1,3,4,5,7}) = 1

I_d ({2}) = 0 I_d ({3,4,5}) = 0

I_d ({3,4,5,6}) = 1 I_d ({3}) = 0

I_d ({1,2,6}) = 1

Як показано вище, підмножина {1, 2, 6} має коефіцієнт невідповідності 1. Аксіома 6 часто має велике значення для об'єднання підмножини. Наприклад, додавання аксіоми 6 змінює коефіцієнт невідповідності на 1 для підмножини {2, 6}. Це справедливо і для підмножини {3, 4, 5}.

Аналогічно, можемо визначити іншу характеристичну функцію, яка призначає набору аксіом 0, якщо поняття *A* є здійсненним і 1 в іншому випадку.

Визначення 3. Значення несумісності набору аксіом *K'* у відповідності до концепції *A* (*A* зустрічається у *K*) визначається як:

I_A*(K)* = {0, якщо *A* є здійсненною у відповідності до *K'*, 1 — в іншому випадку}.

Приклад 2. Деякі пов'язані з концепцією значення невідповідності робочого прикладу:

I_{A1}({1, 2}) = 1 I_{A1}({1, 3, 4, 5}) = 1

I_{A3}({3,4,5}) = 1 I_{A3}({3,4,9}) = 0

Мірою неконсистентності є кількісне визначення вкладу кожної аксіоми або твердження в базі знань в загальну невідповідність. Чим більше це значення, тим більше аксіома сприяє неконсистентності.

Шеплі [10] запропонував підхід, відомий як значення Шеплі, в контексті теорії ігор в 1953 році, який описує справедливий розподіл одержаних прибутків співробітництвом між кількома агентами.

Значення Шеплі визначається для гри, яка має n агентів. У грі агенти можуть утворювати коаліції, які є підмножиною n агентів. Коаліція виграє, коли всі його члени працюють разом як команда. Питання, яке може виникнути "який агент найбільше сприяє виграшу в різних коаліціях?"

Рішення цієї проблеми може допомогти визначити, який агент має більше значення для гри, ніж інші. Значення Шеплі пропонується для вирішення цієї проблеми. Основна ідея полягає в наступному. Припустимо, що агенти приєднуються до певної коаліції за певним порядком і виграш агента в цій коаліції - його граничний внесок у виграш коаліції. Значення Шеплі бере всі можливі порядки формування коаліції та вираховує граничний внесок агента.

Оскільки перевірка невідповідності може розглядатися як гра, кожна аксіома (або твердження) в базі знань можна вважати агентом. Аналогічно, внесок кожної аксіоми (або твердження) до неконсистентності може бути виміряний використовуючи значення Шеплі.

Нехай K — база знань, σ^K — набір всіх перестановок K, $n = |K|$ - потужність K (кількість неповторюваних елементів). Існує порядок $\sigma \sqsubseteq \sigma^K$, позначаємо p^{α}_{σ} для позначення набору всіх аксіом та тверджень в σ, які передують аксіомі (або твердженню) α.

Визначення 4. Значення Шеплі для аксіоми (або твердження) α в базі знань K визначається як:

$$S_{\alpha}(K) = \frac{1}{n!} \sum_{\sigma \sqsubseteq \sigma K} v(p^{\alpha}_{\sigma} \cup \{a\}) - v(p^{\alpha}_{\sigma})$$

Значення Шеплі можна безпосередньо обчислити з можливих коаліцій без урахування перестановок за наступним виразом:

$$S_\alpha(K) = \sum_{C \subseteq K} \frac{(c-1)!\,(n-c)!}{n!} (v(C) - v(C/_{\{a\}}))$$

де C - будь-яка коаліція аксіом та тверджень K, n = | K |, і c = | C |.

3.1 Міра неконсистентності на основі значення Шеплі

Приймаємо міру неконсистентності, визначену в Визначенні 2 (те ж саме обчислення може бути застосоване до поняття пов'язаної міри, визначеної в Визначенні 3) як характеристичну функцію, а потім використовуємо значення Шеплі для обчислення в якій мірі аксіома або твердження стосується невідповідності.

Наприклад, нехай, K - база знань, а α - аксіома (або твердження) в K, тоді значення Шеплі для α, виходячи з значення невідповідності I_d визначається як:

$$S_\alpha(K) = \sum_{C \subseteq K} \frac{(c-1)!\,(n-c)!}{n!} (I_d(C) - I_d(C/_{\{a\}}))$$

де n = | K | і c = | C |.

