Софийски университет „Свети Климент Охридски„
Факултет по математика и информатика

РАЗСЪЖДЕНИЯ В МНОГОКОНТЕКСТНИ СИСТЕМИ

Дипломна работа
на

Атанас Георгиев Георгиев
специализация „Изкуствен интелект„
E-mail: fn41270@sparc10.fmi.uni-sofia.bg

Научен ръководител:

Христо Дичев
Институт по информационни технологии - БАН
E-mail: cdichev@iinf.bg

София,
29 май 1996 г.

Абстракт

В тази дипломна работа е направен опит за формализация на някои аспекти от човешките разсъждения - локалност, схематичност, локалност на противоречието, способности за обобщаване и конкретизиране (специализиране). Процесът на разсъждения се разбива на последователни фрагменти от локални разсъждения. Ключовата дума тук е „контекст„. На контекста се гледа като на теория, породена от множество аксиоми и затворена относно множество правила за извод. Многоконтекстната система представлява множество от контексти и правила за прехвърляне на процеса на разсъждение от един контекст в друг. Разгледани са две различни множества от контекстни правила за извод и е доказана тяхната еквивалентност. Доказана е и независимостта на всеки контекст от наличието или отсъствието на противоречиви контексти в многоконтекстната система. Дефинирани са няколко операции за сформиране на нови контексти на базата на първоначалните и е илюстрирано тяхното приложение с примери. Направена е и примерна програмна реализация, която демонстрира някои предимства и проблеми свързани с практическото приложение на разгледания модел. Предначертани са и някои насоки за бъдещи негови разширения.

Съдържание

Глава 1

Увод

1.1 Въведение в проблема

Както знаем, в основите на дисциплината изкуствен интелект лежи открай-колешния стремеж на човека да създаде свое изкуствено подобие, притежаващо съответното ниво на интелигентност. В своите разсъждения обаче хората твърде често боравят с абстрактни и размити понятия. От друга страна, в изчислителната техника всичко е точно и строго - тя не може да работи с абстрактни понятия. Затова е налице необходимост от формализация (моделиране) на абстрактните понятия с точни термини.

Въпреки че са предложени много варианти за архитектура на символните системи за изкуствен интелект, повечето от тях се основават на една и съща концепция: Системата разполага с една база знания и множество процедури, които оперирайки с тези знания постигат донякъде интелигентно поведение. Входната информация се записва в базата знания, като преди това се превежда на нейния език.

Според тенденциите доминиращи в момента, базата знания използва декларативно кодиране, а процедурите са дедуктивни. Системата има определена област на компетентност (проблемна област) и всъщност крайната цел на дисциплината изкуствен интелект е създаването на система, чиято област на компетентност е сравнима с тази на хората. Базата знания съдържа главно познания за областта на компетентност и по-рядко - мета-знания за това как да използва своите познания.

Базата знания представлява множество изречения, изказани с определени термини (речник). Всяко изречение е едно твърдение за областта на компетентност. Неговият смисъл и истинност не зависят от наличието или отсъствието на други изречения. Такива твърдения се наричат *универсални*. Всички възможни (и предвидени в модела) фактори, от които зависи твърдението, се изразяват явно в него.

Това е коренно различно от ситуациите в ежедневието на хората. Изказванията на естествен език никога не са универсални. В тях винаги са вложени огромен брой предположения и тяхното значение зависи изключително много от ситуацията, в която са направени. Факторите, които имат значение, са не само предишните изказвания, но и целите на хората, които разговарят, социалната и културната обстановка и т.н.

Тъй като смисълът и верността на никое изречение не зависят от другите изречения, това изречение не може да се осланя на никакви предположения. Това важи с пълна сила за системите, основаващи се на логиката от първи ред. За системи, чиято първостепенна задача е да гарантират точност в аргументацията на своите разсъждения, това е много желано качество. Обаче за системите, чиято задача е да правят изводи за неща от ежедневието на хората, това се оказва по-скоро проблем.

Друг интересен аспект на базите знания е тяхната хомогенност. Тя се изразява по няколко начина:

• Докато много бази знания структурират проблемната област (например с таксономични йерархии), те рядко структурират познанията си за тази област. Например, в цялата база знания се използва един и същ речник.

• Базата знания съдържа единствена теория за света. За да има полза от нея, тя трябва да се поддържа непротиворечива.

• Наличието на единствена теория за света предполага, че тази теория трябва е независима от някои специфични проблеми, т.е. нейното представяне не трябва да е специално насочено към решаването на определени конкретни проблеми.

Колкото и да е добър, този модел си има своите недостатъци. На практика повечето изследвания в областта на изкуствения интелект през последните 30 години са насочени към неговото подобряване. Проблемите, които възникват с него, са свързани главно с речника на базата знания и с нейната теория за света.

Едно от нещата, които трябва да направим за да изградим една база знания, е да изберем речник, с чиито термини да кодираме познанията. Този речник трябва да покрива всички явления, които смятаме, че имат място в разглежданата област на компетентност. Това означава, че със самия избор на речник, някои явления може да се изключат от базата знания. Това съответства на предположението че тези явления не съществуват или не са важни. Тяхното изключване може да стане както съзнателно, така и случайно. Почти неизбежно е в процеса на представяне и кодиране на знанията, част от проблемната област да бъде пропусната по невнимание.

Естествено, когато се натъкнем на неизразими с речника явления, бихме могли да го прецизираме. Това обаче ще го направи в повечето случаи излишно подробен и неудобен за използване. Постигането на някакъв баланс между прецизността на речника и удобното му използване не изглежда много добро решение. Освен това, при по-внимателно търсене, винаги можем да открием още явления, които са неизразими и с новия речник. На даден етап просто трябва да се спрем на окончателна версия на речника и да продължим с представянето на знанията.

Първата стъпка за постигането на съвместимост при разширяването на речника е да се отчете факта, че с избора на речник ние правим определени предположения.

Аналогично, с теорията, която кодираме, също правим определени предположения. Може да имаме аксиоми, които да твърдят, че при покупка, купувачът обикновено се нуждае от закупения предмет, продавачът се нуждае от пари, след покупката купувачът е собственик на предмета и т.н. Въпреки, че това са съвсем основни правила при покупка, те също правят предположения. Те предполагат, например, рационалност в действията на купувача и продавача. Ако купувачът прави покупката без определена причина или с цел да си навлече финансови загуби, тогава почти никоя от тези аксиоми няма да се отнася до него - неговият случай е извън обхвата на теорията.

И ограниченията в речника, и предположенията на теорията са проблеми, с които се сблъскваме, когато изграждаме една база знания. Стандартният подход е да създадем един много богат речник (с който да могат да се опишат всички явления, на които можем да се натъкнем) и да открием и изразим явно всички предположения, на които почива теорията ни. Тази практика не е единственото решение на проблема. Ето някои нейни недостатъци:

• Неизбежно ще има някакви ограничения в речника и предположения на теорията. Когато тези предположения бъдат открити, при разширяването на базата знания за да обхване новите явления може да се наложи съществена преработка на голяма част от нея. Това не е желателно - нуждаем се от по-рационален начин за разширяване на базата знания.

• Речник и теория, които имат много широка проблемна област е много вероятно да са тромави и обременителни. Това е нежелателно както от гледна точка на записа на аксиомите, така и от гледна точка на процеса на извод.

Целта на настоящата дипломна работа е да представи един логически апарат за формализация на човешките разсъждения, който избягва споменатите по-горе недостатъци и в който са застъпени някои съществени аспекти на човешкото мислене: локалност на разсъжденията, схематичност на разсъжденията, локалност на противоречието, обобщаване и специализиране на ситуации. В основата на този апарат лежи понятието контекст.

Какво е това контекст? Всеки от нас си има някаква интуитивна представа какво представлява контекстът. Самата дума „контекст„ е много широко употребявана, както самостоятелно, така и в утвърдени фразеологични изрази, като „в този контекст„, „извън разглеждания контекст„ и т.н. И въпреки това, когато се стигне до дефиниция на това понятие, много от нас биха се затруднили да я дадат, а ако успеят, то едва ли ще има две еднакви. Както John McCarthy отбелязва в [7], контекстът е абстрактно понятие, което трудно се поддава на дефиниция.

Това обаче не означава, че разсъжденията в контекст не се поддават на формализация. Ясно е например, че контекстът може да се възприеме като подмножество на нашите знания за света. Когато се изправим пред един проблем, ние не го атакуваме директно с всичките си познания, а само с част от тях, която считаме за имаща най-голямо отношение към случая. При това, както вече споменахме, голяма част от тези знания са наши предположения за ситуацията, а не непосредствени наблюдения.

Какво правим, например, когато се качим в колата си и искаме да потеглим, но тя не може да запали? Мислим си за възможните причини за нейното поведение - дали има бензин в двигателя, дали не е спаднал зарядът на акумулатора, дали и преди е правила така и т.н. Или, ако бързаме за някъде, търсим друг начин, по който да стигнем дотам. В такива случаи обикновено не се сещаме за миграциите на пойните птици през есента или за черните дупки в центъра на галактиката.

Това е едно много съществено свойство на човешките разсъждения, което тук ще наричаме *локалност*. То има пряко отношение към ефективността на разсъжденията. През своя живот човек натрупва толкова много познания, че ако използваше всичките за разрешаване на всеки възникнал проблем, той просто нямаше да може да реагира адекватно.

Друга важна характеристика на човешките разсъждения е тяхната *схематичност*. При решаването на повечето проблеми човек съставя не една, а няколко вериги на разсъждения, които са свързани помежду си в цялостна схема на разсъжденията. При това отделните вериги са локални разсъждения в един контекст и на техен фон преходите от един контекст в друг са редки. Това от една страна се дължи на локалността на разсъжденията, а от друга - на инертността на човешкото мислене.

Ето един интересен пример за инертността, а оттам и за локалността на човешките разсъждения: В едно издание на популярното телевизионно състезание „Минута е много„ единият от съзтезателите беше накаран да произнесе много пъти бързо сричката „ню„ - първо веднъж, след това два пъти и така до пет пъти. След това трябваше бързо да каже коя е столицата на САЩ. Той много добре знаеше, че това е Вашингтон, но в резултат на инерцията, получил от непрекъснатото повтаряне на сричката „ню„, каза „Ню Йорк„.

Друг подобен случай е експеримент на група руски учени, проведен преди няколко години. На една черна дъска са написани две групи думи, едната - на руски, а другата - на немски. Думите са съставени изцяло от букви, които ги има и в кирилицата, и в латиницата. Участниците в експеримента владеят отлично и двата езика. На всеки от тях се показват последователно думи в произволен ред и той трябва да ги произнесе. Резултатът бе, че след първите няколко думи, всички участници загубваха ориентация и започваха да бъркат руски думи с немски и обратно. Думата *"mama"*, например, която означава „мама„ на немски, беше често произнасяна като „тата„, която няма смисъл на руски.

Голяма част от комичните ситуации в ежедневието се дължат също на неспособността на човека, бързо да сменя контекста на своето мислене. Достатъчно е да си припомним любимите детски анимационни филми, на които и сега се смеем: „Том и Джери„, „Ну погоди!„, „Патето Яки„ и др. Общ подход на авторите им е в началото да ни въведат в контекста „отрицателният герой гони положителния„, след което в различните епизоди специализират този контекст със ситуации при които „все нещо се случва„ и положителният герой се изплъзва. Комичността идва от стръвта, с която отрицателният герой преследва положителния. Той до такава степен се е вглъбил в контекста на своята роля, че не може да излезе от него и да вземе предвид елементарни неща, като лопатата, поставена на пътя, хипопотама насреща и т.н.

