

# ВИМІРЮВАННЯ, КОНТРОЛЬ І ТЕХНІЧНА ДІАГНОСТИКА

УДК 681.518:662.248:004.89

## АБДУКТИВНЕ ВИВЕДЕННЯ ЗНАНЬ ПРО ПРОЦЕС БУРІННЯ НА ОСНОВІ МУЛЬТИМЕДІЙНИХ ДАНИХ ПРО БУРОВЕ ОБЛАДНАННЯ

В. Д. Мельник

ІФНТУНГ, 76019, м. Івано-Франківськ, вул. Карпатська, 15;  
e-mail: doc.edu.if@gmail.com

Основною метою пропонованого дослідження є побудова методології поєднання технології видобування даних з фреймворком логічного програмування в обмеженнях, що дозволятиме інтеграцію доменних знань про процес буріння в загальну структуру видобування даних та знань про нафтогазову предметну область з можливостями їх безпосереднього застосування на рівні вибраного класифікатора, зокрема, у вигляді дерева рішень при використанні чітких та імовірнісних представлень на рівні фреймворк-застосування, що в своїй основі ґрунтується на абдуктивних методах побудови логічного виведення. Методи дослідження, що застосовуються, зводяться до концепції абдуктивного фреймворку, який на вході приймає набори даних у формі послідовності графічних маркерів, що відповідають конкретним рівням або мета-рівням доменних знань в формі растрових зображень, що загалом уможлиблює побудову обчислюваних рішень, що виконують ефективну верифікацію початкових спостережуваних гіпотез. Основним результатом є те, що виконуване застосування класифікатора у формі деревоподібної структури мережі обмежень дозволяє застосування прямих абдукційних обчислень до нових спостережень при досягненні відповідного порогового значення кількості правил, представлених у формі «умова релевантності – умова задоволення» що, в свою чергу, значно розширює область застосування абдуктивних обчислень загалом, зокрема, до рівня багатозначної логіки. Як основний висновок слід виділити представлене застосування методології логічного програмування в обмеженнях до задач видобування чистих реляційних мультимедійних даних або масивів даних, що потребують застосування первинного очищення шляхом внесення послідовності значущих графічних маркерів на основі дескриптивних засобів першого порядку, що загалом дозволяє побудову застосування в обмежених пошукових просторах на основі маркованих та імовірнісних значень, що в кінцевому підсумку дозволяє розширення початкового фреймворку до рівня абдуктивного імовірнісного фреймворку, що базується на семантично обґрунтованих розширеннях логіки першого порядку.

Ключові слова: знання, маркери, інтелектуальна підтримка прийняття рішень, буріння нафтових і газових свердловин, логічне виведення, правила, бази знань, абдуктивний фреймворк, коефіцієнти впевненості, обмеження, вагові значення.

Основной целью предлагаемого исследования является построение методологии сочетания технологии добычи данных с фреймворком логического программирования в ограничениях, что позволит интеграцию доменных знаний о процессе бурения в общую структуру добычи данных и знаний о нефтегазовой предметной области с возможностями их непосредственного применения на уровне выбранного классификатора, в частности, в виде дерева решений при использовании четких и вероятностных представлений для структуры фреймворк-приложения, что в своей основе базируется на абдуктивных методах построения логического вывода. Методы исследования, что применяются, сводятся к концепции абдуктивного фреймворка, который на входе принимает наборы данных в форме последовательности графических маркеров, отвечающих конкретным уровням или мета-уровням доменных знаний в форме растровых изображений, что в целом позволяет построение вычислительных решений, которые выполняют эффективную верификацию начальных наблюдаемых гипотез. Основным результатом является то, что выполняемое применение классификатора в форме древовидной структуры сети ограничений позволяет применение прямых абдукционных вычислений к новым наблюдениям при достижении соответствующего порогового значения количества правил представленных в форме «условие релевантности - условие удовлетворения» что, в свою очередь, значительно расширяет область применения абдуктивных вычислений в целом, в частности, до уровня многозначной логики. В качестве основного вывода следует выделить представленное применение методологии логического программирования в ограничениях к задачам добычи чистых реляционных мультимедийных данных или массивов данных, требующих применения первичной очистки путем внесения последова-

тельности значимых графических маркеров на основе дескриптивных средств первого порядка, что в целом позволяет построение применения в ограниченных поисковых пространствах на основе маркированных и вероятностных значений, что в конечном итоге позволяет расширение начального фреймворка до уровня абдуктивного вероятностного фреймворка, который будет базироваться на семантически обоснованных расширениях логики первого порядка.

Ключевые слова: знания, маркеры, интеллектуальная поддержка принятия решений, бурение нефтяных и газовых скважин, логический вывод, правила, базы знаний, абдуктивный фреймворк, коэффициенты уверенности, ограничения, весовые значения.

*The main objective of the proposed research is to build up the methodology for combining data mining technologies with the framework of logic programming with constraints that allow the integration of domain knowledge's about the drilling process into the overall structure of the data and knowledge's mining for the oil and gas domain with the possibilities of direct application with the selected classifier, particularly, in the form of decision tree that make use of crispy and possibilistic level of framework application that basically implements the abductive methods of inference construction. The used research methods can be reduced to the pure concept of the abductive framework implementation which at the very input receives the data sets in the form of the graphic markers that correspond to the specific levels or meta-levels for the domain knowledges in the form of raster images, which generally allows to build the computable solutions that perform effective verification of the initial hypotheses that were observed. The main result is that the executable application classifier in the form of a tree structure of the constraints network allows the use of direct abductive computing to the new observations with the purpose of achieving the appropriate threshold number of rules presented in the form of «relevancy condition – satisfaction condition» that, in turn, greatly expands the scope of abductive computing as a whole, particularly the level of multiple-valued logic. As the main result there should be highlighted the submitted application methodology of logic programming with constraints to the problems of extracting pure relational media or data sets that require the use of primary treatment by creating of the sequences of significant graphical markers based on descriptive tools of the first order, which generally allows the building of applications for the restricted search spaces based on marked and probabilistic values, which ultimately allows the extension of the original framework for the level of abductive probabilistic framework that will be based on the semantically founded extensions of first-order logic*

Keywords: knowledge, markers, intelligent decision support, drilling of oil and gas wells, logical inference, rules, knowledge bases, abductive framework, certainty factors, constraints, weight values.

**Вступ.** Процес абдукції [1–4] полягає в обчисленні певних пояснень для спостережень і є прикладом немонотонних міркувань, оскільки пояснення, що відповідає одному стану бази знань може ставати суперечливим при надходженні деякої нової інформації. Абдуктивні міркування характеризуються існуванням множинних пояснень і вибір преференційного пояснення є важливою складовою даної проблеми.

Дослідження дедукції виконується в контексті логічного програмування [5–6].

Якщо розглядати інформаційний потенціал нафтогазовидобувного підприємства в цілому [7], зокрема щодо даних та знань про процес буріння, як досягнутого рівня задоволення інформаційних потреб реальних операторів технологічного процесу буріння, що забезпечуватиме необхідний рівень видобутку нафти і газу, то інформаційний ресурс як частина використаного інформаційного потенціалу для досягнення поставленої мети – збільшення видобутку вуглеводнів. Даний процес включає в себе всі джерела інформаційних потреб і послуг необхідних в нафтогазовидобувній промисловості, особливо це стосується їх інтелектуального супроводу, наприклад для випадку бурового обладнання.

Застосування методу абдукції є особливо ефективним в задачах класифікації [7–9] і пояснення результатів застосування технології видобування даних. Зокрема можна вводити класифікаційні правила виведені з даних і інформацію отриману від експертів предметної області. Також можна використовувати асоціативні правила із введеними коефіцієнтами впевненості, що дозволяє підвищити загальну

якість результатів і рівень автоматизації процесу без безпосередньої потреби у наявних попередніх знаннях та експертному досвіді щодо предметної області [10–11].