Значення Шеплі бази знань K - це вектор, де кожен елемент позначає значення Шеплі кожної аксіоми (або твердження) в K.

Приклад 3. Значення Шеплі бази знань K {C ⊔ ¬C ⊑ C ⊓ ¬C, T ⊑ ∃R.B, T ⊑ ∀R.¬B, A⊑D} = $(\frac{4}{6}, \frac{1}{6}, \frac{1}{6}, 0)$.

Приклад 4.Значення Шеплі для робочого прикладу: $((\frac{268}{7!}, \frac{250}{7!}, \frac{120}{7!}, \frac{120}{7!}, \frac{120}{7!}, \frac{3774}{7!}, \frac{268}{7!}, \frac{120}{7!}, 0)$

Це показує, що аксіома 6 є такою, що викликає більшість проблем. 3, 4, 5 і 8 однаково відповідальні за неконсистентності, як і 1 і 7. Аксіома 9 має значення 0, що означає, що вона не сприяє неконсистентності. Аксіома 2 має більше значення, ніж 3, 4 або 5, що показує, що неконсистентність однаково розподілена між 3, 4 і 5.

3.2 Властивості міри неконсистентності

В даному підрозділі представлені декілька спостережень відносно

властивостей міри неконсистентності.

Визначення 5. Характеристична функція v зростає, якщо *X* належить *Y*, *v(X) <= v(Y)*.

Зростаюча функція вказує на те, що додавання агентів до коаліції ніколи не призведе до зменшення значення Шеплі. Завдяки монотонності DL обробки, це можна довести наступним прикладом.

Пропозиція 1. Різке (пов'язане з поняттям) значення невідповідності (див. Визначення 2 та 3) зростає. Набір аксіом і тверджень називається конвергентним, якщо його значення невідповідності є конвергентним, тобто воно відповідає значенню неконсистентності супер наборів, і додавання будь-яких інших аксіом або тверджень не змінює значення невідповідності.

Визначення 6. Конвергентна підмножина *K′* бази знань *K* визначається як набір аксіом і тверджень, що задовольняє двом наступним властивостям:

1. $I_d(K') = 1$ (або $I_A(K') = 1$), та

2. $I_d(K'') = 0$ (або $I_A(K'') = 1$) для всіх $K'' \in K$

Інтуїтивно, *K′* є збіжною точкою, в якій значення невідповідності зміщується від 0 до 1. Існує пряме відношення між конвергентними підмножинами і мінімально неконсистентними підмножини бази знань, визначені наступним чином.

Визначення 7. Говорять, що *T′* є мінімально неконсистентною підмножиною бази знань *T*, якщо виконуються наступні твердження:

1. *T′* є неконсистентною

2. *T″* є неконсистентною для кожного *T″*, яке належить *T′*.

Пропозиція 1. Нехай *K′* (визначення 6) є мінімально неконсистентною підмножиною. Міра невідповідності коаліції *K′* може бути визначена за кількістю мінімально непослідовних підмножин, які будуть видалені, якщо видалити *K′* з бази знань. Іншими словами,

вимірюється вплив K' на базу знань, формалізований наступним чином.

Визначення 8. Міра невідповідності підмножини K' в базі знань K може бути визначена наступним чином:

$I_i(K') = |MIS(K)| - |MIS(K - K')|$

де $|MIS\ (K')|$ позначає кількість мінімально неконсистентних підмножин K'.

Приклад 4. У робочому прикладі зазначено чотири конвергентних підмножини, тобто чотири MIS. Приклад

$MIS_1 = \{1, 2, 7\}$, $MIS_2 = \{3, 4, 5, 8\}$, $MIS_3 = \{1, 3, 4, 5, 7\}$, $MIS_4 = \{6\}$

$I_i(\{1\}) = I_i(\{7\}) = I_i(\{3\}) = I_i(\{4\}) = I_i(\{5\}) = 2$

$I_i(\{2\}) = I_i(\{6\}) = I_i(\{8\}) = 1$

$I_i(\{9\}) = 0$

Очевидно, що видалення 1, 7, 3, 4 або 5 призведе до видалення найбільшої кількості невідповідностей. Видалення аксіоми 9 взагалі не вплине на неконсистентність.

3.3 Застосовувати міри невідповідності до виразів

Розглянуті в даній роботі невідповідності до цих пір застосовувалися до аксіом в онтологіях. Це виключає можливість перевірки компонентів аксіом. Зокрема, якщо невідповідність рівня одиничної аксіоми, то можна отримати лише два значення: консистентне або неконсистентне.