Третата особеност на човешкото мислене, на която се отделя внимание в тази дипломна работа, е *локалността на противоречието*. Какво прави човек, когато някоя верига разсъждения го доведе до противоречие? Значи ли, че от този момент нататък, той ще приеме, че всичко е вярно, както повелява логическата теорема $a \wedge \neg a \rightarrow b$? От опит знаем, че това не е така. В такива ситуации обикновено човек се съмнява в някое от априорните си знания. Дори и да не открие кое е то, той запазва способността си да разсъждава трезво при останалите ситуации. Човекът просто приема, че познанията му в дадената област са противоречиви и следващия път преди да се допита до тях ще има едно наум.

Интерес представляват и начините за сформиране на контексти. Ясно е, че човек не пази в главата си пълно описание на познанията си за всяка ситуация. Това не би могло да бъде, тъй като ситуациите са безброй. За някои от тях човек помни факти, които е натрупал от личен опит. Останалите ситуации той ги свежда до познати чрез поредици от разсъждения.

Два основни вида такива разсъждения са *обобщаване* и *специализиране*. Ако имаме описанието на една ситуация и премахнем част от него, ще получим описание на по-обща ситуация, защото освен първоначалната, на него ще отговарят и други. Това е *обобщаване*. Например, ако изходната ситуация е „карам червена кола„, то „карам кола„ е нейно обобщение. То включва както ситуацията „карам червена кола„, така и „карам синя кола„, „карам зелена кола„ и т.н. За най-обща може да се приеме ситуацията, за която не знаем нищо - нея няма как да я обобщим повече.

Обратно, специализирането представлява конкретизиране на дадена ситуация. Например, „карам червена спортна кола„ е специализация на „карам червена кола„. Очевидно една ситуация може да се специализира до безкрайност, но на практика е достатъчно тя да се специализира само до степен, при която дава възможност да се реши конкретния проблем. В тази дипломна работа обобщаването и специализирането се разглеждат като основни начини за сформиране на нови контексти.

Изразените дотук аргументи се базират повече на интуитивни схващания, отколкото на строги научни доказателства. Това обаче не намалява тяхното значение. Ето защо те са залегнали в основата на настоящата работа.

1.2 Съществуващи разработки

По въпроса за формализиране на разсъжденията в контекст се работи главно на три места: Института за научни и технически изследвания в Геноа, Италия, Университета в Станфорд, САЩ и Института по информационни технологии, София.

Идеята за използване на контекстите, като средство за представяне на знания, е предложена за първи път от проф. John McCarthy, който е ръководител на групата по формални разсъждения (Formal reasoning group) в университета в Станфорд. Един от членовете на тази група R. V. Guha защитава докторската си дисертация [6] в Станфорд, която McCarthy използва в по-нататъшните си изследвания [7]. Неговите усилия са насочени в областта на разсъжденията на т.нар. „интелигентни агенти„ (Intelligent agents) и представянето на такива концепции, като познания на агента за собственото си вътрешно състояние (самопознание), различни степени на увереност в знанията и т.н.

Италианците Giunchiglia и Serafini разглеждат модел с многоконтекстни системи, които наричат многоезични [4, 5]. Те дефинират многоезичните системи като формални аксиоматични системи, използващи различни езици, и построяват многоезична йерархична логика, която предлагат като алтернатива на модалната логика. По-нататък разглеждат приложимостта на многоезичните системи в области, в които се използва модалната логика.

Настоящата дипломна работа е естествено продължение на работите на Дичев [1, 2]. В [1] е представен езикът UCD-Prolog - едно разширение на езика Prolog, което дава възможност за разделяне на базата знания на относително самостоятелни сегменти. В последствие възниква идеята за формализация на разсъжденията в контекст и в [2] е мотивирана необходимостта и са предначертани насоките за една такава формализация.

Основната разлика в описания тук подход спрямо разработките в Станфорд и Генуа е в отношението към локалността на разсъжденията и противоречието. В дисертацията си R. V. Guha предлага голямо количество правила за смяна на контекста, които са интуитивно коректни, но благодарение на тях възникнало противоречие в един контекст, автоматично довежда до противоречие във всички контексти. В предложената от Giunchiglia и Serafini многоезична йерархична логика пък имената на контекстите не са част от езика. Това води до невъзможност за комуникация между контексти, които не са свързани по права линия в дървото на йерархия. С наложеното в дипломната работа единствено правило за смяна на примитивни контексти се дава едно възможно решение, при което се постига локалност на противоречията, а локалността на разсъжденията не се постига за сметка на ограничаване на свободата на общуване между контекстите.

1.3 Структура на изложението

Изложението на дипломната работа е структурино по следния начин:

В глава 2 са дадени дефинициите на основните понятия - какво ще разбираме под термина „контекст„, какво представлява една многоконтекстна система. В раздел 2.1 са дадени дефиниции, свързани със синтаксиса и синтактичната изводимост в многоконтекстните системи. В 2.2 са описани основните предположения и ограничения, които използваме при разглежданията по-нататък, а раздел 2.3 съдържа дефиниции, свързани със семантиката и семантичната изводимост в многоконтекстните системи.

Глава 3 съдържа дефиниции, теореми и факти от класическата логика, които се използват наготово в дипломната работа. Раздел 3.1 е посветен на класическото предикатно смятане (Functional Calculus), а в раздел 3.2 е описана логиката NK, която въвежда Генцен в своята докторска дисертация [3].

Основните резултати са описани в глава 4. В раздел 4.1 са дефинирани контексти с правила за извод, съответстващи на предикатното смятане и на логиката NK и се доказва тяхната еквивалентност. В раздел 4.2 се разглеждат въпроси, касаещи изводимостта в многоконтекстните системи. Тук се доказва теоремата за локалност на противоречието, както и различни свойства на изводимостта в няколко контекста. Коректността на синтактичния извод, както и въпросът за неговата пълнота е разгледан в раздел 4.3.

В глава 5 се разглеждат начините за формиране на нови контексти. В раздел 5.1 са дадени разширените дефиниции за синтаксис и семантика, включващи и съставни контексти. В раздел 5.2 е направен преглед на свойствата на многоконтекстните системи със съставни контексти, а в раздел 5.3 са описани операциите затваряне (капсулиране), обобщаване и специализиране на контекст, както и различни техни свойства.

Глава 6 е повече демонстративна. В началото й е описан създадения метаинтерпретатор на езика Prolog за работа с контексти, а след това са показани неговите възможности.

Изводите и заключителните бележки са изложени в глава 7.

Глава 2

Основни дефиниции, предположения и свойства

2.1 Синтаксис

След като ще работим с много контексти, едно важно изискване към синтаксиса е той да дава ясна представа в кой контекст се водят разсъжденията. Тъй като позволяваме една и съща формула да има различно значение в различните контексти, в нейната нотация трябва задължително да присъства и контекстът, в който я разглеждаме. За целта въвеждаме записа $\langle c, \alpha \rangle$, в който α е формула, а c е разглежданият контекст.

Както вече споменахме, един контекст от нашите разсъждения съдържа част от познанията ни за света, която има отношение по разглеждания случай. Това означава, че е подходящо да представим контекстите под формата на множество от формули. В това множество можем да разграничим два вида формули. Първият вид формули са такива, които изразяват познания за света, т.е. те са нормални формули от първи ред. Например:

$$
\begin{aligned}
\langle flora, fruit(apple) \rangle \\
\langle fauna, eagle(x) \rightarrow bird(x) \rangle \\
\langle c, \alpha \wedge \beta \rightarrow \gamma \rangle
\end{aligned}
$$

Вторият вид формули са мета-формулите, които са необходими за взаимодействието между различните контексти. Те също могат да изразяват познания за света, но в тях по един или друг начин са намесени различни контексти. В основата на такива формули стои формулата $\langle c_1, ist(c_2, \alpha) \rangle$, която означава, че в контекста c_1 е вярно, че в контекста c_2 е изпълнено α. С нейна помощ ще можем в един контекст да записваме твърдения за друг. Ето няколко примера:

$$\langle animals, ist(birds, flies(eagle)) \rightarrow flies(eagle)\rangle$$
$$\langle pet, cat(tom) \wedge ist(cat, young(tom)) \rightarrow kitty(tom)\rangle$$
$$\langle speed, ist(car, fast(toyota)) \wedge ist(car, faster(porshe, toyota)) \rightarrow ist(car, fast(porshe))\rangle$$

Забележете, че $\langle c_1, ist(c_2, \alpha)\rangle$ не означава непременно, че α е изпълнено в c_2. Смисълът на тази формула трябва да се възприема като „контекстът c_1 *вярва*, че в контекста c_2 е вярно α„, т.е. това негово убеждение може да е, но може и да не е вярно. Един начин да се стигне до заблуждение е да се проведе верига от мета-разсъждения, т.е. на базата на свои познания за света и за познанията на c_2 c_1 да реши, че в c_2 е изпълнено α, без това да е вярно. Свойствата на ist-формулите ще разгледаме по-нататък.

За да постигнем добра изразителна сила, трябва да можем да изразяваме най-различни твърдения за съдържанието на различните контексти. Сама по себе си ist-формулата не е достатъчна - трябва да позволим композицията й с другите формули. Искаме изразите от вида $ist(c_1, \alpha) \vee \beta, \forall\, c\; ist(c, \alpha), \neg ist(c, \alpha), ist(c_1, ist(c_2, \alpha))$ също да са формули. Затова е подходящо да третираме синтактично контекстите като обекти от първи ред, подобно на индивидуалните константи и променливи. Така стигаме до следните индуктивни дефиниции:

Нека $\mathcal{A}$ е крайна азбука, несъдържаща символите $\langle, \rangle, (,)$ и запетая. Нека имаме едно множество от индекси I и $\mathcal{A}_i$ за $i \in I$ са крайни азбуки, несъдържащи символите $\langle, \rangle, (,), \neg, \vee, \wedge, \rightarrow, \forall, \exists, \bot, \top$ и запетая. Нека още

$\mathcal{VAR}_i$ са избрани множества от думи над $\mathcal{A}_i$, чиито елементи ще наричаме съждителни променливи и обикновено ще бележим с $p, q, r, p_1, p', \ldots$

$\mathcal{IND}_i$ са избрани множества от думи над $\mathcal{A}_i$, чиито елементи ще наричаме индвидуални променливи и обикновено ще бележим с $x, y, z, x_1, x', \ldots$

$\mathcal{CONST}_i$ са избрани множества от думи над $\mathcal{A}_i$, чиито елементи ще наричаме индвидуални константи.

$\mathcal{F}_i$ са избрани множества от думи над $\mathcal{A}_i$, чиито елементи ще наричаме функционални символи и обикновено ще бележим с $f, g, h, f_1, f', \ldots$

$\mathcal{PR}_i$ са избрани множества от думи над $\mathcal{A}_i$, чиито елементи ще наричаме предикатни символи и обикновено ще бележим с $P, Q, R, P_1, P', \ldots$

$\mathcal{CC}$ е множество от думи над $\mathcal{A}$, такова че $|\mathcal{CC}| = |I|$, чиито елементи ще наричаме контекстни константи и обикновено ще бележим с $c_1, c_2, c^{'}, ...$

$\mathcal{CV}$ е множество от думи над $\mathcal{A}$, чиито елементи ще наричаме контекстни променливи и обикновено ще бележим с $v_1, v_2, v^{'}, ...$

$\mathcal{CX} = \mathcal{CC} \cup \mathcal{CV}$.

Нека за всяко $i \in I$ множествата $\mathcal{VAR}_i$, $\mathcal{IND}_i$, $\mathcal{CONST}_i$, $\mathcal{F}_i$, $\mathcal{PR}_i$, $\mathcal{CC}$ и $\mathcal{CV}$ са най-много изброими и са чужди помежду си, като $\mathcal{VAR}_i$, $\mathcal{IND}_i$, $\mathcal{CONST}_i$, $\mathcal{F}_i$ и $\mathcal{PR}_i$ не съдържат думата *ist*.

Нека също $ar_i : \mathcal{F}_i \cup \mathcal{PR}_i \rightarrow \mathcal{N}$ са функции, които определят размерността (арността) на функционалните и предикатните символи, т.е.