Набори обмежень [12–14], що будуються в режимі виконання приймають участь в процесі пошуку, тобто це означає, що поточний стан набору обмежень може викликати нові виведення або запускати процедуру зворотного висновку відповідно. Такий рівень рішення досягається шляхом імплементації для кожного обмеження домену, що є набором міркувань над множиною термів і скінченного домену виразів для обмежень, що дозволяють керувати моніторингом стану виразу по відношенню до поточного набору обмежень.

У кожному випадку коли виконується вибір нової гіпотези для виконання обчислень, відповідна процедура виконує перевірку її консистентності (несуперечливості) по підношенню до релевантних обмежень цілісності (в даному випадку кожна така процедура припускає принаймні одне виводиме входження в кожному обмеженні цілісності), що відкриває відповідно нову фазу консистентності (несуперечливості). У загальному випадку легко побачити, що фаза консистентності (несуперечливості) може вимагати, в свою чергу, побудови нової абдуктивної фази і так далі.

Також є бажаним, якщо така процедура гарантовано не генерує жодних пояснень, що є не консистентними з базою знань. Така ситуація є неможливою в загальному випадку, оскільки консистентність пояснення з базою знань повинна бути перевірена поза абдуктивною системою інференції міркувань даного рівня складності відповідно. Слід також відмітити, що не

всі не консистентні пояснення генеруються системою, навпаки система може генерувати тільки ті пояснення, що припускають існування літералів, які можуть бути досягнуті відповідно з початкових формул шляхом застосування ланцюга зворотного висновку. Таким чином, задача визначеності консистентності (несуперечливості) є нерозв'язною в загальному випадку хоча існують вирішувані випадки і ряд пояснень може бути швидко відкинуто як такі, що неконсистентні з початковою базою знань або системою чи ієрархією обмежень відповідно. Зокрема, в більшості випадків припущення може бути відкинуто якщо воно порушує множинну обмежень, щодо виду або впорядкування виведень. Тому обмеження виду є особливо ефективними щодо усунення неконсистентних пояснень.

Роль абдуктивних міркувань полягає в забезпеченні автоматичної редукції (звуження) представлень проблем високого рівня і цілей верхнього рівня відповідно до нижчого рівня обчислювальних задач для незалежних форм представлення загальної проблеми. Використання техніки логічного програмування в обмеженнях (CLP – *constraints logic programming*) є фундаментальним рішенням для підсилення ефективності обчислювального процесу абдуктивного висновку подібно до того, як це застосовується на високому рівні представлення проблеми, що розглядається. Власне, такий підхід забезпечує зв'язок між властивостями домену проблем високого рівня і доменно-незалежними методами рішення.

**Недослідженим** залишається питання використання засобів абдуктивних міркувань для задач видобування даних. Слід дослідити, яким чином класифікація може бути інтерпретована як одна з задач абдуктивного логічного програмування, що дозволяє використовувати доменні обмеження задані користувачем. Моделі класифікації на основі дерев рішень інтерпретуються абдуктивним способом, використовуючи обмеження і отримуючи покращення ефективності у випадках відсутності певних даних. Слід виконати розширення загального фреймворку з метою розгляду імовірнісної інформації на основі базових та формальних теорій, що виводяться, що зробить абдуктивний фреймворк, який базується на вартісних коефіцієнтах, придатним для додатків видобування даних, що дозволятиме також покращити загальну якість результатів процесів видобування та виведення загалом на множині баз даних та знань, що описують предметну область.

Таким чином, основною **метою** пропонованого дослідження є обґрунтування того факту, що технологія абдукції може бути ефективно використана для задач видобування даних в формі їх виведення на основі масивів мультимедійних входжень, що описують нафтогазові об'єкти.

**Основна частина.** Фреймворк абдуктивного логічного програмування означається у вигляді триплета  $\langle CLP, Hyp^{Set}, C^{LF} \rangle$ , що склада-

ється з логічної програми  $CLP$ , множини базових виводимих атомів  $Hyp^{Set}$  і множини логічних формул  $C^{LF}$ , що називаються обмеженнями цілісності. Атоми в  $Hyp^{Set}$  є можливими абдуктивними гіпотезами, які можуть розглядатися в якості припущень з метою пояснення заданих спостережень в контексті  $CLP$ , при умові, що ці припущення є консистентними (несуперечливими) з обмеженнями цілісності в  $C^{LF}$ . В багатьох випадках зручно визначати  $Hyp^{Set}$  як множину предикатних символів з деяким планованим значенням, що виводяться, таким, що кожен базовий атом, чий предикатний символ є в  $Hyp^{Set}$ , є можливою гіпотезою.

Для початково заданого абдуктивного фреймворку загальне визначення абдуктивних пояснень можна ввести наступним чином.

**Означення 1.** Нехай  $\langle CLP, Hyp^{Set}, C^{LF} \rangle$  є абдуктивним фреймворком і нехай  $Case$  є ціллю  $Goal$ . Тоді під абдуктивним поясненням для  $Case$  будемо розуміти множину  $\Delta \subseteq Hyp^{Set}$  базових атомів, що виводяться, таких, що:

- 1)  $[CLP \cup \Delta \neq Case]$ .
- 2)  $CLP \cup \Delta \cup C^{LF}$  є консистентним (несуперечливим).

**Означення 2.** Нехай  $\langle CLP, Hyp^{Set}, C^{LF} \rangle$  є абдуктивним фреймворком,  $\Delta_{mit}$  є множиною виводимих входжень і  $Case$  є ціллю. Вважатимемо, що  $\Delta$  є абдуктивним поясненням для  $Case$  при заданому  $\Delta_{mit}$ , якщо  $\Delta' \subset \Delta$  є абдуктивним поясненням для  $Case$ .

При цьому слід відмітити, що таке формулювання імплікує, що задана множина входжень  $\Delta_0$ , що виводяться, повинна бути консистентною (несуперечливою) з заданими обмеженнями цілісності  $C^{LF}$ .

**Означення 3.** Нехай  $\langle CLP, Hyp^{Set}, C^{LF} \rangle$  є абдуктивним фреймворком,  $\Delta_{mit}$  є множиною входжень, що виводяться, і  $Case$  є ціллю. Вважатимемо, що  $\Delta \in \Delta_{mit}$  – мінімальним поясненням для  $Case$ , якщо  $\Delta$  є поясненням для  $Case$  при заданому  $\Delta_{mit}$  і для кожної власної підмножини  $\Delta'$  для  $\Delta$  ( $\Delta' \subset \Delta$ ),  $\Delta'$  є поясненням для  $Case$  при заданому  $\Delta_{mit}$ .

Введена ідея абдукції, згідно описаного представлення, має природнім чином побудований обчислювальний відповідник в межах власного розширення стандартного SLD-базованого обчислювального механізму логічного програмування. Зокрема, коли обмеження цілісності представляється у формі заперечень, тобто у формі  $\neg(p_1, \dots, p_n)$ , то процедура доведення в даному випадку може бути успішно адаптована для обчислення мінімальних абдуктивних пояснень.

В рамках такого фреймворку можна визначити абдуктивну шему у вигляді  $Schema^{ab} = \langle F_{Set}, Hyp^{Set} \rangle$ , де  $F_{Set}$  – множина тверджень Хорна. Змінні в  $F_{Set}$  є неявно універсально квантифікованими. Нехай  $F_{Set}$  є множиною базових інстанціцій елементів з  $F_{Set}$ , а  $Hyp^{Set}$  є множиною атомів, що називаються очікуваними або можливими гіпотезами:

$$Hyp^{Set} = Hyp^{Expc} \cup Hyp^{PsbI}.$$

Система будує наступні припущення стосовно базової теорії:

1. Не існує правил в  $F_{Set}$ , заголовки яких уніфікуються з деяким членом в  $Hyp^{Set}$ .

2. Якщо  $F_{Set}$  є множиною базових інстанціцій елементів  $F$ , то є можливим присвоєння числових натуральних значень до кожного базового атома такого, що для кожного правила в  $F_{Set}$  атоми в тілі правила є строго меншими, ніж атоми в заголовках.