Вирази є складовою частиною аксіоми DL, тому можна визначити підхід, що виявляє частини аксіом, що відповідальні за неконсистентність. Отже, міра невідповідності може також застосовуватися до виразів, і це дозволяє нам це "подивитися всередину" аксіоми і визначите, яка частина аксіоми сприяє невідповідності.

4. Обчислювальна складність задачі

Найважливіше джерело обчислювальної складності при розрахунку значення Шеплі - це труднощі отримання міри неконсистентності аксіоми. Це безпосередньо залежить від складності перевірки послідовності в DL

обробці. Крім того, складність також пов'язана з самим процесом обчислення. Обчислення значення Шеплі розглядає всі підмножини аксіом / тверджень в базі знань, і, отже, це призводить до складності EXP-time. Однак насправді не потрібно обчислювати значення невідповідності всіх підмножин.

У DL, аксіоми можуть бути пов'язані одна з одною через відповідність структури. Наприклад, аксіома $A \sqsubseteq B$ структурно пов'язана з $\neg B \sqsubseteq T$, але не пов'язана з $\neg C \sqsubseteq D$. Додавання структурно непов'язаної аксіоми до коаліції не буде змінювати значення невідповідності. Структурна відповідність — еквівалентність відношення, отже, його можна використовувати для розбиття аксіом, як показано нижче. Такий підхід можна використовувати як оптимізацію, щоб прискорити обчислення невідповідності Шеплі.

Визначення 9. Аксіома структурно пов'язана з іншою аксіомою, якщо їх перетин (сукупність всіх (відхилених) назв концептів і імена ролей, що зустрічаються в аксіомі) не порожній. Структурна релевантність є транзитивним закриттям прямого структурного зв'язку.

Розбиття аксіом на групи на основі структурного зв'язку зберігає порядок значень Шеплі аксіом (і тверджень) всередині однієї групи. Іншими словами, якщо аксіома має більш високе значення Шеплі, ніж інша аксіома в групі, то в базі знань також буде вище значення Шеплі.

4.1 Оптимізація на основі властивостей міри невідповідності

Розбиття бази знань відповідно до структурної релевантності має сенс, коли розміри $|Ki|$ невеликі, особливо коли розмір $|Ki|$ обмежений константою. Однак це значною мірою залежить від конкретної бази знань. В даній роботі пропонується алгоритм розрахунку значення Шеплі, що базується на властивостях міри невідповідності, особливо властивості конвергентності. Цей метод спрямований на зменшення кількості перевірок послідовностей.

Основна ідея: згідно з визначенням 6 у розділі 3.2, конвергентна підмножина бази знань - це максимальна підмножина, значення невідповідності якої повинно бути розраховане. Будь-яка множина ковергентної бази знань може мати значення невідповідності 1 за рахунок монотонності DL. Отже, якщо підмножина є конвергентною, немає необхідності обчислювати її множину. Детальний алгоритм зазначений нижче.

Input: axiom α in K

Output: Shapley value of α

For all subsets K` ⊆ K

While I(K` ∪α) = 0

Go to next unvisited K`

For all supersets K`` of K`

I(K``) = 1

K`` → visited

Go to next K`

Compute Shapley value of α

Приклад 5. Щоб побачити, як цей алгоритм працює, застосуємо оптимізацію до одного з розділів робочого прикладу:

{A1 ⊑ A2 ⊓ ¬A ⊓A3, A2 ⊑ A ⊓ A4, A3 ⊑ A5 ⊓ A4, A4 ⊑ C ⊓ ∀S.¬B, A5 ⊑ ∃S.¬B, A1(a), A3(b)}.

Алгоритм спочатку обчислює значення неконсистентності $A1(a)$, коаліцію $A1(a)$ та будь-яку іншу аксіому, а потім коаліцію $A1(a)$ та будь-які дві інші аксіоми, оскільки коаліція $A1$ належить $A2 \sqcap \neg A \sqcap A3$, $A2 \sqsubseteq A \sqcap A4$ і $A1(a)$ має значення невідповідності 1, обчислення значення невідповідність множин пропускається.

Такий підхід застосовується і для множини

A1 ⊑ A2 ⊓ ¬A ⊓ A3, A3 ⊑ A5 ⊓ A4, A4 ⊑ C ⊓ ∀S.¬B, A5 ⊑ ∃S.¬B та A1(a).