• ако $f \in \mathcal{F}_i$, то $ar_i(f) = n$ означава, че f е n-местен функционален символ в контекста c_i

• ако $P \in \mathcal{PR}_i$, то $ar_i(P) = n$ означава, че P е n-местен предикатен символ в контекста c_i.

Забележете, че функциите ar_i не са част от синтаксиса на езика. Те ни помагат да следим размерността на предикатните и функционалните символи в различните контексти.

Дефиниция 2.1. Всеки език над азбуката $\mathcal{A} \cup \cup_{i \in I} \mathcal{A}_i$, в който са фиксирани горните множества и функции, ще наричаме *многоконтекстен език от първи ред.*

Дефиниция 2.2. Понятието *терм в един многоконтекстен език* $\mathcal{MCL}$ ще дефинираме със следната индуктивна дефиниция:

1. *атомарни терми* са $\langle e, x \rangle$ и $\langle e, a \rangle$, където $e \in \mathcal{CX}$, $x \in \mathcal{IND}_i$, а $a \in \mathcal{CONST}_i$.

2. Ако $f \in \mathcal{F}_i$, $ar_i(f) = n$ за някое $i \in I$ и $\langle e, t_1 \rangle$, ..., $\langle e, t_n \rangle$ са терми, то $\langle e, f(t_1, ..., t_n) \rangle$ е също терм, който ще наричаме *съставен.*

Дефиниция 2.3. *Правилно построена формула (ППФ) в* $\mathcal{MCL}$ дефинираме също индуктивно:

1. *атомарни формули* са

• $\langle e, \bot \rangle$, където $e \in \mathcal{CX}$

• $\langle e, \top \rangle$, където $e \in \mathcal{CX}$

• $\langle e, p \rangle$, където $e \in \mathcal{CX}$, а $p \in \mathcal{VAR}_i$ за някое $i \in I$

• $\langle e, P(t_1, ..., t_n) \rangle$, където $e \in \mathcal{CX}$, $P \in \mathcal{PR}_i$, $ar_i(P) = n$, за някое $i \in I$ и $\langle e, t_j \rangle, j = 1..n$ са терми

• $\langle e, ist(e_1, \alpha) \rangle$, където $e \in \mathcal{CX}$, $e_1 \in \mathcal{CX}$ и $\langle e_1, \alpha \rangle$ е правилно построена формула.

2. Ако $\langle e, \alpha \rangle$ и $\langle e, \beta \rangle$ са правилно построени формули, $e \in \mathcal{CX}$, $x \in \mathcal{IND}_i$ за някое $i \in I$ и $v \in \mathcal{CV}$, то $\langle e, \neg\alpha \rangle$, $\langle e, (\alpha \vee \beta) \rangle$, $\langle e, (\alpha \wedge \beta) \rangle$, $\langle e, (\alpha \rightarrow \beta) \rangle$, $\langle e, (\forall x)\alpha \rangle$, $\langle e, (\exists x)\alpha \rangle$, $\langle e, (\forall v)\alpha \rangle$, $\langle e, (\exists v)\alpha \rangle$, са също правилно построени формули. Тях ще наричаме *съставни.*

Дефиниция 2.4. *Многоконтекстна система MCS над езика* $\mathcal{MCL}$ ще наричаме наредената двойка $\langle \{c_i\}_{i \in I}, \Delta \rangle$, където c_i са примитивни контексти, а Δ е множество от правила за извод със смяна на контекста.

Дефиниция 2.5. *Примитивен контекст c_i на MCS* ще наричаме формално аксиоматичната система $\langle L_i, A_i, \Delta_i \rangle$, където езикът L_i е множеството от всички ППФ от вида $\langle c_i, \varphi \rangle$, $A_i \subseteq L_i$ е множество от формули, които ще наричаме *контекстни аксиоми на c_i*, а Δ_i е множество от функции от вида $L_i^* \rightarrow L_i$, които ще наричаме *правила за извод на контекста c_i*.

Дефиниция 2.6. *Контекстни аксиоми на многоконтекстната система MCS* ще наричаме множеството $Axioms = \cup_{i \in I} A_i$.

Дефиниция 2.7. Формулите от типа $\langle e, ist(c_i, \alpha) \rangle$ ще наричаме *ist-формули*, а всички формули, съдържащи в себе си поне една ist-формула, ще наричаме *мета-формули*.

От тези дефиниции става ясно, че всяка ППФ от първи ред, записана в контекстна нотация е ППФ в някоя многоконтекстна система. Другият вид ППФ в многоконтекстните системе са мета-формулите.

Контекстите се описват с помощта на контекстни аксиоми. Тези аксиоми всъщност представляват формули, описващи знанията, които се подразбират в даден контекст. За да осигурим максимална автономност на контекстите, а оттам и максимална гъвкавост, всеки контекст разполага със собствен език и собствени правила за извод. Тук са дефинирани само примитивните контексти; за съставните ще говорим в глава 5.

Дефиниция 2.8. *Правила за смяна на контекста* ще наричаме такива правила, чиито заключения са в друг контекст спрямо някоя от предпоставките.

Правилата за извод представляват правила, чрез които от няколко налични формули - предпоставки, можем да изведем нова формула - заключение и евентуално да елиминираме предположение (допускане), което сме направили по-рано, ако заключението не зависи от това допускане. Всеки контекст разполага със собствени правила, които се използват в рамките на контекста - при тях всички формули задължително са от този контекст. Отделно от тях многоконтекстната система предоставя набор от общи правила, с чиято помощ разсъжденията могат да преминават от един контекст в друг. Синтактично правилата ще записваме по следния начин:

$$\frac{\begin{array}{c}[assumption_1]\ \ldots\ [assumption_k]\\ premiss_1\ \ premiss_2\ \ \ldots\ premiss_n\end{array}}{conclusion}\, name$$

Тук $assumption_1 ... assumption_k$ са допусканията, които правилото елиминира. Те се ограждат с квадратни скоби и се разполагат над предпоставките $premiss_1 ... premiss_n$, които се намират над хоризонталната черта, а под нея се поставя заключението *conclusion*. Вдясно от хоризонталната черта се поставя името *name* на правилото.

Общият вид на едно просто правило за смяна на контекста е този:

$$\frac{\langle c_1, \varphi \rangle}{\langle c_2, \psi \rangle}$$

Тук заключението се извежда само от една предпоставка, т.е. извършва се преход от контекста c_1 в c_2. Възможно е използването и на по-общи правила за многоконтекстен извод, в които участват няколко предпоставки и дори допускания, които се елиминират. Най-общият вид на едно такова правило е

$$\frac{\begin{array}{c}[\langle c_{i_1}, \alpha_{i_1} \rangle] \; ... \; [\langle c_{i_k}, \alpha_{i_k} \rangle] \\ \langle c_1, \beta_1 \rangle \quad \langle c_2, \beta_2 \rangle \quad ... \langle c_n, \beta_n \rangle\end{array}}{\langle c, \gamma \rangle} name$$

В тази дипломна работа обаче ще използваме най-често правило от първия вид:

$$\frac{\langle c_i, \varphi \rangle}{\langle c_j, ist(c_i, \varphi) \rangle} SF$$

Името SF на това правило произлиза от *Switch Forward* - преход напред.

Дефиниция 2.9. Ако $\langle c_i, t \rangle$ е терм в MCS, ще казваме, че t е терм в контекста c_i. Аналогично ще казваме, че φ е формула в c_i, когато $\langle c_i, \varphi \rangle$ е формула в MCS.

По-нататък следват многоконтекстни аналози на стандартни дефиниции в логиката от първи ред:

Дефиниция 2.10. Нека $v \in \mathcal{CV}$и $x \in \mathcal{IND}_i$, за някое $i \in I$. Ще казваме, че x / v *се среща* в терма $\langle c_i, t \rangle$ / формулата $\langle c_i, \varphi \rangle$, ако x / v е един от символите на $\langle c_i, t \rangle$ / $\langle c_i, \varphi \rangle$.

Дефиниция 2.11. Казваме, че дадено срещане на $v \in \mathcal{CV}$/ $x \in \mathcal{IND}_i$ за някое $i \in I$ във формулата $\langle e, \alpha \rangle$ е *свързано*, ако α има вида $\alpha = ...(Qx)\beta...$, където $\langle e, \beta \rangle$ е формула, Q е един от квантификаторите $\forall$ и $\exists$ и срещането на v / x е непосредствено след Q или е в β.

Дефиниция 2.12. Казваме, че дадено срещане на v / x във формулата φ е *свободно*, ако то не е свързано.

Пример.

1. В $\langle e, (\forall x)P(x)\rangle$ има две срещания на x като и двете са свързани.

2. В $\langle e, (\forall x)P(y)\rangle, x \neq y$ има 1 свързано срешане на x и едно свободно срещане на y.

3. В $\langle e, (\forall x)P(x) \vee P(x)\rangle$ има 3 срещания на x, първите две от които са свързани, а последното е свободно.

4. В $\langle e, (\exists v)ist(v, ist(w,p)\rangle, v \neq w$ има две свързани срещания на v и едно свободно на w.

Дефиниция 2.13. Нека $\langle e, t\rangle$ и $\langle e, h\rangle$ са терми, а $x \in \mathcal{IND}_i$ за някое $i \in I$. Със записа $\langle e, t(x)\rangle$ ще означаваме терма $\langle e, t\rangle$ и факта, че обръщаме внимание на всички срещания на x в него. Ако на мястото на всяко срещане на x поставим h, резултатът ще означаваме с $\langle e, t(h)\rangle$.

Дефиниция 2.14. Нека $\langle e, \alpha\rangle$ е формула, $\langle e, t\rangle$ е терм, а $x \in \mathcal{IND}_i$ за някое $i \in I$. Със записа $\langle e, \alpha(x)\rangle$ ще означаваме формулата $\langle e, \alpha\rangle$ и факта, че обръщаме внимание на всички срещания на x в нея. Ако на мястото на всяко срещане на x поставим t, резултатът ще означаваме с $\langle e, \alpha(t)\rangle$.

Лема 2.1. Ако $\langle e, t\rangle$ и $\langle e, h\rangle$ са терми, то $\langle e, t(h)\rangle$ е също терм.

Лема 2.2. Ако $\langle e, \alpha\rangle$ е формула, а $\langle e, t\rangle$ е терм, то $\langle e, \alpha(t)\rangle$ е също формула.

Няма да се спираме на доказателството на тези леми, защото то е тривиално копие на доказателството им в класическата логика.

Дефиниция 2.15. Казваме, че термът $\langle e, t\rangle$ е *допустим за заместване на $x \in \mathcal{IND}_i$ във формулата* $\langle e, \alpha(x)\rangle$, ако x не се среща в областта на действие на квантификатор (Qy), където y е променлива от t.

Дефиниция 2.16. Една формула ще наричаме *затворена*, ако няма свободни променливи. В противен случай ще я наричаме *отворена*.

Дефиниция 2.17. Нека имаме една непразна, крайна редица от формули от една многоконтекстна система MCS, всеки член на която е или контекстна аксиома, която ще наричаме *предпоставка*, или произволна формула, която ще наричаме *допускане*, или се получава от няколко предходни (може и 0) по някое правило за извод или правило за смяна на контекста. Всяка такава редица, в която няма неелиминирани допускания, ще наричаме *доказателство* в MCS.

Тази дефиниция не е пълен аналог на традиционната. Разликите са две: наличието на допускания в доказателството и липсата на логическите аксиоми. Логическите аксиоми са характерни за класическото предикатно смятане. Има логики, които не ги използват - една такава ще разгледаме в глава 3. Това че не присъстват явно в тази дефиниция, е направено с цел тя да бъде максимално обща, като същевременно запази автономността на контекстите. Ако в един контекст се използва апарата на класическата логика от първи ред, тогава тези аксиоми можем да ги формулираме като негови правила (както сме направили това в раздел 4.1); в противен случай - аксиомите са излишни.