3. Для кожного значення  $AF^{CF}$ , що не передбачалося спочатку, якщо  $AF^{CF}$  є істинним, тоді принаймні одне з тіл  $rc_1$ , таке, що  $AF^{CF} \leftarrow rc_1$ , повинно бути істинним. Крім того, можна стверджувати, що правила в  $F_{Set}$  мають місце для кожного атома, що подається як базовий (тобто такий, що не є гіпотезою).

4. Тіла правил в  $F_{Set}$  для кожного атома є взаємовиключними.

На формальному рівні фреймворк абдукції з ваговими коефіцієнтами є способом представлення мультимедійних знань з домену предметної області, що моделюються кортежем:  $AF^{CF} = (Hyp^{Set}, Rules, CF, Case^{Set})$ , де:  $Hyp^{Set}$  – множина гіпотез або пропозиційних тверджень;  $Rules$  – множина правил виду:  $\{ \{ rc_1 \wedge rc_2 \wedge \dots \wedge rc_{n_i} \} \} \rightarrow rc_{n_i+1}$ , де  $rc_1, \dots, rc_{n_i+1} \in Hyp^{Set}$ ;  $c$  – відповідна функція,  $c : Hyp^{Set} \rightarrow Rules^+$ , де  $c(rc)$  розглядається як вартість (вагове значення) очікуваності гіпотези  $rc \in Hyp^{Set}$ ;  $Case \in Hyp^{Set}$  – множина цілей або підтверджень.

Таким чином, кінцеве завдання полягає в пошуку доведення з найменшою вартістю (ваговим коефіцієнтом) для заданого підтвердження, де вартість доведення обчислюється як сума вартостей всіх гіпотез, що повинні припускатися з метою завершення доведення. Відповідно, гіпотеза може бути перетворена в істинну відповідно двома способами:

1) вона може припускатися як істинна, шляхом оцінки вартості її припущення;

2) шляхом доведення.

При цьому слід зазначити: якщо гіпотеза проявляється як наслідок правила  $Rule$ , тоді вона може бути доведена без визначення вартості (ваги) як така, що є істинною шляхом пере-

творення всіх антецедентів в  $Rule$  в істинні, шляхом припущення або доведення. Якщо відповідна гіпотеза не проявляється як консеквент для жодного правила, тоді вона не може бути доведена і, відповідно, вона може бути встановлена як істинна тільки шляхом припущення. Якщо вартість гіпотези є достатньо великою, тоді вона не може бути встановлена шляхом припущення і вона може бути тільки доведена. Для цього множина гіпотез  $HypSet$  розділяється на дві множини  $Hyp_1^{Set}$  та  $Hyp_2^{Set}$ . Перша множина містить гіпотези, що не можуть бути доведені, тобто такі, що ніколи не проявляються в правих частинах правила. Друга множина містить всі інші гіпотези.

В загальному випадку, в абдуктивних міркуваннях множина  $AF^{CF}$  розглядається як множина здійснених рішень для проблеми доведення з найменшою вартістю, якщо відповідно вона є підмножиною для  $Hyp_1^{Set}$ , що є достатнім для доведення множини цілей  $Goal^{Set}$ . Відповідно, оптимальним рішенням для проблеми доведення з найменшою вартістю є здійснене рішення, що мінімізує задану загальну вартість, тобто суму вартостей всіх гіпотез  $AF^{CF}$ .

В концепції абдуктивного навчання (ACL) на вході система отримує деяку абдуктивну теорію  $T = \langle CLP, Hyp^{Set}, C^{LF} \rangle$  як спосіб представлення знань і дві множини базових атомів в якості прикладів. Відповідно, засоби ACL дозволяють знаходити абдуктивну теорію з  $CLP' \supset CLP$  та  $C^{LF} \supset C^{LF}$  такі, що  $T' = Case_{sample} CASE_{Set}^+$  та  $\forall Case_{sample}^- \in CASE_{Set}^-$ ,  $T' \neq_{rc} Case_{sample}^-$ , де  $CASE_{Set}^+$  відповідає за кон'юнкцію всіх позитивних прикладів (що позначаються знаком "+"); відповідно  $CASE_{Set}^-$  відповідає за диз'юнкцію всіх негативних прикладів (що позначаються знаком "-").

Таким чином, вивчення абдуктивних теорій дозволяє виконувати процес навчання, базуючись на неповних знаннях, оскільки виводимі предикати можуть бути використані для представлення неповних знань.

Таким чином, існуючі алгоритми дозволяють вивчати, перш за все, всі правила, після чого всі обмеження. Вивчення правил виконується ІЛР алгоритмом, що функціонує зверху вниз, а тестування покриття виконується з використанням абдуктивної процедури доведення. В даний алгоритм закладено пучкову пошукову стратегію спеціальну евристичну функцію, що дає відмінні вагові значення для прикладів, що покриваються засобами абдукції або без її використання. На виході фази вивчення правил отримується відповідна множина правил і множина пояснень за принципом: одне пояснення, можливо порожнє, для кожного позитивного прикладу і всі можливі пояснення, серед яких не має порожніх, для кожного негативного прикладу. Вимога того, що жодне з

пояснень для негативних прикладів не є порожнім необхідна для того, щоб виключити їх всіх у наступній фазі, що полягає у вивченні обмежень.

Обмеження вивчається на основі ICL, при чому вхідний потік ICL складається з абдуктивних пояснень для позитивних прикладів, як виду позитивних інтерпретацій і абдуктивні пояснення для негативних прикладів розглядаються, відповідно, як вид негативних інтерпретацій. Таким чином, в кінцевому підсумку вивчені правила разом з правилами з множини початкових знань утворюють основу системи знань в ICL.

Основне завдання полягає в тому, щоб показати як класифікаційна задача може бути перетворена безпосередньо в абдукційну проблему шляхом застосування дерев рішень. Для вирішення даної задачі необхідно ввести формальні означення, які описують процес класифікації на основі дерев рішень і дозволяють виконувати трансформацію дерев рішень у відповідну абдуктивну логічну програму.

Для заданої мережі рішень з обмеженнями  $CN$  і шляху  $route$  в  $CN$  позначимо через  $Case^{Example}(route)$  деякий приклад:  $sc_1 = v_1, \dots, sc_k = v_k$ , де кожне  $sc_i$  не є листовим вузлом в  $CN$  і кожне  $v_i$  є значенням, що позначає вітку, яка виходить з  $sc_i$ . Означимо через  $Case^{CLASS}(route)$  – клас, що позначає лист вузла для  $route$ .

Визначимо концепції класифікації прикладів засобами дерев рішень у випадку відсутньої інформації. Для цього слід переписати звичайне означення класифікації як приклад окремого випадку, в якому не має відсутньої інформації.

**Означення 4.** Нехай  $CN^{Solve}$  є мережею рішень із введеними обмеженнями:

$$CN^{Solve} : CH^{CNSolve} = \bigcup_i \left[ CS_i = \bigcup_j CSet_j \right],$$

$Solve^{sample}$  – екземпляр (приклад) рішення і  $Class$  – кейс клас рішення. Будемо вважати, що можна стверджувати той факт, що екземпляр  $Solve^{sample}$  може бути класифікований як  $Class$ , засобами  $CN$  через  $A$ , що позначається через  $CN \rightarrow Class, A$  якщо існує шлях  $route$  в  $CN$  з вузлом  $Case^{CLASS}(route) = Class$  таке, що  $Solve^{sample} \cup Case^{Example}(route)$  є теж відповідним екземпляром (прикладом) рішення і  $A = Case^{Example}(route) / Solve^{sample}$ .

Якщо  $CN \xrightarrow{CN, \{ \}} Case_{sample}$ , то говорять, що екземпляр (приклад)  $Case_{sample}$  класифікується як  $CN$  засобами  $CN$ , що представляється як  $CN \xrightarrow{CN} Case_{sample}$ .