5. Поєднання алгоритмів MI та Shapley

Оскільки можна розглянути мінімальні непослідовні підмножини як фундаментальні ознаки, що характеризують невідповідність, ми використовуємо аксіому тут як основну міру невідповідності.

Визначення 10

Значення MIV_C визначається як

$$MIV_C(K, \alpha) = \sum_{M \in MI(K) S.t. \alpha \in M} \frac{1}{|M|}$$

Це дозволяє нам визначити більш точне значення невідповідності, як це показано у наступному прикладі.

Приклад 6

Нехай K1 = {a, ¬a, ¬a ⊓ c, a ∨ d, ¬d, b ⊓ ¬b, e} ,

значення MIV_C :

$$MIV_C(K_1, \alpha) = 1; \; MIV_C(K_1, \neg\alpha) = \frac{5}{6}$$

$$MIV_C(K_1, \neg\alpha \wedge c) = \frac{1}{2}; \; MIV_C(K_1, \alpha \vee d) = \frac{1}{3}$$

$$MIV_C(K_1, \neg d) = \frac{1}{3}; \; MIV_C(K_1, b \wedge \neg b) = 1$$

Визначення 11 Метод невідповідності *MI* визначається як кількість мінімальних непослідовних множин *K*, тобто:

$I_{MI}(K) = |MI(K)|$

Приклад

K = {a, ¬a, ¬a ∧ c, a ∨ d, ¬d, b ∧ ¬b, e}

Таким чином ми отримуємо наступне

$I_{MI}(K) = 4$ $I_{MI}(\{a, \neg a, \neg a \wedge c\}) = 2$ $I_{MI}(\{b \wedge \neg b, e\}) = 1$ $I_{MI}(\{\neg a, \neg a \wedge c\}) = 0$

Пропозиція 3 Міри невідповідності *MI* I_{MI} є базовою мірою невідповідності, тобто вона задовольняє властивостям послідовності, монотонності, незалежності вільної формули. Наступний результат

показує, що дане MI Шеплі значення є таким самим як і MIV_C

Припущення 2. $S_{\alpha}^{I_{MI}}(K) = MIV_C(K, \alpha)$

Розглянемо спочатку наступну лему, яка буде використана в доведенні.

Лемма 1. Якщо проста гра у множинній формі з безліччю гравців $N = \{1, \ldots, n\}$ визначається єдиною множиною, яка перемагає $C' \sqsubseteq N$, наприклад:

$$v(C) = \begin{cases} 1 \; if \; C` \in C \\ 0 \end{cases}$$

Тоді значення Шеплі :

$$S_i(v) = \begin{cases} 0 \; if \; i \notin C` \\ \frac{1}{|C`|} \; if \; i \in C` \end{cases}$$

Доведення припущення 1.

1. Нехай α — вільна формула K, тоді $Simi\alpha = 0$, $MIVC\ (K, \alpha) = 0$. Відповідно $Simi\alpha = MIVC\ (K, \alpha)$

2. Нехай α не є вільною формулою K.

I_{MI} можна предстаити як $I_{MI}\ (C) = \Sigma_{M\ \epsilon\ MI\ (K)} \acute{M}(C)$, де $\acute{M}$ — характеристична функція

$$\acute{M}(C) = \begin{cases} 1 \; if \; M \subseteq C \\ 0 \end{cases}$$

Позначимо $\acute{M}(K)$ гру, визначену множинною формою від K і характеристичною функцією $\acute{M}$. Тоді;

$$S_{\alpha}^{I_{MI}}(K) =$$

$$\sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!} \left(I_{MI}(C) - I_{MI}\left({}^{C}/_{\{\alpha\}} \right) \right) =$$

$$\sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!} \left(\sum_{M \in MI(K)} \acute{M}\ (C) - \sum_{M \in MI(K)} \acute{M}\ \left({}^{C}/_{\{\alpha\}} \right) \right) =$$

$$\sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!} \left(\sum_{M \in MI(K)} \left(\acute{\text{M}}\,(\text{C}) - \acute{\text{M}}\left(C/_{\{\alpha\}} \right) \right) \right) =$$

$$\sum_{C \subseteq K} \sum_{M \in MI(K)} \frac{(c-1)!(n-c)!}{n!} \left(\acute{\text{M}}\,(\text{C}) - \acute{\text{M}}\left(C/_{\{\alpha\}} \right) \right) =$$

$$\sum_{M \in MI(K)} \sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!} \left(\acute{\text{M}}\,(\text{C}) - \acute{\text{M}}\left(C/_{\{\alpha\}} \right) \right) = \sum_{M \in MI(K)} S_\alpha(\acute{\text{M}}(\text{K}))$$