Наличието на допускания е обосновано в раздел 4.1. Тяхното елиминиране става с помощта на някои от правилата. *Правило, елиминиращо допускане* ще наричаме правило, което може да има предпоставка, зависеща от дадено допускане, но заключението му не зависи от това допускане. Съответното допускане е в сила (неелиминирано) от мястото на появяването си в редицата на доказателството до мястото на срещане на правило, което го елиминира. Ако в една редица има допускане, което остава в сила до края на редицата, ще казваме че това правило е *неелиминирано* и съответната редица няма да я считаме за доказателство.

Пример. Едно класическо правило, елиминиращо допускане е правилото *Implication Introduction* - въвеждане на импликацията:

$$\frac{\begin{array}{c}[\langle c, \alpha\rangle] \\ \langle c, \beta\rangle\end{array}}{\langle c, \alpha \to \beta\rangle} II$$

Дефиниция 2.18. Една формула ще наричаме *доказуема в* MCS, ако е пореден член на някое доказателство.

Дефиниция 2.19. Нека MCS е една многоконтекстна система и φ е правилно построена формула. Ще казваме, че *от* MCS *следва синтактично* φ, записваме $MCS \vdash \varphi$, когато съществува крайна редица от формули $\varphi_1, ..., \varphi_n$, която е доказателство в MCS и чийто последен член е $\varphi_n = \varphi$. Редицата $\varphi_1, ..., \varphi_n$ се нарича *доказателство на* φ *от* MCS.

Забележка. Със записа $A \vdash \varphi$ означаваме, че формулата φ е изводима от A според синтаксиса на класическата логика от първи ред, ако A е множество. Ако A е многоконтекстна система, то $A \vdash \varphi$ означава, че φ е изводима от A според синтаксиса на многоконтекстните системи.

Дефиниция 2.20. С $Th(X)$, както в класическата логика от първи ред, ще бележим всички формули, изводими от множеството X, т.е. $Th(X) = \{\varphi | X \vdash \varphi\}$. Аналогично, множеството формули, изводими от една многоконтекстна система X, ще наричаме *многоконтекстна теория* и ще бележим $Mcth(X) = \{\varphi | X \vdash \varphi\}$.

Дефиниция 2.21. *Литерал (прост факт)* ще наричаме атомарна формула или нейно отрицание. С $Lit(X)$ ще означаваме всички литерали изводими от множеството X или от многоконтекстната система X.

Дефиниция 2.22. Нека $MCS = \langle \{c_i\}_{i \in I}, \Delta\rangle$ е една многоконтекстна система, в която $c_i = \langle L_i, A_i, \Delta_i\rangle, i \in I$ и нека $\langle c_k, \varphi\rangle, k \in I$ е правилно построена формула в MCS. Със записа $MCS \cup \langle c_k, \varphi\rangle$ ще означаваме $\langle \{c_i\}_{i \in I, i \neq k} \cup \{c_k = \langle L_k, A_k \cup \{\langle c_k, \varphi\rangle\}, \Delta_k\rangle\}, \Delta\rangle$.

Дефиниция 2.23. Нека MCS е една многоконтекстна система. Със записа $MCS(i|c)$ ще означаваме многоконтекстната система, която се получава от MCS като на мястото на контекста c_i се постави примитивния контекст $c_0 = \langle L_0, A_0, \Delta_0 \rangle$, ако $i \in I$, или в противен случай контекстът c се добави с нов индекс.

Дефиниция 2.24.

1. Два контекста $c_i = \langle L_i, A_i, \Delta_i \rangle$ и $c_j = \langle L_j, A_j, \Delta_j \rangle$ ще наричаме *равни*, бележим $c_i = c_j$, ако $L_i \equiv L_j$, $A_i \equiv A_j$ и $\Delta_i \equiv \Delta_j$.

2. Два контекста $c_i = \langle L_i, A_i, \Delta_i \rangle$ и $c_j = \langle L_j, A_j, \Delta_j \rangle$ ще наричаме *синтактично еквивалентни*, ако $Mcth(MCS(i|c_j)) \equiv Mcth(MCS)$. Този факт ще бележим с $c_1 \simeq c_2$

Дефиниция 2.25. Нека $c_i = \langle L_i, A_i, \Delta_i \rangle$ и $c_j = \langle L_j, A_j, \Delta_j \rangle$ са два контекста в многоконтекстната система MCS. Нека $B_i = \{\alpha | \langle c_i, \alpha \rangle \in Mcth(MCS)\}$ и $B_j = \{\beta | \langle c_j, \beta \rangle \in Mcth(MCS)\}$. Със записа $c_i \subseteq c_j$ ще означаваме факта $B_i \subseteq B_j$.

Дефиниция 2.26. Нека имаме една многоконтекстна система MCS и $i \in I$. *Импортна част на контекста* c_i ще наричаме множеството формули, изводими в c_i без помощта на неговите контекстни аксиоми, т.е. $Imp(c_i) = Mcth(MCS(i|\langle L_i, \phi, \Delta_i \rangle))$.

Дефиниция 2.27. *Собствена част на контекста* c_i ще наричаме множеството формули, изводими в c_i без помощта на правила за смяна на контекста, т.е. $Prop(c_i) = Mcth(\langle \{c_i\}, \phi \rangle)$.

2.2 Предположения

Дадените по-горе дефиниции, свързани със синтаксиса на многоконтекстните системи, са възможно най-общи. На практика при използването им в този най-общ вид възникват редица затруднения, свързани с нотацията и семантиката на мета-формулите.

Трудностите с нотацията се дължат на факта, че различните контексти могат да имат различни азбуки. Например, за да има семантичен смисъл формулата $\langle c_k, ist(c_l, \alpha) \rangle$ (за семантиката ще говорим след малко), α трябва да е правилно построена формула в контекста c_l. Но контекстите c_k и c_l могат да имат съвсем различни езици. В такъв случай възниква въпрос: Как да направим така, че α да е правилно построена формула едновременно в c_k и в c_l?

Едно възможно решение на този проблем е на всеки контекст да се осигури „превод„ на формулите от езиците на другите контексти на неговия. Това е възможно да се постигне с една превеждаща функция $map(c_i, \alpha)$, която по зададено име на контекст c_i и формула α в този контекст да ни връща тази формула, преформулирана от езика на c_i на езика на контекста, в който се прави текущото разсъждение. Това решение обаче на практика е използваемо, само ако контекстите имат различни азбуки, но не и различни правила за извод.

По-сериозните трудности идват от начините на използване на формулите в различните контексти т.е. правилата за извод. Ако искаме пълна автономия на контекстите, това означава, че всеки контекст може да има различни правила за извод. В такъв случай изразната сила на различните контексти ще бъде различна. Можем, например, да съставим контекст, в който да няма логическа функция „или„. Тогава преводът на формули, съдържащи тази логическа функция, на езика на този контекст се превръща в изключително трудна задача.

На практика, преводът на знания между различни формализми е много актуална тема, по която в момента се работи усилено на няколко места по света. Само за информация - един от екипите, занимаващи се с този проблем е групата по логика в Станфордския университет, която се състои от такива световно известни учени като Майкъл Генесерет, Ричард Файкс и др. Решаването на този проблем обаче е далеч от целите на тази дипломна работа и затова в нея ще прибегнем до следните предположения:

Оттук нататък ще разглеждаме само многоконтекстни системи, чиито контексти са изградени над един и същ език $L = L_i$, за $i \in I$, и правила за извод в тези контексти, които са равномощни на правилата на класическата логика от първи ред. Ще се спрем само многоконтекстни системи с единствено правило за смяна на примитивен контекст

$$\frac{\langle c_i, \varphi \rangle}{\langle c_j, ist(c_i, \varphi) \rangle} SF.$$

2.3 Семантика

Нека имаме едно множество $D : |D| > 1$, което ще наричаме *проблемна област* или *област на компетентност*, и една функция $\rho : \mathcal{MCL} \to D$, която ще наричаме *смислова функция*. Нека ρ е такава, че ако $c_k \in \mathcal{CC}$, то

- ако $a \in \mathcal{CONST}$, то $\rho(\langle c_k, a \rangle) \in D$
- ако $f \in \mathcal{F}$, $ar(f) = n$, то $\rho(\langle c_k, f \rangle) : D^n \to D$
- ако $P \in \mathcal{PR}$, $ar(P) = n$, то $\rho(\langle c_k, P \rangle) \subseteq D^n$
- ако $p \in \mathcal{VAR}$, то $\rho(\langle c_k, p \rangle) \in \{true, false\}$
- $\rho(\langle c_k, \bot \rangle) = false, \rho(\langle c_k, \top \rangle) = true$

Дефиниция 2.28. *Интерпретация на една многоконтекстна система* MCS ще наричаме наредената двойка $J = \langle D, \rho \rangle$.

Дефиниция 2.29. *Оценка* V в J ще наричаме всяка функция $V : \mathcal{CC} \times \mathcal{IND} \to D$ и $V : \mathcal{CV} \to \mathcal{CC}$.

Дефиниция 2.30. *Стойност на терм t при оценка V* дефинираме по следния начин:

1. ако $t = \langle c_k, a\rangle, a \in \mathcal{CONST}$, то $V(t) = \rho(\langle c_k, a\rangle)$
2. ако $t = \langle c_k, x\rangle, x \in \mathcal{IND}$, то $V(t) = V(\langle c_k, x\rangle)$
3. ако $t = \langle c_k, f(t_1, ..., t_n)\rangle, f \in \mathcal{F}$, $ar(f) = n$,
то $V(t) = \rho(\langle c_k, f\rangle)(V(\langle c_k, t_1\rangle), ..., V(\langle c_k, t_n\rangle))$
4. ако $t = \langle v, h\rangle, v \in \mathcal{CV}$, то $V(t) = V(\langle V(v), h\rangle)$

Дефиниция 2.31. *Стойност на формула φ при оценка V* също дефинираме индуктивно:

1. $V(\langle c_k, \bot\rangle) = false$, $V(\langle c_k, \top\rangle) = true$
2. ако $\varphi = \langle c_k, p\rangle, p \in \mathcal{VAR}$, то $V(\varphi) = \rho(\langle c_k, p\rangle)$
3. ако $\varphi = \langle c_k, P(t_1, ..., t_n)\rangle, P \in \mathcal{PR}$, $ar(P) = n$,
то $V(\varphi) = true \Leftrightarrow (V(\langle c_k, t_1\rangle), ..., V(\langle c_k, t_n\rangle)) \in \rho(\langle c_k, P\rangle)$
4. ако $\varphi = \langle c_k, \neg\alpha\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle c_k, \alpha\rangle) = false$.
5. ако $\varphi = \langle c_k, \alpha \vee \beta\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle c_k, \alpha\rangle) = true$ или $V(\langle c_k, \beta\rangle) = true$
6. ако $\varphi = \langle c_k, \alpha \wedge \beta\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle c_k, \alpha\rangle) = true$ и $V(\langle c_k, \beta\rangle) = true$
7. ако $\varphi = \langle c_k, \alpha \rightarrow \beta\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle c_k, \alpha\rangle) = false$ или $V(\langle c_k, \beta\rangle) = true$
8. ако $\varphi = \langle c_k, (\forall x)\alpha\rangle, x \in \mathcal{IND}$, образуваме нова оценка $V_a^x(y) = a$, ако $y = x$ и $V_a^x(y) = V(y)$, ако $y \neq x$. Тогава $V(\varphi) = true \Leftrightarrow$ за всяко $a \in D$ $V_a^x(\langle c_k, \alpha\rangle) = true$
9. ако $\varphi = \langle c_k, (\exists x)\alpha\rangle, x \in \mathcal{IND}$, образуваме нова оценка $V_a^x(y) = a$, ако $y = x$ и $V_a^x(y) = V(y)$, ако $y \neq x$. Тогава $V(\varphi) = true \Leftrightarrow$ съществува $a \in D$ $V_a^x(\langle c_k, \alpha\rangle) = true$
10. ако $\varphi = \langle c_k, (\forall v)\alpha\rangle, v \in \mathcal{CV}$, образуваме нова оценка $V_c^v(y) = c$, ако $y = v$ и $V_c^v(y) = V(y)$, ако $y \neq v$. Тогава $V(\varphi) = true \Leftrightarrow$ за всяко $c \in \mathcal{CC}$ $V_c^v(\langle c, \alpha\rangle) = true$
11. ако $\varphi = \langle c_k, (\exists v)\alpha\rangle, v \in \mathcal{CV}$, образуваме нова оценка $V_c^v(y) = c$, ако $y = v$ и $V_c^v(y) = V(y)$, ако $y \neq v$. Тогава $V(\varphi) = true \Leftrightarrow$ съществува $c \in \mathcal{CC}$ $V_c^v(\langle c, \alpha\rangle) = true$
12. ако $\varphi = \langle c_k, ist(c_m, \alpha)\rangle$, то $V(\varphi) = V(\langle c_m, \alpha\rangle)$
13. ако $\varphi = \langle v, \alpha\rangle, v \in \mathcal{CV}$, то $V(\varphi) = V(\langle V(v), \alpha\rangle)$

Дефиниция 2.32. Нека $J = \langle D, \rho\rangle$ е една интерпретация на многоконтекстната система MCS. J се нарича *модел* на една формула φ (φ е вярна в J), ако за всяка оценка V в J $V(\varphi) = true$.