Слід також відмітити, що в представленні  $CN \xrightarrow{CN, A} Case_{sample}$  умова того, що

$Case_{sample} \cup Hyp_{Names}^{Set}(route)$  повинно бути екземпляром гарантує відповідно, що значення атрибуту для  $Case_{sample}$  є сумісними з множиною тестів представлених через  $(route)$ . Іншими словами, якщо для деякого атрибуту  $rc$ ,  $rc = [v \in Case_{sample}]$  та  $rc = [v' \in Hyp_{Names}^{Set}(route)]$ , тоді  $v = v'$ . Крім того,  $A$ , представлятиме розширення до рівня прикладу  $Case_{sample}$ , яке є необхідним для виконання його класифікації як  $CN$  з вибраного шляху.

Розглянемо тепер спосіб перетворення мережі рішень  $CN$  в абдуктивну логічну програму з обмеженнями. Нехай  $CN$  – мережа рішень і  $route$  – шлях в  $CN$ , тоді правило  $Rule_{route} = [rc_{route} \rightarrow sc_{route}]$  асоційоване з  $route$  буде твердженням виду  $Solve^{sample} \leftarrow \{sc_1(v_1), \dots, sc_n(v_n)\}$ , таке, що:

- 1)  $Case^{Example}(route) = Class$ ;
- 2)  $Solve^{sample}(route) = \{sc_1 = v_1, \dots, sc_n = v_n\}$ .

Слід відмітити, що імена атрибутів розглядаються як унарні предикатні символи і, що пара атрибут/значення  $a = v$  відображається в атом  $rc(route)$ . Більше того, при заданому  $(v)$  і при заданому шляху  $route$ ,  $Hyp_{Names}^{Set}(route)$  слід розглядати як екземпляр (приклад). Програма  $CLP_{CN}$  на основі тверджень Хорна, асоційована з деревом рішень  $CN$  є множиною правил виду:

$$CLP^{CN} = \{r_{route} / route\},$$

де  $route$  є шляхом в  $CN$ .

В кінцевому підсумку виконується зв'язування з деревом рішень  $CN$  множини  $C^{LF}$  канонічних обмежень цілісності, що містять заперечення виду:

$$\leftarrow rc(x), rc(y), x \neq y,$$

для кожного атрибута релевантності  $rc \in RC^{Set}$ .

Імена атрибутів розглядаються як предикати, що виводяться, множина всіх предикатів, що виводяться, буде представляти можливі класифікації. Обмеження цілісності, означені подібним чином, забезпечують той факт, що в кожній консистентній множині предикатів, що виводяться, кожен атрибут зустрічатиметься не більше одного разу.

Для заданого екземпляру (прикладу)  $Case_{sample} = \{rc_1 = v_1, \dots, rc_k = v_k\}$ , нехай  $A_{Case_{sample}}$  є множиною атомів:

$$A_{Case_{sample}} = \{rc_1(v_1), \dots, rc_k(v_k)\}.$$

Таким чином, приведені формальні представлення дають всі необхідні складові для прив'язки абдуктивного фреймворку до дерева рішень  $CN$ .

**Означення 5.** Для заданої мережі рішень  $CN$  абдуктивний фреймворк  $AF_{CN}^{ab}$  асоційований з  $CN$  представлятиметься кортежем:

$AF_{CN}^{ab} = \langle CLP_{CN}, Hyp_{Names}^{Set}, CH_{CN} \rangle$ , де  $CLP_{CN}$  – це логічні програми з обмеженнями, асоційовані з  $CN$ ,  $CH_{CN}$  – це множина обмежень накладених на  $CN$ ,  $Hyp_{Names}^{Set}$  – це множина імен атрибутів гіпотез.

Виконаємо формалізацію відповідності між класифікаціями в  $CN$  і абдуктивними поясненнями в  $AF_{CN}^{ab}$ .

Нехай  $CN$  є деревом рішень і  $AF_{CN}^{ab}$  є відповідним абдуктивним фреймворком, а  $Case_{sample}$  є екземпляром (прикладом), тоді  $\Delta \in \Delta_{Case_{sample}}$  – мінімальним поясненням для  $Class$  по відношенню до  $AF_{CN}^{ab}$ , якщо і тільки якщо для деякого шляху  $route$  в  $CN$  ми матимемо, що  $CN \xrightarrow{Class, \Delta} Case_{sample}$  і  $\Delta = \Delta_A$ . Припустимо, що  $CN \xrightarrow{Class, \Delta} Case_{sample}$ , тоді згідно введених означень для деякого шляху  $route$  в  $CN$ ,  $Case_{sample} \cup Solve^{sample}(route)$  є екземпляром (прикладом) і  $\Delta = Solve^{sample}(route)/Case_{sample}$ . Розглянемо правило виду  $rc_{route}$ :

$$Class \leftarrow [rc_1(v_1), \dots, rc_n(v_n)].$$

Нехай  $SC(r_{route}) = \{rc_1(v_1), \dots, rc_n(v_n)\}$ , тоді зрозуміло, що  $SC(rc) = \Delta_{Solve^{sample}(route)}$  і  $[route_{CN} \cup SC(rc_{route})] = Class$ . Нехай  $\Delta = SC(rc_{route})/\Delta_{Case_{sample}}$ , зрозуміло, що  $[route_{CN} \cup \Delta_{Case_{sample}} \cup \Delta] = Class$  та  $\Delta = \Delta_A$ . Більше того, оскільки  $Case_{sample} \cup Solve^{sample}(route)$  є екземпляром (прикладом рішення), то  $(\Delta_{Case_{sample}} \cup \Delta) \subseteq \Delta_{[Case_{sample} \cup Solve^{sample}(route)]}$ , та  $SC_{CN} \cup \Delta_{Case_{sample}} \cup \Delta \cup CH_{CN}$  є консистентними. Таким чином,  $\Delta$  виступає в ролі пояснення для  $Class$  при заданому  $\Delta_{Case_{sample}}$ . Згідно із способом побудови,  $\Delta$  є також  $\Delta_{Case_{sample}}$  – мінімальним поясненням.

Припустимо, що  $\Delta \in \Delta_{Case_{sample}}$  – мінімальним поясненням для  $Class$ . Очевидно, що  $\Delta \cap \Delta_{Case_{sample}} = \{ \}$  на основі властивостей мінімальності для  $\Delta$ . Оскільки згідно із способом побудови  $CH_{CN}$ ,  $[CH_{CN} \cup \Delta_{Case_{sample}} \cup \Delta] = Class$ , то тоді існує шлях  $route$  в  $CN$ , такий, що  $rc_{route}$  є правилом виду:

$$Class \leftarrow [rc_1(v_1), \dots, rc_n(v_n)] \text{ і } \{rc_1(v_1), \dots, rc_n(v_n)\} \subseteq [\Delta_{Case_{sample}} \cup \Delta].$$

Нехай  $\Delta = Solve^{sample}(route)/Case_{sample}$ . За способом побудови і мінімальності  $\Delta$  ми матимемо, що  $\Delta = \Delta_A = \{rc_1(v_1), \dots, rc_n(v_n)\}/\Delta_{Case_{sample}}$ .

В кінцевому підсумку буде очевидно, що  $Case_{sample} \cup Solve^{sample}(route)$  є екземпляром (прикладом) на основі консистентності  $SC \cup \Delta_{Case_{sample}} \cup \Delta \cup CH_{CN}$  і загального спостереження того, що  $\Delta_{Case_{sample} \cup Solve^{sample}(route)} = \Delta_{Case_{sample}} \cup \Delta$ .

На рисунку 1 представлено кінцеву діаграму класифікаційної системи. Як показано на даному рисунку, вхідний потік даних складають інстанції для класифікації і відповідний набір даних. Дерево рішень індукується з набору даних за допомогою відповідного алгоритму інстанціації, а інстанції алгоритмів видобувають часті входження і відповідні асоціативні правила в формі відношень «умова релевантності – умова задоволення». Абсолютне значення підтримки частих входжень є параметром, що використовується для обчислення вартості їх абдукції (як частини інстанційованого процесу видобування даних і знань).