За лемою 1:

$$S_\alpha\left(\acute{\text{M}}\,(\text{K})\right) = \frac{1}{|M|}$$

Відповідно:

$$S_\alpha^{I_{MI}}(K) = \sum_{M \in MI(K)} \frac{1}{|M|} = MIV_C(K, \alpha)$$

Таким чином поєднання алгоритмів MI та Шеплі для визначення міри неконсистентності потенційно може бути більш ефективним з точки зору обчислювальної складності та часу виконання ніж алгоритми MI та Шеплі окремо.

На основі зазначеного вище припущення можемо визначити алгоритм для обчислення MI Шаплі міри неконсистентності.

В запропонованому алгоритмі для обчислення міри неконсистентності спочатку розраховуються мінімально неконсистентні підмножини. Потім для отриманих підмножин обчислюється значення Шеплі. Таким чином значення Шеплі розраховується не для всіх аксіом бази знань, а лише для її мінімально неконсистентних підмножин, що скорочує кількість обчислень.

6. Експериментальні результати

Щоб оцінити, описаний метод, був реалізований запропонований в попередньому розділі алгоритм і виконані експерименти. Алгоритм був

запущений для онтології Коала з 14 твердженнями та 19 аксіомами. Тести були виконані на системі Windows 10, процесор Intel Core i5 2.30 Гц, 8GB пам'ті. Реалізація розроблена за допомогою мови програмування в Java (JDK 1.5.0). Оригінальна онтологія Коала є консистентною і має 3 невиконувані концепції відповідно. В нашому експерименті, ми додали нових індивідів у невиконувані концепти для того, щоб зробити онтологію неконсистентною. В онтології Коали існували три джерела невідповідності, і для розрахунку міри невідповідності за допомогою алгоритму Шеплі, MI та MI-Шеплі необхідно 467 мс, 546 мс та 350 мс відповідно.

Також була проведена серія експериментів для різних онтологій. Результат показав, що швидкість обробки неконсистентності онтології збільшилася в середньому в 1,5 рази.

Висновки. На сьогодні існує декілька методів обчислення неконсистентностей онтологій. В даній роботі був запропонований підхід на основі поєднання алгоритмів MIV та Шеплі. Також була визначена еквівалентність між значеннями невідповідності Шеплі і мінімальним неконсистентним значенням, що дозволяє модифікувати алгоритм Шеплі за допомогою мінімально неконсистентних підмножин. Запропонований алгоритм був протестований на різних за розміром та кількістю джерел неконсистентностей онтологіях. В середньому швидкість обробки неконсистентностей зменшилася в 1,5 рази.

Література

1. Deng, X., Haarslev, V., Shiri, N.: A framework for explaining reasoning in description logics. In: Proceedings of the AAAI Fall Symposium on Explanation-aware Computing, Washington, DC, USA, AAAI Press (2005) 189–204.
2. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging owl ontologies. In:

Proceedings of the 14th International World Wide Web Conference (WWW 2005), Chiba, Japan, ACM Press (2005) 633–640.

3. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proceedings of the eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03), Acapulco, Mexico, Morgan Kaufmann (2003) 355–362.
4. Hunter, A., Konieczny, S.: Shapley inconsistency values. In: Proceedings of the International Conference on Knowledge Representation (KR'06), Windermere, UK, AAAI Press (2006) 249–259.
5. Conitzer, V., Sandholm, T.: Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence (AAAI 2004), San Jose, California, USA, AAAI Press/The MIT Press (2004) 219–225.
6. In: OWL Web Ontology Language Overview. (2004) http://www.w3.org/TR/owl-features/
7. Baader, F., Nutt, W.: Basic description logic. In Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003) 5–44.
8. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006). Volume 4130 of Lecture Notes in Artificial Intelligence., Seatle, Washingtion, USA, Springer (2006) 292–297.
9. Haarslev, V., M¨oller, R.: Racer system description. In R. Gori, A. Leitsch, T.N., ed.: Proceedings of International Joint Conference on Automated Reasoning (IJCAR 2001), Siena, Italy, Springer-Verlag (2001) 701–705.

10. Shapley, L.: A value for n-person games. In Kuhn, H., Tucker, A., eds.: Contributions to the Theory of Games. Volume 2. Princeton University Press (1953) 307–317.