Дефиниция 2.33. Една формула се нарича *предикатна тавтология (закон)* на многоконтекстната система $MCS = \langle\{c_i\}_{i\in I}, \Delta\rangle$, $c_i = \langle L_i, A_i, \Delta_i\rangle$, ако всяка интерпретация на $MCS = \langle\{\langle L_i, \phi, \Delta_i\rangle\}_{i\in I}, \Delta\rangle$ е неин модел.

Дефиниция 2.34. Нека MCS е една многоконтекстна система над $\mathcal{MCL}$ и φ е една правилно построена формула над $\mathcal{MCL}$. Казваме, че *от MCS следва семантично φ*, бележим $MCS \models \varphi$, ако всеки модел на MCS е модел и на φ.

Дефиниция 2.35. Една интерпретация J е *модел на контекста c_k на многоконтекстната система MCS*, ако е модел на неговите контекстни аксиоми. J е *модел на многоконтекстната система MCS*, ако е модел на всички нейни контексти.

Както се вижда, тези дефиниции са преки обобщения на дефинициите в класическата логика. Единственият предикат, който има по-специален смисъл е предикатът ist, чието значение вече изяснихме.

Глава 3

Безконтекстни системи

3.1 Класическа логика от първи ред

Определено, най-разпространеният логически апарат от първи ред е предикатното смятане от първи ред, което тук ще наричаме логика FC (FC - Functional Calculus). Изводът при логиката FC се основава на дванадесет логически аксиоми $Ax_1 ... Ax_{12}$ и две правила - MP и Gen:

$$
\begin{array}{rl}
Ax_1. & \alpha \to (\beta \to \alpha) \\
Ax_2. & (\alpha \to \beta) \to ((\alpha \to (\beta \to \gamma)) \to (\alpha \to \gamma)) \\
Ax_3. & \alpha \to (\beta \to \alpha \wedge \beta) \\
Ax_4. & \alpha \wedge \beta \to \alpha \\
Ax_5. & \alpha \wedge \beta \to \beta \\
Ax_6. & \alpha \to \alpha \vee \beta \\
Ax_7. & \beta \to \alpha \vee \beta \\
Ax_8. & (\alpha \to \gamma) \to ((\beta \to \gamma) \to (\alpha \vee \beta \to \gamma)) \\
Ax_9. & (\alpha \to \beta) \to ((\alpha \to \neg\beta) \to \neg\alpha) \\
Ax_{10}. & \neg\neg\alpha \to \alpha \\
Ax_{11}. & (\forall x)\alpha(x) \to \alpha(t) \\
Ax_{12}. & (\forall x)(\alpha \to \beta) \to (\alpha \to (\forall x)\beta)
\end{array}
$$

$$
MP : \frac{\alpha \quad \alpha \to \beta}{\beta} \qquad\qquad Gen : \frac{\alpha}{(\forall x)\alpha}
$$

В тези аксиоми за улеснение не са включени логическите константи $\leftrightarrow$, $\exists$, $\top$ и $\bot$, но те могат да се изразят чрез останалите:

$$\begin{aligned}
\alpha \leftrightarrow \beta \quad & \Leftrightarrow (\alpha \rightarrow \beta) \wedge (\alpha \leftarrow \beta) \\
(\exists x)\alpha \quad & \Leftrightarrow \neg(\forall x)\neg\alpha \\
\top \quad & \Leftrightarrow \alpha \vee \neg\alpha \\
\bot \quad & \Leftrightarrow \alpha \wedge \neg\alpha
\end{aligned}$$

По дефиниция доказателство в логиката FC наричаме непразна редица от формули, всеки член на която се получава от един предходен по правилото Gen, от два предходни - по правилото MP, или е логическа аксиома. Както вече споменахме, в многоконтекстния случай тази дефиниция сме я изменили и обобщили с цел обхващане и на други логики.

От теорията на предикатното смятане от първи ред ще използваме наготово следните теореми:

Теорема 3.1. **(Теорема за дедукцията)** Нека X е едно множество от формули и α и β са формули. Тогава извеждането на β от $X \cup \{\alpha\}$ е равносилно на извеждането на $\alpha \rightarrow \beta$ от X, т.е.

$$X \cup \{\alpha\} \vdash \beta \Leftrightarrow X \vdash \alpha \rightarrow \beta$$

Теорема 3.2. **(Теорема за пълнота)** Ако една формула α е синтактично изводима от едно множество X, то тя е семантично изводима от това множество и обратно, ако α е семантично изводима от X, то съществува крайна редица от формули, която е доказателство на α от X. Записано накратко:

$$X \vdash \alpha \Leftrightarrow X \models \alpha$$

Няма да се спираме по-подробно на логиката FC, защото тя е широко разпространена.

3.2 Логика NK. Правила за естествена дедукция

Един основен въпрос в теорията на общите доказателства, който има и фундаментално значение за теорията на редукцията, е как да се анализира нотацията на логическите доказателства в езиците от първи ред. Работата на Генцен [3] може да се разглежда като отговор на този въпрос. Отговорът е даден на две стъпки: В първия си анализ Генцен показва как нотацията може да се дефинира в термините на определени формални системи, конструирани от него. След това, с по-задълбочен анализ на структурата на тези доказателства, той показва че те могат да се запишат в много специална форма, която като цяло дава добри резултати при разбирането на тези доказателства.

3.2.1 Основна идея

Системата за естествена дедукция на Генцен възниква от един анализ на дедуктивните изводи, при който основна роля играят различните логически константи. Изводите са разделени на атомарни стъпки по такъв начин, че на всяка стъпка се разглежда точно една логическа константа. Стъпките биват два вида и за всяка логическа константа има по една стъпка и от двата вида - това са *въвеждане* на логическа константа и *елиминиране* на логическа константа.

Доказателствата започват с *допускания*, които на определени стъпки се елиминират. Това става най-често с правилото II - допускането α се елиминира с въвеждането на импликацията $\alpha \to \beta$.

3.2.2 Правила за извод

Удобно е доказателствата да се представят под формата на дърво. В такъв случай най-горните формули (формулите по листата на дървото) са допусканията, а останалите следват от формулите, които са непосредствено над тях, по някое от правилата за извод, които формализират атомарните стъпки. Една формула α в дървото зависи от тези допускания над нея, които не са елиминирани от правило, което да я предхожда. Отворени допускания в дървото са тези, от които зависи последната формула.

Една формула, заградена в квадратни скоби и поставена над някоя предпоставка, означава, че допусканията от този вид, които се срещат над предпоставката, се елиминират от това правило. Повечето правила се именуват с инициала на логическата константа, за която се отнасят, следван от I, когато тази константа се въвежда, и от E, когато тя се елиминира.

$$AI: \quad \frac{\alpha \quad \beta}{\alpha \wedge \beta} \qquad\qquad AE: \frac{\alpha \wedge \beta}{\alpha} \quad \frac{\alpha \wedge \beta}{\beta}$$

$$OI: \quad \frac{\alpha}{\alpha \vee \beta} \quad \frac{\beta}{\alpha \vee \beta} \qquad\qquad OE: \frac{\alpha \vee \beta \quad \begin{matrix}[\alpha] \\ \gamma\end{matrix} \quad \begin{matrix}[\beta] \\ \gamma\end{matrix}}{\gamma}$$

$$II: \quad \frac{\begin{matrix}[\alpha] \\ \beta\end{matrix}}{\alpha \to \beta} \qquad\qquad IE: \frac{\alpha \quad \alpha \to \beta}{\beta}$$

$$NI: \quad \frac{\begin{matrix}[\alpha] \\ \bot\end{matrix}}{\neg\alpha} \qquad\qquad NE: \frac{\alpha \quad \neg\alpha}{\bot}$$

$$FI: \quad \frac{\alpha(a)}{(\forall x)\alpha(x)} \qquad\qquad FE: \frac{(\forall x)\alpha(x)}{\alpha(t)}$$

$$EI: \quad \frac{\alpha(t)}{(\exists x)\alpha(x)} \qquad\qquad EE: \frac{(\exists x)\alpha(x) \quad \begin{matrix}[\alpha(a)] \\ \beta\end{matrix}}{\beta}$$

$$\frac{\bot}{\alpha} \qquad\qquad \frac{\top}{\alpha \vee \neg\alpha}$$

Собствена променлива на някое правило FI (съответно EE) се нарича свободната предметна променлива, обозначена в съответното правило с a. Предполага се, че тя съществува, т.е. че свързаната променлива, означена с x се среща във формулата $\alpha(x)$.

Наложени са следните ограничения на променливите: Собствената променлива на FI не трябва да се среща във формулата $(\forall x)\alpha(x)$, нито в някое допускане, от което тази формула зависи. Собствената променлива на EE не трябва да се среща във формулите $(\exists x)\alpha(x)$ и β, нито в някое от допусканията, от което зависи последната, с изключение на това, което е означено с $\alpha(a)$ и е съпоставено на това правило.

Смисълът на повечето от правилата е ясен, затова ще опишем само някои от тях, като същевременно ще покажем, че описаното изчисление наистина възпроизвежда действително разсъждение.

II. Това правило може да се изрази с думи така: „Ако β е доказано с помощта на допускането α, то (вече без това допускане) от α следва β.

OE. Ако сме доказали $\alpha \vee \beta$, то можем да направим разбор на случаите. Да допуснем най-напред, че е вярно α и от него изведем γ. Ако след това допуснем, че е вярно β и от него също изведем γ, тогава γ е вярно независимо от двете допускания.

FI. Ако $\alpha(a)$ е доказано за произволно a, тогава за всяко x е вярно $\alpha(x)$.

EE. Имаме $(\exists x)\alpha(x)$. По-нататък разсъждаваме така: нека a е такъв обект, за който е в сила $\alpha(x)$, т.е. да допуснем, че е вярно $\alpha(a)$. Ако в резултат на това допускане докажем някакво твърдение β, което не съдържа a и не зависи от никакви други допускания, съдържащи a, тогава β е доказано, независимо от допускането $\alpha(a)$.

NE. α и $\neg\alpha$ означава противоречие, а това не може да съответства на действителността (закон за противоречието).

NI. Ако от допускането α следва някакво противоречие ($\bot$), тогава α не е вярно. Следователно е вярно $\neg\alpha$.