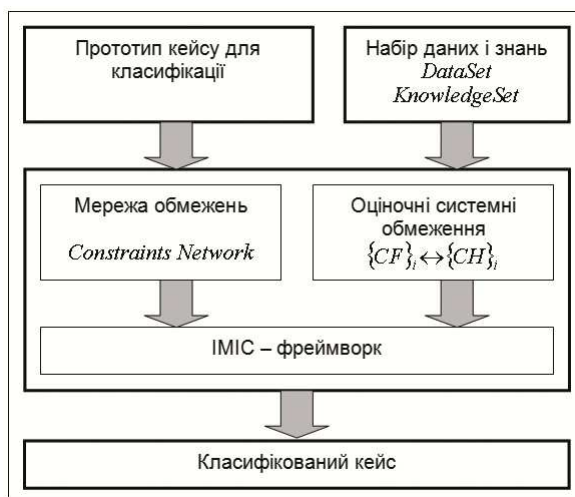


Рисунок 1 – Абдукційно-класифікаційна схема виділення кейсів

Таким чином, проєктований фреймворк базується на абдукції з вартісними (ваговими) коефіцієнтами, де кожен елемент, що виводиться, має асоційований з ним вартісний (ваговий) коефіцієнт, що повинен бути активований в ході побудови припущення. В пропонованому підході правила також мають асоційоване значення вартості (ваговий коефіцієнт), що повинен бути активований, якщо правило використовується в процесі класифікації. Відповідно, кінцевим класом буде вважатися клас, що імплікується мінімальною вартістю (мінімальним сумарним ваговим коефіцієнтом) відповідного абдуктивного виведення.

Наведемо основні характеристики абдуктивного фреймворку, що використовується:

1) кожен атом, що входить принаймні в одне правило є асоційованим з цілочисельним значенням, що представляє його вартість (вагу). Згідно з розширеною інтерпретацією тверджень Хорна матимемо:

$$\left\{ \left[ \text{Head} : - \text{Body} \right] \rightarrow \left[ \text{SC} : - \text{RC} \right] \right\} = \\ = \left[ \text{SC}^{CF_w(\text{SC})} \leftarrow \text{RC}_1^{CF_w(\text{RC}_2)} \wedge \text{RC}_2^{CF_w(\text{RC}_2)} \right],$$

де  $SC$  – (умови задоволення),

$RC_1, RC_2$  – (умови релевантності) представляють відповідні входження в правила з їх вартісними ваговими значеннями, що наведені в правій частині. 2) заголовок правила має вартісну вагу меншу або рівну ніж ту, що припускається в тілі:

$$CF_w(SC) \leq CF_w(RC_1) + CF_w(RC_2).$$

3) надлишковості використовуються з метою пошуку доведення з найменшою вартістю (ваговим значенням).

Вартості (вагові значення), що відповідно асоціюються кожному атому, правилу або рішенню співвідносяться до їх імовірності. Вагове значення атома визначається до кількості кортежів, в яких даний атом є хибним:

$$glb(|DataSet|) - glb(Rule^{Set}) = glb(\overline{Rule^{Set}}),$$

де  $|DataSet|$  є кількістю кортежів у наборі даних  $|DataSet|$ . Відповідно, в термінах імовірності вартість (вага) атома  $R$  з  $(\overline{Rule^{Set}})$  становитиме:

$$CF_w(RulesSet.R) = |DataSet| / RC(R).$$

Подібним чином кожне правило має вартість (вагу), що рівна кількості кортежів, в яких правило є хибним. Наприклад, для використання правила  $SC \leftarrow RC$  обчислення вартості (ваги) складатиме:

$$CF_w(SC \leftarrow RC) = glb(RC \wedge \overline{SC})$$

В термінах коефіцієнта впевненості для правила матимемо:

$$CF_w(SC \leftarrow RC) = glb(RC \wedge \overline{SC}) = \\ = |DataSet| \text{CLP} \cdot RC(R \wedge \overline{SC}) = \\ = |DataSet| \text{CLP} \cdot RC(\overline{SC} | RC) \text{CLP} \cdot RC(RC) = \\ = |DataSet| (1 - \text{CLP} \cdot RC(SC | RC)) \text{CLP} \cdot RC(RC) = \\ = (1 - \text{verify}(SC \leftarrow RC)) glb(RC)$$

Абдуктивним рішенням  $Sol^{AB}$  для цілі  $Goal.Case_i$  вважатимемо множину атомів  $rc_1 \wedge \dots \wedge rc_n$ , тобто  $Sol^{AB} \rightarrow Goal.Case_i$ , де  $Sol^{AB} = \{rc_1 \wedge \dots \wedge rc_n\}$ .

В даній ситуації бажаним є те, щоб вартість  $CF_w(Sol^{AB})$  була інверсно пов'язана з частотою рішення  $Sol^{AB}$ . З припущення щодо взаємного виключення серед інвертованих атомів матимемо, що:

$$|DataSet| \text{CLP} \cdot RC(\overline{Sol^{AB}}) = CF_w(Sol^{AB}),$$

$$|DataSet| \text{CLP} \cdot RC(\overline{Sol^{AB}}) =$$

$$= |DataSet| \text{CLP} \cdot RC(\overline{rc_1} \vee \dots \vee \overline{rc_n})$$

Шляхом використання гіпотези щодо взаємного виключення отримаємо, що:

$$|DataSet| \text{CLP} \cdot RC(\overline{rc_1} \vee \dots \vee \overline{rc_n}) = \\ = |DataSet| (\text{CLP} \cdot RC(\overline{rc_1}) + \dots + \text{CLP} \cdot RC(\overline{rc_n})) = \\ = CF_w(rc_1) + \dots + CF_w(rc_n) = CF_w(Sol^{AB}).$$

Такий підхід є нещільною апроксимацією імовірності  $CLP \cdot RC(\overline{Sol^{AB}})$ , оскільки не має прив'язки до проміжку  $[0, 1]$ . Проте, в даному випадку, це не є проблемою, оскільки немає реальної потреби в фактичних значеннях інвертованої частоти, а існує тільки потреба в рішеннях з низькою вартістю, що є найбільш імовірними в певних відношеннях. Таким чином, останнє твердження показує нам зв'язок між вартістю та імовірністю абдуктивного рішення. В кінцевому підсумку фреймворк узагальнюється наскільки це можливо наступним чином:

- 1) всі предикати є виводимими, навіть ті, що знаходяться в заголовках правил;
- 2) база знань може містити петлі;
- 3) всі системи обмежень  $CS$  та ієрархії обмежень  $CH$  є несуперечливими та консистентними.

Процедура виведення, що імплементується в даному підході, може обробляти обидві дані вимоги. Дана процедура була вперше розроблена як надбудова до Prolog, що дозволяє обробляти петлі на етапах пошуку і, відповідно, відтінати певні можливі виведення відповідно до вагових значень атомів, що виводяться. Залишкові процедури, такі як сортування здійснених рішень ( $Solve^{exe}$ ) і видалення деяких інших обмежень цілісності на атоми виконується в наступних кроках обробки. На рисунку 2 представлена загальна структура системи, а на рисунку 3 деталізовано особливості офлайнових обчислень при побудові бази знань. Додатковим результатом є те, що спроектований фреймворк допомагає користувачу зрозуміти причини та міркування, згідно з якими ті чи інші інстації були асоційовані з виділеним класом.



Рисунок 2 – Видобування знань в процесі кейс-солвінгу



Рисунок 3 – Локалізація процедури побудови БЗ кейсів

Слід відмітити, що всі правила видобуваються автоматично з набору даних, шляхом використання апріорного алгоритму. Єдине, що отримується на вході, – це набір даних і приклад для класифікації. Особливість також полягає в тому, що інформація, яка індукується і використовується абдуктивною системою в загальному випадку, повністю ігнорується класифікаторами на основі дерев рішень.