3.2.3 Системите NJ и NK

В своя труд [3] Генцен разглежда най-различни логики, част от които са NJ и NK. Логиката NJ той нарича интуиционистка и включва горните правила, без последното вдясно - закона за изключеното трето. Ако към логиката NJ прибавим това правило, се получава класическата логика NK, за която Генцен доказва следната теорема:

Теорема 3.3. Ако една формула α е изводима от множеството X по правилата на класическата логика от първи ред, то съществува дърво на извод на α по правилата на логиката NK, чиито неелиминирани предпоставки са от X. Обратно, ако съществува извод на α по правилата за естествена дедукция, чиито неелиминирани предпоставки са от множеството X, то α е изводима от X по правилата на класическата логика от първи ред. Записано накратко:

$$X \vdash_{FC} \alpha \Leftrightarrow X \vdash_{NK} \alpha$$

3.2.4 Пример

Няма да се спираме на предимствата и недостатъците на логиката NK. Ще илюстрираме само нейната приложимост с един пример.

Пример. Да се докаже верността на формулата

$$(x \lor (y \land z)) \rightarrow ((x \lor y) \land (x \lor z))$$

Формално това изглежда така:

$$\dfrac{\dfrac{\overset{2}{x \lor (y \land z)} \qquad \dfrac{\dfrac{\overset{1}{x}}{x \lor y}\,OI \qquad \dfrac{\overset{1}{x}}{x \lor z}\,OI}{(x \lor y) \land (x \lor z)}\,AI \qquad \dfrac{\dfrac{\dfrac{\overset{1}{y \land z}}{y}\,AE}{x \lor y}\,OI \qquad \dfrac{\dfrac{\overset{1}{y \land z}}{z}\,AE}{x \lor z}\,OI}{(x \lor y) \land (x \lor z)}\,AI}{(x \lor y) \land (x \lor z)}\,OE1}{(x \lor (y \land z)) \rightarrow ((x \lor y) \land (x \lor z))}\,II2$$

Преведено на естествен език това доказателство изглежда така: Нека е вярно x или $y \land z$. Да разгледаме следните два случая: 1. Вярно е x. и 2. Вярно е $y \land z$. В случай 1 от допускането следва както $x \lor y$, така и $x \lor z$, а следователно и $((x \lor y) \land (x \lor z))$. В случай 2 е вярно $y \land z$ т.е. верно е както y, така и z. От y следва $x \lor y$, а от z следва $x \lor z$. Затова и в този случай е вярно $((x \vee y) \wedge (x \vee z))$. Така последната формула е изведена от $(x \vee (y \wedge z))$ и в общия случай, т.е.

$$(x \vee (y \wedge z)) \to ((x \vee y) \wedge (x \vee z)).$$

Глава 4

Многоконтекстни системи с примитивни контексти

4.1 Извод в многоконтекстни системи

В настоящата работа ще разглеждаме два набора контекстни правила за извод. Единият ще бележим с Δ_{FC}, защото той е разширение на правилата за извод в класическата логика от първи ред (Functional Calculus). Другият набор ще наречем Δ_{NK}, защото съдържа правилата за естествена дедукция на логиката NK.

4.1.1 FC-извод

За дефиниране на Δ_{FC} ще използваме аналог на аксиомите на класическата логика. Аксиомите по-долу, които ще наричаме *логически аксиоми* за разлика от контекстните аксиоми, са преформулировка на класическите аксиоми според синтаксиса на многоконтекстните системи:

$$
\begin{aligned}
Ax_1.&\ \langle c_i, \alpha \to (\beta \to \alpha)\rangle\\
Ax_2.&\ \langle c_i, (\alpha \to \beta) \to ((\alpha \to (\beta \to \gamma)) \to (\alpha \to \gamma))\rangle\\
Ax_3.&\ \langle c_i, \alpha \to (\beta \to \alpha \wedge \beta)\rangle\\
Ax_4.&\ \langle c_i, \alpha \wedge \beta \to \alpha\rangle\\
Ax_5.&\ \langle c_i, \alpha \wedge \beta \to \beta\rangle\\
Ax_6.&\ \langle c_i, \alpha \to \alpha \vee \beta\rangle\\
Ax_7.&\ \langle c_i, \beta \to \alpha \vee \beta\rangle\\
Ax_8.&\ \langle c_i, (\alpha \to \gamma) \to ((\beta \to \gamma) \to (\alpha \vee \beta \to \gamma))\rangle\\
Ax_9.&\ \langle c_i, (\alpha \to \beta) \to ((\alpha \to \neg\beta) \to \neg\alpha)\rangle\\
Ax_{10}.&\ \langle c_i, \neg\neg\alpha \to \alpha\rangle\\
Ax_{11}.&\ \langle c_i, (\forall x)\alpha(x) \to \alpha(t)\rangle\\
Ax_{12}.&\ \langle c_i, (\forall x)(\alpha \to \beta) \to (\alpha \to (\forall x)\beta)\rangle
\end{aligned}
$$

Виждаме, че на практика това са аксиомите на класическата логика, преформулирани в контекстна нотация, т.е. на всяка логическите аксиоми на един контекст съответства класическа аксиома и обратно. Този факт ще го използваме по-нататък в доказателствата.

В аксиомите за улеснение не са включени логическите формули от типа $\langle c_i, a \leftrightarrow \beta \rangle$, $\langle c_i, (\exists x)\alpha(x) \rangle$, $\langle c_i, \bot \rangle$ и $\langle c_i, \top \rangle$, но както знаем те се изразяват чрез останалите:

$$
\begin{aligned}
&\langle c_i, a \leftrightarrow \beta \rangle = \langle c_i, (a \rightarrow \beta) \wedge (a \leftarrow \beta) \rangle \\
&\langle c_i, (\exists x)\alpha(x) \rangle = \langle c_i, \neg(\forall x)\neg\alpha \rangle \\
&\langle c_i, \bot \rangle = \langle c_i, \alpha \wedge \neg\alpha \rangle \\
&\langle c_i, \top \rangle = \langle c_i, \alpha \vee \neg\alpha \rangle
\end{aligned}
$$

Класическите правила за извод са две: Modus ponens (MP) и Generalization (Gen):

$$
MP : \frac{\langle c_i, \alpha \rangle \quad \langle c_i, \alpha \rightarrow \beta \rangle}{\langle c_i, \beta \rangle} \qquad\qquad Gen : \frac{\langle c_i, \alpha \rangle}{\langle c_i, (\forall x)\alpha \rangle}
$$

Към тях ще добавим още едно, Premiss elimination (PE), чийто смисъл ще изясним след малко:

$$
PE : \qquad \frac{\begin{array}{c}[\langle c_i, \alpha \rangle] \\ \langle c_i, \beta \rangle\end{array}}{\langle c_i, \alpha \rightarrow \beta \rangle}
$$

Според дефиниция 2.19, в редицата, представляваща доказателство на една формула в една многоконтекстна система, не може да се срещат свободно логически аксиоми. Това не е недостатък на дефиницията - както вече споменахме, тя е формулирана по възможно най-общия начин с цел да осигури максимална независимост на контекстите. Според нея изводимостта във всеки контекст се определя (като изключим правилата за смяна на контекста, които са валидни за всички контексти) от неговите правила за извод.

Това, което всъщност дефиниция 2.19 ни забранява, е не да използваме логическите аксиоми в доказателството, а да ги използваме с аргумента, че те са логически аксиоми и следователно са верни при всяка интерпретация. В такъв случай е необходимо просто да сменим мотивировката за тяхното използване. Това може да стане като ги добавим под формата на правила за извод XI_i (Axiom introduction) към горните три:

$$XI_i : \frac{\langle c_i, \top \rangle}{Ax_i}$$

Така окончателно дефинираме

$$\Delta_{FC} = \{MP,\ Gen,\ PE, XI_i, i = 1..12\}$$

Сега да видим защо ни е необходимо правилото PE. Първо ще отбележим, че в безконтекстния случай с него не може да се изведе нищо ново. Наистина, нека имаме една редица $\varphi_1, ..., \varphi_n$, която е доказателство на една формула от множеството формули X с използването на това правило. За конкретност да предположим, че последната формула от редицата е $\varphi_n = \alpha \to \beta$ и е получена по правилото PE. Това означава, че редицата $\varphi_1, ..., \varphi_{n-1}$ е доказателство на формулата β от множеството формули $X \cup \{\alpha\}$, т.е. $X \cup \{\alpha\} \vdash \beta$. Тогава от теоремата за дедукцията имаме, че $X \vdash \alpha \to \beta$. Следователно φ_n може да се изведе от X и без помощта на правилото PE, т.е. с това правило в безконтекстния случай не може да се изведе нищо ново.

Тогава възниква въпросът: Защо сме го въвели? Да разгледаме следния пример: Нека имаме множеството $X = \{\varphi \to \neg\psi, \psi \to \varphi\}$. От него можем да изведем $\neg\psi$ по следния начин:

1. $\varphi \to \neg\psi$
2. $\psi \to \varphi$
3. $(\varphi \to \neg\psi) \to (\psi \to (\varphi \to \neg\psi))\ Ax_1$
4. $\psi \to (\varphi \to \neg\psi)\ MP\ 1,3$
5. $(\psi \to \varphi) \to ((\psi \to (\varphi \to \neg\psi)) \to (\psi \to \neg\psi))\ Ax_2$
6. $(\psi \to (\varphi \to \neg\psi)) \to (\psi \to \neg\psi)\ MP\ 2,5$
7. $\psi \to \neg\psi\ MP\ 4,6$
8. $\psi \to (\psi \to \psi)\ Ax_1$
9. $(\psi \to (\psi \to \psi)) \to ((\psi \to ((\psi \to \psi) \to \psi)) \to (\psi \to \psi))\ Ax_2$
10. $(\psi \to ((\psi \to \psi) \to \psi)) \to (\psi \to \psi)\ MP\ 8,9$
11. $\psi \to ((\psi \to \psi) \to \psi)\ Ax_1$
12. $\psi \to \psi\ MP\ 10,11$
13. $(\psi \to \psi) \to ((\psi \to \neg\psi) \to \neg\psi)\ Ax_9$
14. $(\psi \to \neg\psi) \to \neg\psi\ MP\ 12,13$
15. $\neg\psi\ MP\ 7,14$

Нека сега разгледаме следната двуконтекстна система:

$$A_i = \{\langle c_i, ist(c_j, \varphi) \rightarrow \neg\psi \rangle\}$$
$$A_j = \{\langle c_j, ist(c_i, \psi) \rightarrow \varphi \rangle\}$$

На практика ситуацията е същата като по-горе, но формулите φ и ψ тук са в различни контексти. Да се опитаме да изведем ψ от тази двуконтекстна система без помощта на правилото PE. Доказателството би изглеждало така:

1. $\langle c_i, ist(c_j, \varphi) \rightarrow \neg\psi \rangle$
2. $\langle c_j, ist(c_i, \psi) \rightarrow \varphi \rangle$
3. $\langle c_i, (ist(c_j, \varphi) \rightarrow \neg\psi) \rightarrow (\psi \rightarrow (ist(c_j, \varphi) \rightarrow \neg\psi)) \rangle$ XI_1
4. $\langle c_i, \psi \rightarrow (ist(c_j, \varphi) \rightarrow \neg\psi) \rangle$ MP $1, 3$
5. $\langle c_i, (\psi \rightarrow ist(c_j, \varphi)) \rightarrow ((\psi \rightarrow (ist(c_j, \varphi) \rightarrow \neg\psi)) \rightarrow (\psi \rightarrow \neg\psi)) \rangle$ XI_2

Оттук обаче не можем да продължим. Би трябвало да можем да приложим MP към 2 и 5, както направихме в безконтекстния случай. Интуитивно ни е ясно, че от $\langle c_i, \psi \rangle$ следва $\langle c_j, ist(c_i, \psi) \rangle$, откъдето $\langle c_j, \varphi \rangle$, оттам $\langle c_i, ist(c_j, \varphi) \rangle$ и накрая $\langle c_i, \neg\psi \rangle$, т.е. имаме $\langle c_i \psi \rightarrow \neg\psi \rangle$. Нещо повече, ако ψ е изводимо в c_i, тогава без никакви условности е изводимо и $\neg\psi$:

1. $\langle c_i, ist(c_j, \varphi) \rightarrow \neg\psi \rangle$
2. $\langle c_j, ist(c_i, \psi) \rightarrow \varphi \rangle$
3. $\langle c_i, \psi \rangle$
4. $\langle c_j, ist(c_i, \psi) \rangle$ $SF3$
5. $\langle c_j, \varphi \rangle$ MP $2, 4$
6. $\langle c_i, ist(c_j, \varphi) \rangle$ $SF5$
7. $\langle c_i, \neg\psi \rangle$ MP $1, 6$

Разсъждавайки аналогично на теоремата за дедукцията, би следвало да предположим, че след като $MCS \cup \{\langle c_i, \psi \rangle\} \vdash \langle c_i, \neg\psi \rangle$, то би трябвало и $MCS \vdash \langle c_i, \psi \rightarrow \neg\psi \rangle$. Оказва се, че в този, както и в други случаи, логическите аксиоми не могат да ни свършат работа - има формули, които бихме искали да изведем, но не можем. Ето тук идва на помощ правилото PE.