В загальному випадку абдуктивні системи здатні виводити певну інформацію щодо екземплярів, які необхідно класифікувати шляхом використання домену знань виведеного з набору даних. Додаткова похідна інформація (тобто та, що виведена наявними засобами) дозволяє будувати певні покращення функціонування класифікатора, тобто отримувати класифікації вищої якості.

Сутність тестування системи полягає в запуску задач класифікації на різних екземплярах, що показує, що у всіх випадках отримується однакова класифікація як з дерева рішень, так і з абдуктивного фреймворку відповідно, але з деякою відчутною різницею щодо пояснення таких результатів. Різниця полягає тільки

в способах пояснення отриманих результатів, оскільки зрозуміло, що не всі моделі представлення знань будуть такими, що, відповідно, можуть інтерпретуватися людиною.

Крім того, в ряді випадків, абдуктивна система не використовує жодного класифікаційного правила, і є можливим, що вартість (вагове значення) абдукції в деяких випадках є далеким від мінімальності. В таких випадках значенням такого класифікаційного пояснення є те, що інформація щодо об'єкту, що повинен бути класифікованим, не є достатньою для прив'язки його до певного специфікованого цільового класу. В таких випадках слід відповідно вибрати найбільш імовірний клас.

При проведенні тестування виконується розбиття множини записів на два відмінних набори даних: тренувальний набір і тестувальний набір. Різні розбиття створюються з метою уникнення неоднозначності внаслідок окремого часткового розбиття. Відповідно, експериментальні результати показують, що класифікація на основі дерева рішень і абдуктивна класифікація є еквівалентними по відношенню до точності класифікації. Причому така еквівалентність має місце, якщо не має відсутніх даних в тестовому наборі.

Абдуктивний фреймворк демонструє суттєві покращення функціональності, коли виконується додавання нової інформації, наприклад, при видобуванні асоціативного правила в базі знань. Можна бачити, що додавання асоціативних правил до бази знань абдуктивної системи робить таку систему здатною підвищувати точність класифікації, шляхом отримання результатів досить близьких до класифікатора на основі дерева рішень.

Нехай  $AnswerSet$  множина імен атрибутів і  $\{Case_i, Class\}_i$  множина класів (можливих класифікацій). Припустимо, що кожній множині рішень може бути співставлене присвоєння скінченної послідовності значень. Відповідно, приклад рішення  $Solve^{sample}$  можна розглядати як множину пар атрибут/значення виду:

$$Solve^{solution} = \{rc_1.value = v_1, \dots, rc_n.value = v_n\},$$

де  $rc_i$  є іменем і-того атрибуту.

**Означення 6.** Мережею обмежень  $CN$  над  $AnswerSet$  та  $\{Case_i, Class\}_i$  є множиною обмежень така, що:

- 1) кожен не листовий вузол помічений атрибуту  $rc \in AnswerSet$ ;
- 2) кожен листовий вузол помічений класом  $CF_w \in Case.Class$ ;
- 3) кожна вітка помічена значенням  $v \in V$ ;
- 4) значення, що помічають всі вітки, які виходять із заданого вузла, є відмінними;
- 5) мітки шляху теж є відмінними;
- 6) суміжні системи обмежень є несуперечливими.

Відповідно, остання вимога формалізує той факт, що на кожному шляху тільки один тест може бути виконаний для кожного атрибута.



Дана задача переадресується на виконання прямому алгоритму, що включає наступні кроки:

1) для кожної обширної множини входжень  $LE$  знайти всі не порожні підмножини;

2) для кожної такої підмножини  $LE^{sub} \subset LE$  з  $LE$  вивести правило  $LE \rightarrow LE^{sub}$  якщо і тільки, якщо:

$$CF = \frac{SF(LE)}{SF(LE^{sub})} \geq CF^{min} = CF^{thres},$$

де  $CF$  – коефіцієнт впевненості (довіри);

$SF$  – коефіцієнт підтримки;

$CF^{min}$  – мінімальне значення коефіцієнта впевненості;

$CF^{thres}$  – порогове значення коефіцієнта впевненості.

Дамо формальне визначення проблеми видобування наборів частих маркованих входжень.

**Означення 7.** Множиною входжень вважатимемо скінчену множину маркерів, що позначається через  $ES$ . Елементи  $ES$  позначатимемо через  $es, es', es_1, \dots$  і посилатися на них як на елементи маркованих входжень.

**Означення 8.** Множина елементів розглядається як множина  $ES$ . Множини елементів будемо позначати через  $ES, ES', \dots$ .

**Означення 9.** Множину елементів  $ES$  будемо називати  $|n|$ -множиною елементів, якщо потужність  $|ES| = n$ , де  $|ES|$  позначає потужність  $ES$ .

**Означення 10.** Транзакцією  $CN_T$  будемо вважати деяку множину маркованих елементів і позначатимемо її через  $CN_T, CN'_T$ .

**Означення 11.** Набором даних  $DataSet$  вважатимемо мультимножину транзакцій. Для заданого набору даних  $DataSet$  максимальною довжиною транзакцій  $DataSet$  будемо вважати:

$$l_{DataSet} = \max\{CN_T | CN_T \in DataSet\}.$$

Визначимо формально проблему видобування частих наборів маркованих даних.

**Означення 12.** Нехай  $DataSet$  є набором даних і  $ES$  є набором маркованих входжень (елементів). Тоді,  $ES$  будемо вважати частим набором маркованих входжень (множиною маркованих елементів) по відношенню до  $DataSet$  і з коефіцієнтом підтримки  $SF$ , де  $0 < SF \leq 1$ , якщо:

$$\{|CN_T \in DataSet | ES \subseteq CN_T\} \geq SF |DataSet|.$$

Нехай  $nSet_n(SF, DataSet)$  є множиною всіх  $|n|$ -наборів (множин) маркованих елементів, що є частими входженнями по відношенню до  $SF$  та  $DataSet$ . Тоді, шукана проблема означається як задача визначення  $nSet_n(SF, DataSet)$  для кожного  $n$ , такого, що  $0 < n \leq l_{DataSet}$ .

Означимо формально проблему агрегованих маркерів.

**Означення 13.** Позначатимемо деяку скінчену множину графічних маркерів через  $L^{Set}$ . Відповідно,  $L^{Set}$  буде позначатись через  $l, l', l_1$  і буде посилатися на них як на виділені графічні маркери.

**Означення 14.** Потокотом маркерів будемо вважати послідовність  $seq^l = l_1, \dots, l_n$ , такі, що  $l_i \in L^{Set}$  для кожного  $l \leq i \leq n$ . Довжину потоку  $n$  будемо позначати як модуль  $seq^l$ .

**Означення 15.** Для заданого потоку  $seq^l$  і маркеру  $l$ , через  $freq(l^l)$  позначатимемо кількість входжень  $l$  в  $seq^l$ .

**Означення 16.** Нехай  $L^{Set}$  є множиною маркерів, а  $seq^l$  є потоком маркерів і  $num$  є дійсним числом, таким, що  $0 < num \leq l$ . Тоді проблему агрегації маркерів означимо як задачу визначення підмножини  $L^{Set}(num, seq^l)$ , що визначається як:

$$L^{Set}(num, seq^l) = \{l \in L^{Set} | freq(l) > num | seq^l\}.$$

Якщо  $l$  належить до  $L^{Set}(num, seq^l)$ , то можна стверджувати, що  $l$  є видом агрегованого маркера по відношенню до  $L^{Set}$ ,  $seq^l$  та  $num$ .

В даному контексті важливо відмітити, що в конкретному застосуванні проблем видобування частих наборів даних і агрегованих маркерів, вхідна множина  $DataSet$  та  $seq^l$  є, як правило, великими і можуть бути зчитані тільки послідовно, тобто, відповідно до ідентифікаторів транзакції в першому випадку і до порядку послідовності в другому.