Ако бяхме на мястото на агента c_i бихме разсъждавали така: Да допуснем, че е вярно ψ. Тогава, ако попитаме c_j дали според него е вярно φ, той ще ни каже „да„. Но ние знаем, че ако според c_j е вярно φ, тогава е вярно $\neg\psi$. Получаваме противоречие. Следователно допускането, че ψ е истина не е вярно, откъдето е вярно $\neg\psi$. Формално доказателството изглежда така:

1. $\langle c_i, ist(c_j, \varphi) \to \neg\psi \rangle$
2. $\langle c_j, ist(c_i, \psi) \to \varphi \rangle$
3. $\langle c_i, \psi \rangle P$
4. $\langle c_j, ist(c_i, \psi) \rangle \ SF3$
5. $\langle c_j, \varphi \rangle \ MP\ 2, 4$
6. $\langle c_i, ist(c_j, \varphi) \rangle \ SF5$
7. $\langle c_i, \neg\psi \rangle \ MP\ 1, 6$
8. $\langle c_i, \psi \to \neg\psi \rangle \ PE\ 3$
9. $\langle c_i, \psi \to (\psi \to \psi) \rangle \ XI_1$
10. $\langle c_i, (\psi \to (\psi \to \psi)) \to ((\psi \to ((\psi \to \psi) \to \psi)) \to (\psi \to \psi)) \rangle \ XI_2$
11. $\langle c_i, (\psi \to ((\psi \to \psi) \to \psi)) \to (\psi \to \psi) \rangle \ MP\ 9, 10$
12. $\langle c_i, \psi \to ((\psi \to \psi) \to \psi) \rangle \ XI_1$
13. $\langle c_i, \psi \to \psi \rangle \ MP\ 11, 12$
14. $\langle c_i, (\psi \to \psi) \to ((\psi \to \neg\psi) \to \neg\psi) \rangle \ XI_9$
15. $\langle c_i, (\psi \to \neg\psi) \to \neg\psi \rangle \ MP\ 13, 14$
16. $\langle c_i, \neg\psi \rangle \ MP\ 8, 15$

Този пример демонстрира смисъла на правилото PE. То ни дава възможност да провеждаме разсъждения от типа на: „Да допуснем α. Тогава..., откъдето следва β. Следователно е вярно, че $\alpha \to \beta$.„

4.1.2 NK-извод

Аналогично на логическите аксиоми и правилата за FC-извод, преформулираме правилата за естествена дедукция според синтаксиса на многоконтекстните системи:

$$AI: \quad \frac{\langle c, \alpha \rangle \quad \langle c, \beta \rangle}{\langle c, \alpha \wedge \beta \rangle} \qquad AE: \frac{\langle c, \alpha \wedge \beta \rangle}{\langle c, \alpha \rangle} \quad \frac{\langle c, \alpha \wedge \beta \rangle}{\langle c, \beta \rangle}$$

$$OI: \quad \frac{\langle c,\alpha\rangle}{\langle c,\alpha\vee\beta\rangle} \quad \frac{\langle c,\beta\rangle}{\langle c,\alpha\vee\beta\rangle} \qquad OE: \frac{\langle c,\alpha\vee\beta\rangle \quad \begin{matrix}[\langle c,\alpha\rangle]\\ \langle c,\gamma\rangle\end{matrix} \quad \begin{matrix}[\langle c,\beta\rangle]\\ \langle c,\gamma\rangle\end{matrix}}{\langle c,\gamma\rangle}$$

$$II: \quad \frac{\begin{matrix}[\langle c,\alpha\rangle]\\ \langle c,\beta\rangle\end{matrix}}{\langle c,\alpha\to\beta\rangle} \qquad IE: \frac{\langle c,\alpha\rangle \quad \langle c,\alpha\to\beta\rangle}{\langle c,\beta\rangle}$$

$$NI: \quad \frac{\begin{matrix}[\langle c,\alpha\rangle]\\ \langle c,\bot\rangle\end{matrix}}{\langle c,\neg\alpha\rangle} \qquad NE: \frac{\langle c,\alpha\rangle \quad \langle c,\neg\alpha\rangle}{\langle c,\bot\rangle}$$

$$FI: \quad \frac{\langle c,\alpha(a)\rangle}{\langle c,(\forall x)\alpha(x)\rangle} \qquad FE: \frac{\langle c,(\forall x)\alpha(x)\rangle}{\langle c,\alpha(t)\rangle}$$

$$EI: \quad \frac{\langle c,\alpha(t)\rangle}{\langle c,(\exists x)\alpha(x)\rangle} \qquad EE: \frac{\langle c,(\exists x)\alpha(x)\rangle \quad \begin{matrix}[\langle c,\alpha(a)\rangle]\\ \langle c,\beta\rangle\end{matrix}}{\langle c,\beta\rangle}$$

$$R1: \quad \frac{\langle c,\bot\rangle}{\langle c,\alpha\rangle} \qquad R2: \frac{\langle c,\top\rangle}{\langle c,\alpha\vee\neg\alpha\rangle}$$

Забележка. Тук правилото IE е всъщност Modus ponens, FI е Generalization, а II е правилото Premiss elimination. Когато използваме тези правила в логиката NK, ще предпочитаме да следваме наложилите се в тази логика имена IE, FI и II.

Δ_{NK} го дефинираме като множеството от горните правила за извод:

$$\Delta_{NK} = \{AI, AE, OI, OE, II, IE, NI, NE, FI, FE, EI, EE, R1, R2\}$$

Следвайки напълно конвенцията на логиката NK би трябвало да дефинираме извода по правилата Δ_{NK} като дърво, а не като редица. Всъщност, дали доказателството представлява дърво или редица е без значение, защото, както ще видим в следващия раздел, двете са еквивалентни.

4.1.3 Еквивалентност на FC-извода и NK-извода

Дефиниция 4.1. *Дърво на извод на φ от MCS* ще наричаме дърво със следните свойства:

1. То е крайно и непразно.
2. В корена му стои формулата φ.
3. В листата му стоят контекстни аксиоми (предпоставки) или произволни формули, наречени *допускания*.

4. Формулата във всеки възел е резултат от прилагането на някое от правилата за извод върху формулите, стоящи във възлите, които са преки наследници на този възел.

5. В дървото няма неелиминирани допускания.

Дефиниция 4.2. *Сегмент в една MCS* ще наричаме дърво на извод на една формула φ от няколко други $\varphi_1, ..., \varphi_n$ по някое контекстно правило за извод (без правилата SF), в което освен контекстни аксиоми, предпоставки могат да бъдат и ist-формули, и не е задължително да са елиминирани всички допускания. Формулата φ ще наричаме *заключение на сегмента*, а предпоставките и допусканията на дървото на извод ще наричаме съответно *предпоставки и допускания на сегмента.*

Лема 4.1. Всички формули в един сегмент са от един и същ контекст.

Доказателство. Разглеждаме формулата в един произволен възел. Тъй като тя е получена по някое контекстно правило за извод, то нейните предпоставки принадлежат на същия контекст. Започвайки от корена, получаваме, че всички предпоставки на корена са от неговия контекст. Прилагайки индуктивно наблюдението по-горе върху всички преки наследници на корена, получаваме, че всички преки наследници на преките наследници на корена са от неговия контекст. Продължаваме така, докато стигнем до листата. Процесът е краен, защото дървото е крайно. $\diamondsuit$

Дефиниция 4.3. *Скелет на извод на φ от MCS* ще наричаме дърво със следните свойства:

1. Всеки възел на скелета представлява сегмент, всяка предпоставка на който е или контекстна аксиома, или допускане, или е от вида $\langle c_k, ist(c_l, \alpha)\rangle$, където $\langle c_l, \alpha\rangle$ е заключение на някой от наследниците на сегмента.

2. Заключението на корена на скелета е формулата φ.

3. Неелиминираните допускания на един сегмент се считат за допускания на неговия родител.

4. В скелета няма неелиминирани допускания.

Предпоставките на сегментите на скелета, които са контекстни аксиоми, ще наричаме *предпоставки на скелета*, а допусканията на всички сегменти ще наричаме *допускания на скелета.*

Лема 4.2. Всяка крайна редица, представляваща доказателство на φ от една многоконтекстна система MCS, може да се преобразува в дърво на извод и обратно, всяко крайно дърво на извод на φ от MCS може да се преобразува в редица, която е доказателство на φ от MCS.

Доказателство. Доказателството е конструктивно.

(i) редица $\rightarrow$ дърво на извод

Нека имаме редицата $\varphi_1, \varphi_2, ..., \varphi_n$. Разглеждаме следния алгоритъм:

1. Ако φ_n е получено от правило, приложено върху формулите $\varphi_{i_1}, \varphi_{i_2}, \varphi_{i_3}, ..., \varphi_{i_n}$, $i_j < i_k, i_n < n$, то конструираме дърво с корен φ_n и наследници резултата от алгоритъма, приложен към редиците $\varphi_1, ..., \varphi_{i_1}$, $\varphi_1, ..., \varphi_{i_2}$, ..., $\varphi_1, ..., \varphi_{i_n}$.

2. Ако φ_n е аксиома или допускане, тогава конструираме листото φ_n.

Очевидно полученото дърво отговаря на изискванията за дърво на извод на φ от MCS. То е крайно, защото всяко рекурсивно изпълнение на алгоритъма се прилага върху крайни редици, които са строго по-къси от предходната. Всички допускания в него са елиминирани, защото това са същите допускания, които присъстват и са елиминирани в редицата.

(ii) дърво на извод $\rightarrow$ редица

Конструираме редицата отзад-напред: Накрая поставяме формулата φ, която е коренът на дървото. Пред нея поставяме всички преки наследници на φ, пред тях всички техни преки наследници и така, докато стигнем до листата. Очевидно по този начин всяка формула от дървото се поставя в редицата след предпоставките и допусканията, от които е изведена, и преди заключението, на което е предпоставка. Процесът отново е краен, защото дървото на извод е крайно. $\diamondsuit$

Лема 4.3. (сгъване на дърво до скелет) Всяко дърво на извод на φ от MCS може да се преобразува в сегмент със заключение φ и същите предпоставки и допускания.

Доказателство. Доказателството е конструктивно. Нека имаме дърво на извод на φ от MCS. Разглеждаме следните алгоритми:

Алгоритъм 1 - Конструиране на сегмент:

1. Ако φ е аксиома или допускане, връщаме „листото„ φ - дърво с единствен елемент φ.

2. Ако φ е от вида $\langle c_i, ist(c_j, \alpha)\rangle$ и има единствен наследник, който е от вида $\langle c_j, \alpha\rangle$, то φ е получено по правилото SF, следователно обявяваме този възел за терминален и връщаме като резултат „листото„ φ.

3. Ако φ не е от горните два вида, тогава тази формула е получена от няколко други по някое контекстно правило за извод. Връщаме като резултат дърво с корен φ, към което прикачаме резултатите от алгоритъма приложен върху нейните наследници в дървото на извод.