Функціональність такого представлення є наступною: для заданого потоку  $seq^l$  і дійсного числа  $num$ , що називається пороговим значенням, алгоритм вимагає одного проходу через вхідний потік з метою пошуку надмножини шуканих  $L^{Set}(num, seq^l)$ . Може бути також виконана друге проходження для знаходження другого значення  $L^{Set}(num, seq^l)$  із збереженням тих самих характеристик ефективності.

Основна ідея такого рішення полягає у прив'язці до кожного  $|n_0|$ -набору елементів певного запиту і подальшої побудови відповідного підходящого потоку запитів  $seq^l_{DataSet}$ , починаючи з заданого набору даних  $DataSet$ , таким чином, що проблема визначення  $nSet_n(SF, DataSet)$  перетворюється в проблему визначення  $L^{Set}(num, seq^l_{DataSet})$ , де  $num$  є функцією від  $num$ ,  $l_{DataSet}$  та  $n_0$ . Після того, як означено таке перетворення, можна використати довільний існуючий алгоритм для проблеми агрегованих маркерів з метою вирішення вихідної проблеми видобування частих входжень наборів даних. Зокрема, можна використати алгоритми, які утримують кількість прохо-

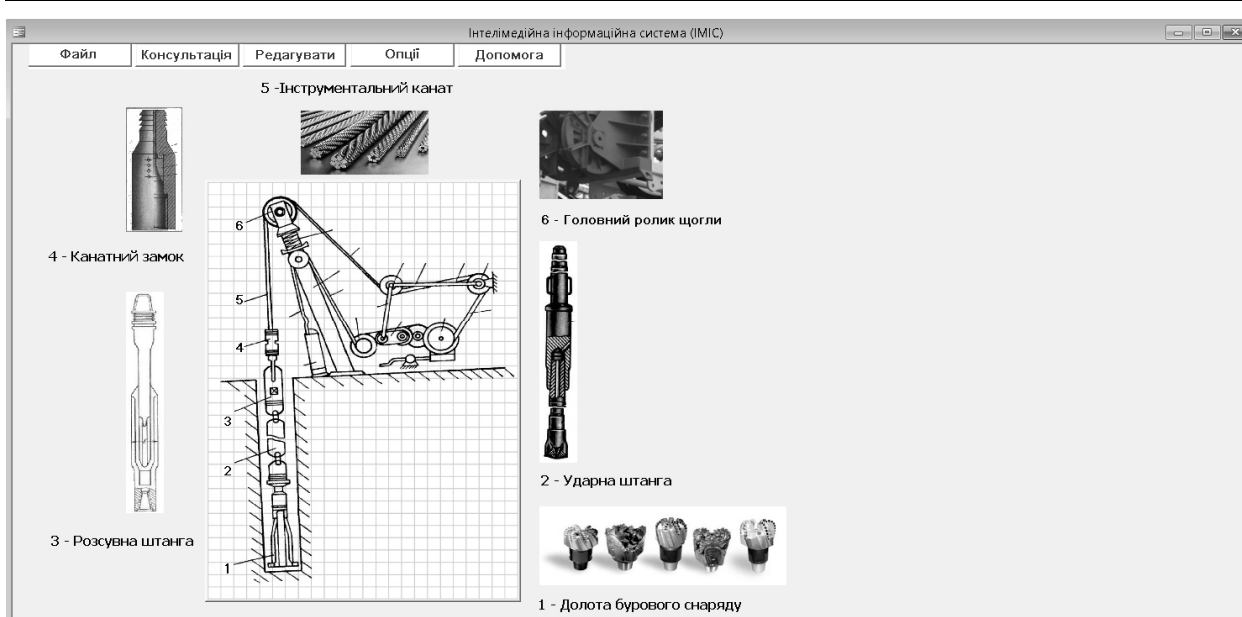


Рисунок 4 – Схематичне зображення базового бурового обладнання процесу буріння з введеними ключовими маркерами

джен через набір даних настільки малим, наскільки це можливо.

Для заданої скінченої множини  $SEQ$  і натурального числа  $n_0$  позначимо через  $SEQ^{n_0}$  множину всіх підмножин  $L^{Set} \subseteq SEQ$  таких, що  $|L^{Set}| = n_0$ . Крім того, для двох заданих послідовностей  $seq^1 = \langle x_1, \dots, x_n \rangle$  та  $seq^2 = \langle x'_1, \dots, x'_m \rangle$  позначимо через  $seq :: seq^2$  послідовність  $\langle x_1, \dots, x_n, x'_1, \dots, x'_m \rangle$ . Для цього визначимо перетворення, яке для заданої проблеми видобування частин входжень наборів даних буде відповідно до проблеми агрегованих маркерів.

Знання, виділені з таких маркованих входжень, дозволяють імітувати власний процес міркувань при прийнятті рішень. Вибрані в якості засобу розробки оболонки експертних систем включають унікальні методи зберігання мультимедійних даних з можливістю їх зіставлення. Технологічно система зберігає дані про об'єкти в структурах, що описується означуваними схемами (*Definitions Schemas*). Для даного випадку це забезпечує дуже зручний спосіб зберігання та вилучення даних. Типи об'єктів цієї системи мають справу з буровим обладнанням, технологічними операціями та режимами буріння. Дані схеми забезпечують способи визначення характеристик об'єктів, в тому числі і по відношенню до інших об'єктів за допомогою технології комірок (семантичний аналог фреймових слотів). Кожна комірка являє собою одну або більше ознак чи відношень, які описують поточний нафтогазовий об'єкт.

Після створення функціональної бази знань необхідно забезпечити відображення інформації у відповідній формі з використанням засобів мультимедійного інтерфейсу. ускладнює процес параметризації представлення на

основі системи ключових маркерів, а, з другого боку, – наявними надміру абстрактними узагальненнями, що не дозволяють параметризацію взагалі

**Означення 17.** Нехай  $iSet$  є набором маркованих елементів, а  $DataSet$  є набором даних і  $n_0$  є натуральним числом, таким, що  $n_0 \leq l_{DataSet}$ . Тоді:

1) алфавіт  $L_{iSet}^{Set}$  визначається як множина  $iSet^{n_0}$ , причому кожна множина в  $iSet^{n_0}$  є символом в алфавіті;

2) для кожного  $CN_T \in DataSet$  потік асоційований з  $CN_T$  є, відповідно, послідовністю  $seq^1_{CN_T} = \langle iSet_1, \dots, iSet_{n_{CN_T}} \rangle$ , таким, що  $\{iSet_1, \dots, iSet_{n_{CN_T}}\} = CN_T^{n_0}$  і кожне  $iSet_j \in CN_T^{n_0}$  зустрічається в  $seq^1_{CN_T}$  точно один раз;

3) якщо  $DataSet = \{CN_{T_1}, \dots, CN_{T_n}\}$ , тоді потік  $seq^1_{DataSet}$  асоційований з  $DataSet$  є послідовністю:

$$seq^1 = seq^1_{CN_{T_1}} :: seq^1_{CN_{T_2}} :: \dots :: seq^1_{CN_{T_n}}$$

Слід відмітити, що в наведеному означенні не означається потік асоційований з транзакціями. Подібним чином можна говорити про потік асоційований з набором даних  $DataSet$ . Дійсно для заданої транзакції  $CN_{T_i}$  існує багато

способів побудови потоку  $seq^1_{CN_{T_i}}$ , що відповідає йому і, відповідно, може існувати багато способів побудови  $seq^1_{DataSet}$ . Тому, фактично вибір  $seq^1_{DataSet}$  є не релевантним, з точки зору коректності такого перетворення.

Покажемо, що проблема видобування частиних наборів маркованих елементів  $nSet_n(SF, DataSet)$  може бути відображена в проблему агрегованих маркерів  $L^{Set}(num, seq^1_{DataSet})$ , де  $seq^1_{DataSet}$  є довільним потоком, побудованим згідно з попереднім означенням, і  $num$  є відповідною функцією для  $SF$ ,  $n_0$  та  $l_{DataSet}$ .

Нехай  $iSet$  – набір маркованих елементів,  $DataSet$  – набір даних,  $n_0$  – натуральне число, таке, що  $n_0 \leq l_{DataSet}$  і  $SF$  – дійсне число, таке, що  $0 < SF \leq 1$ . Нехай  $L^{Set^{iSet}}$  та  $seq^1_{DataSet}$  є відповідно алфавітом і послідовністю маркерів згідно попереднього означення. Нехай також  $Solve^{Solution} = (l_{DataSet}, n_0)$ .

Якщо набір елементів  $iSet$  є частим  $n_0$ -набором маркованих елементів по відношенню до  $SF$  і  $DataSet$ , тоді  $iSet$  є агрегованим маркером по відношенню до  $L^{Set^{iSet}}$ ,  $seq^1_{DataSet}$  та  $num = SF / Solve^{Solution}$ . Тоді

$$iSet \in nSet_n(SF, DataSet) \Rightarrow iSet \in L^{Set}(num, seq^1_{DataSet})$$

Нехай  $|DataSet| = k$  та  $DataSet = \{CN_{T_1}, \dots, CN_{T_n}\}$  матимемо:

$$1) |seq^1_{DataSet}| = \sum_{i=1}^k \binom{|CN_{T_i}|}{n_0} \leq \sum_{i=1}^k \binom{i_{DataSet}}{n_0} = \overline{Solve^{Solution}}, k;$$

2) згідно зі способом побудови  $seq^1_{DataSet}$  матимемо

$$|\{CN_T \in DataSet \mid iSet \subseteq CN_T\}| = freq_{DataSet}(iSet).$$

Згідно з означенням  $nSet_n(SF, DataSet)$  можна отримати, що  $iSet \in nSet_n(SF, DataSet)$ .

На основі зазначеного вище  $|\{CN_T \in DataSet \mid iSet \subseteq CN_T\}| = (CF, k)$ .

Згідно з алгебраїчним способом побудови матимемо, що:

$$freq_{seq_{DataSet}}(iSet) \geq (CF, k).$$

Згідно з означенням для  $num$  матимемо:

$$freq_{DataSet}(iSet) \geq CF / Solve^{Solution} \geq \left\lfloor k, \overline{Solve^{Solution}} \right\rfloor$$

Згідно з означенням для  $L^{Set}(num, seq^1_{DataSet})$  матимемо:

$$freq_{DataSet}(iSet) \geq num |seq^1_{DataSet}|, \\ iSet \in L^{Set}(num, seq^1_{DataSet})$$

Дане твердження дозволяє отримати нову техніку знаходження частиних  $n_0$ -наборів маркованих елементів, що може бути описана наступним чином:

1) побудувати потік  $seq^1_{DataSet}$ , що відповідає  $DataSet$ ;

2) знайти множину агрегованих маркерів  $L^{Set}(num, seq^1_{DataSet})$ .

Таким чином, отримана реалізація проекту інтелектуальної інформаційної системи дозволяє

поєднувати представлення знань у формі обмежень та правил для контролю доменних представлень предметної області із введеними послідовностями маркованих мультимедійних входжень.

## Висновки

В запропонованому дослідженні основний результат полягає у використанні відмінних метрик на основі значущих графічних маркерів, що дозволяють встановлювати логічні зв'язки між описаними кейсами для ситуацій підтримки прийняття рішень. Такий вид зв'язків є важливим в області абдуктивних міркувань в термінах правил та обмежень, оскільки вони можуть ефективно виражати залежності між об'єктами та подіями нафтогазової предметної області, що описуються мультимедійними входженнями в формі баз даних та баз знань. Таким чином, новизна застосування полягає у використанні абдуктивних міркувань в процесі класифікації мультимедійних входжень в формі растрових зображень, що дозволяє суттєво підвищити загальну ефективність інтелектуальної системи, що працює з неповними, неточними та слабкоструктурованими знаннями предметної області. Побудовані представлення включають правила виду «умова релевантності – умова задоволення» та представлення у формі множин, систем та ієрархій обмежень як двох базових концепцій для побудови абдуктивного фреймворку на основі правил, обмежень та абдуктивних логічних виведень. В діалозі з користувачем інтелектуальна система оперує з множиною правил, для яких встановлюються коефіцієнти підтримки та коефіцієнти довіри з додатково встановленими пороговими значеннями.

Подальші дослідження абдуктивного фреймворку з обмеженнями та правилами будуть спрямовані на адаптацію методології класифікації мультимедійних входжень нафтогазової предметної області.

## Література

- 1 Mancarella P. An abductive proof procedure handling active rules / P. Mancarella and G. Terreni // AI\*IA 2003: Advances in Artificial Intelligence / A.Cappelli and F.Turini, editors. – SV of LNAI, 2003. – P. 105-117.
- 2 Kakas A.C. Abductive concept learning / Antonis C. Kakas, F. Riguzzi // New Generation Computing, 2000. – V. 18(3). – P. 243-294.
- 3 Kakas A.C. Aclp: Abductive constraint logic programming / A.C. Kakas, A. Michael, and C. Mourlas // Journal of Logic Programming, 2000. – V. 44 (1-3). – P. 129-177.
- 4 Вагин В.Н. Абдукция в задачах планирования работ в сложных объектах / В.Н. Вагин, К.Ю. Хотимчук // Искусственный интеллект и принятие решений. – М.: Ленанд, 2011. – Т.1. – С. 3-13.

5 Eskilson, J. & Carlsson, M., SICStus MT - Multithreaded Execution Environment for SICStus Prolog. Konstantinos F. Sagonas, ed. Implementation Technology for Programming Languages based on Logic, 1998, pp. 59-71.

6 Mohammed J. Zaki. Theoretical foundations of association rules / Mohammed J. Zaki and Mitsunori Ogihara; in Proceedings of 3<sup>rd</sup> SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98), Seattle, Washington, 1998. – P. 10-17.

7 Юрчишин В.М. Інформаційне моделювання нафтогазових об'єктів: монографія / В.М. Юрчишин, В.І. Шекета, О.В. Юрчишин. – Івано-Франківськ: Вид-во Івано-Франківського нац. техн. ун-ту нафти і газу, 2010 – 196 с.

8 Шекета В.І. Застосування абдукції в задачах класифікації даних про нафтогазові об'єкти / В.І. Шекета, М.М. Демчина, Л.М. Гобир // Нафтогазова енергетика. – 2014. – № 2(22). – С. 86-97.

9 Wenmin Li. SMAR: Accurate and efficient classification based on multiple class-association rules / Wenmin Li, Jiawei Han, and Jian Pei. – In ICDM, 2001. – P. 369-376.

10 Dubois D. Automated reasoning using possibilistic logic: Semantics, belief revision, and variable certainty weights / D. Dubois, J. Lang, H. Prade. – Knowledge and Data Engineering. – 1994. – Feb. – P. 64-71.

11 Jiawei Han. Data Mining: Concepts and Techniques / Jiawei Han and Micheline Kamber. – Morgan-Kaufman, 2000. – 28 p.

12 Liu, B.; Hsu, W. & Ma, Y., Integrating classification and association rule mining. Proceedings of the 4th international conference on Knowledge Discovery and Data mining (KDD'98), AAAI Press, 1998, pp. 80-86.

13 Quinlan J.R. Improved use of continuous attributes in C4.5 / Quinlan J.R. // Journal of Artificial Intelligence Research. – 1996. – V.4. – P. 77-90.

14 Michell T. Machine Learning / T. Michell. – McGraw Hill, 1997. – 414 p.

*Стаття надійшла до редакційної колегії*

*25.04.16*

*Рекомендована до друку*

*професором Горбійчуком М.І.*

*(ІФНТУНГ, м. Івано-Франківськ)*

*професором Петришиним Л.Б.*

*(Прикарпатський національний університет,  
ім. В. Стефаника, м. Івано-Франківськ)*