Непосредствено от дефиницията на сегмент следва, че дървото, построено по този алгоритъм, е сегмент. Той е краен, защото дървото на извод е крайно.

Алгоритъм 2 - Конструиране на скелет:

1. Конструираме скелет с корен сегмента, построен според алгоритъм 1, към който прикачаме резултатите от алгоритъм 2, приложен върху

наследниците на терминалните възли, определени в алгоритъм 1.

Очевидно дървото, получено по алгоритъм 2 е крайно, защото е крайно дървото на извод на φ от MCS. Следователно, от дефиницията за скелет следва, че резултатът от алгоритъм 2 е скелет. $\diamondsuit$

Лема 4.4. (разгъване на скелет до дърво) Всеки скелет със заключение φ може да се преобразува в дърво на извод на φ от MCS, което има същите предпоставки и допускания като скелета.

Доказателство. Доказателството отново е конструктивно:

Алгоритъм 3 - Конструиране на дърво:

1. Нека предпоставките на корена на скелета, които не са аксиоми, са $\varphi_1, \varphi_2, ..., \varphi_n$. Тъй като те са предпоставки на сегмент и не са аксиоми, остава да са от вида $\varphi_i = \langle c_i, ist(c_j, \alpha_i)\rangle$. От дефиницията на скелет следва, че за всяко i съществува наследник на сегмента, чието заключение е $\langle c_j, \alpha_i\rangle$, т.е. формулата φ_i е получена от $\langle c_j, \alpha_i\rangle$ по правилото SF. Следователно, ако към листата φ_i на дървото в корена на скелета прикачим резултата от алгоритъма приложен върху съответния наследник в скелета, то полученото дърво ще е дърво на извод на φ от MCS. Всички допускания в дървото ще са елиминирани, защото в скелета няма неелиминирани допускания. $\diamondsuit$

Лема 4.5. Всяко от правилата

$$\begin{array}{rl} (i) & II \\ (ii) & NI \\ (iii) & OE \\ (iv) & EE \end{array}$$

може да се изрази с правилата Δ_{FC}.

Доказателство. За всяко от правилата $(i), (ii), (iii)$ и (iv) ще построим редица на доказателство на заключението от предпоставките му, в която се елиминират същите допускания, които се елиминират в него:

(i) Това е правилото PE.

(ii)

$$\begin{array}{rl} 1. & \langle c_k, \alpha\rangle\ P \\ & ... \\ n. & \langle c_k, \perp\rangle ? \\ n+1. & \langle c_k, \alpha \to \perp\rangle\ PE\ 1 \\ n+2. & \langle c_k, \perp \to \beta\rangle\ XI \end{array}$$

$n+3.$ $\langle c_k, \bot \to \neg\beta \rangle$ XI

$n+4.$ $\langle c_k, (\bot \to \beta) \to (\alpha \to (\bot \to \beta)) \rangle$ XI_1

$n+5.$ $\langle c_k, (\bot \to \neg\beta) \to (\alpha \to (\bot \to \neg\beta)) \rangle$ XI_1

$n+6.$ $\langle c_k, \alpha \to (\bot \to \beta) \rangle$ $MP\ n+2, n+4$

$n+7.$ $\langle c_k, \alpha \to (\bot \to \neg\beta) \rangle$ $MP\ n+3, n+5$

$n+8.$ $\langle c_k, (\alpha \to \bot) \to ((\alpha \to (\bot \to \beta)) \to (\alpha \to \beta)) \rangle$ XI_2

$n+9.$ $\langle c_k, (\alpha \to \bot) \to ((\alpha \to (\bot \to \neg\beta)) \to (\alpha \to \neg\beta)) \rangle$ XI_2

$n+10.$ $\langle c_k, (\alpha \to (\bot \to \beta)) \to (\alpha \to \beta) \rangle$ $MP\ n+1, n+8$

$n+11.$ $\langle c_k, (\alpha \to (\bot \to \neg\beta)) \to (\alpha \to \neg\beta) \rangle$ $MP\ n+1, n+9$

$n+12.$ $\langle c_k, \alpha \to \beta \rangle$ $MP\ n+6, n+10$

$n+13.$ $\langle c_k, \alpha \to \neg\beta \rangle$ $MP\ n+7, n+11$

$n+14.$ $\langle c_k, (\alpha \to \beta) \to ((\alpha \to \neg\beta) \to \neg\alpha) \rangle$ XI_9

$n+15.$ $\langle c_k, (\alpha \to \neg\beta) \to \neg\alpha \rangle$ $MP\ 12, 14$

$n+16.$ $\langle c_k, \neg\alpha \rangle$ $MP\ 13, 15$

(iii)

$1.$ $\langle c_k, \alpha \rangle$ P

...

$i-1.$ $\langle c_k, \gamma \rangle$?

$i.$ $\langle c_k, \beta \rangle$ P

...

$n-1.$ $\langle c_k, \gamma \rangle$?

$n.$ $\langle c_k, \alpha \vee \beta \rangle$

$n+1.$ $\langle c_k, \alpha \to \gamma \rangle$ $PE\ 1$

$n+2.$ $\langle c_k, \beta \to \gamma \rangle$ $PE\ i$

$n+3.$ $\langle c_k, (\alpha \to \gamma) \to ((\beta \to \gamma) \to (\alpha \vee \beta \to \gamma)) \rangle$ XI_8

$n+4.$ $\langle c_k, (\beta \to \gamma) \to (\alpha \vee \beta \to \gamma) \rangle$ $MP\ n+1, n+3$

$n+5.$ $\langle c_k, \alpha \vee \beta \to \gamma \rangle$ $MP\ n+2, n+4$

$n+6.$ $\langle c_k, \gamma \rangle$ $MP\ n, n+5$

(iv)

$1.$ $\langle c_k, \alpha(y) \rangle$ P

...

$n.$ $\langle c_k, \beta \rangle$?

$n+1.$ $\langle c_k, \alpha(y) \to \beta \rangle$ $PE\ 1$

$n+2.$ $\langle c_k, \neg\forall x \neg\alpha(x)\rangle$

$n+3.$ $\langle c_k, (\alpha(y) \to \beta) \to ((\alpha(y) \to \neg\beta) \to \neg\alpha(y))\rangle$ XI_9

$n+4.$ $\langle c_k, (\alpha(y) \to \neg\beta) \to \neg\alpha(y)\rangle$ MP $n+1, n+3$

$n+5.$ $\langle c_k, ((\alpha(y) \to \neg\beta) \to \neg\alpha(y)) \to$
$(\neg\beta \to ((\alpha(y) \to \neg\beta) \to \neg\alpha(y)))\rangle$ XI_1

$n+6.$ $\langle c_k, \neg\beta \to ((\alpha(y) \to \neg\beta) \to \neg\alpha(y))\rangle$ MP $n+4, n+5$

$n+7.$ $\langle c_k, (\neg\beta \to (\alpha(y) \to \neg\beta)) \to$
$((\neg\beta \to ((\alpha(y) \to \neg\beta) \to \neg\alpha(y))) \to (\neg\beta \to \neg\alpha(y)))\rangle$ XI_2

$n+8.$ $\langle c_k, \neg\beta \to (\alpha(y) \to \neg\beta)\rangle$ XI_1

$n+9.$ $\langle c_k, (\neg\beta \to ((\alpha(y) \to \neg\beta) \to \neg\alpha(y))) \to$
$(\neg\beta \to \neg\alpha(y))\rangle$ MP $n+7, n+8$

$n+10.$ $\langle c_k, \neg\beta \to \neg\alpha(y)\rangle$ MP $n+6, n+9$

$n+11.$ $\langle c_k, \forall y(\neg\beta \to \neg\alpha(y))\rangle$ Gen 10

$n+12.$ $\langle c_k, \forall y(\neg\beta \to \neg\alpha(y)) \to (\neg\beta \to \forall y \neg\alpha(y))\rangle$ XI_{12}

$n+13.$ $\langle c_k, \neg\beta \to \forall y \neg\alpha(y)\rangle$ MP $n+11, n+12$

$n+14.$ $\langle c_k, (\neg\forall x \neg\alpha(x)) \to (\neg\beta \to (\neg\forall x \neg\alpha(x)))\rangle$ XI_1

$n+15.$ $\langle c_k, \neg\beta \to (\neg\forall x \neg\alpha(x))\rangle$ MP $n+2, n+14$

$n+16.$ $\langle c_k, (\neg\beta \to \forall y \neg\alpha(y)) \to ((\neg\beta \to \neg\forall x \alpha(x)) \to \neg\neg\beta)\rangle$ XI_9

$n+17.$ $\langle c_k, (\neg\beta \to \neg\forall x \alpha(x)) \to \neg\neg\beta\rangle$ MP $n+13, n+16$

$n+18.$ $\langle c_k, \neg\neg\beta\rangle$ MP $n+15, n+17$

$n+19.$ $\langle c_k, \neg\neg\beta \to \beta\rangle$ XI_{10}

$n+20.$ $\langle c_k, \beta\rangle$ MP $n+18, n+19$

Съгласно лема 4.2 можем да преобразуваме тези редици в дървета на извод със същите предпоставки, заключения и елиминирани допускания. С това лемата е доказана. $\diamondsuit$

Лема 4.6. За всеки сегмент с изводи по правилата Δ_{NK}, съществува сегмент със същите предпоставки, елиминирани допускания и заключение, но с изводи, направени по правилата Δ_{FC}, и обратно, за всеки сегмент с изводи по правилата Δ_{FC}, съществува сегмент със същите предпоставки, елиминирани допускания и заключение, но с изводи, направени по правилата Δ_{NK}.

Доказателство. $(i) NK \to FC$
Нека предпоставките на сегмента са $\varphi_1 = \langle c_k, \alpha_1\rangle, ..., \varphi_n = \langle c_k, \alpha_n\rangle$, допусканията са $\psi_1 = \langle c_k, \beta_1\rangle, ..., \psi_m = \langle c_k, \beta_m\rangle$, а заключението е $\varphi = \langle c_k, \alpha\rangle$. Всички те са формули на един контекст - това го доказахме в лема 4.1.

Да разгледаме следното съответствие: Всяка формула $\chi = \langle c_k, \gamma\rangle$ в дървото я заместваме с формулата γ. Никъде в сегмента не се използва правилото SF - по дефиниция в него се използват само контекстни правила. Както отбелязахме в раздели 4.1.1 и 4.1.2, между класическите и контекстните правила съществува еднозначно взаимно съответствие. Например, на правилото

$$\frac{\alpha \qquad \alpha \to \beta}{\beta}$$

съответства правилото

$$\frac{\langle c_k, \alpha\rangle \qquad \langle c_k, \alpha \to \beta\rangle}{\langle c_k, \beta\rangle}$$

и обратно.

От теорема 3.3 имаме, че за всяко доказателство по правилата на логиката NK съществува еквивалентно доказателство по правилата на логиката FC. В частност имаме, че за всяко от правилата на NK съществува редица, която е доказателство на неговото заключение от неговите предпоставки по правилата и аксиомите на FC. Но поради еднозначното съответствие между правилата FC и Δ_{FC}, заменяйки всяка от формулите в това доказателство с нейния аналог в контекста c_k и сменяйки мотива за използване на всяка аксиома i със съответното правило XI_i, което я въвежда, ние получаваме редица, която е доказателство на правилата Δ_{NK} по правилата Δ_{FC} за контекста c_k, т.е. правилата Δ_{NK} могат да се изразят чрез правилата Δ_{FC}, когато те се отнасят до един контекст.

Това все още не означава, правилата Δ_{NK} могат винаги да се изразят чрез Δ_{FC}. Достатъчно е да си припомним контрапримера от 4.1.1. Причината се крие в правилата II, NI, OE и EE. Характерното за тези правила е, че те елиминират допускания. Въпреки че допусканията и техните следствия са в същия контекст, във веригата от някое допускане до неговото следствие може да се премине от един контекст в друг. Следователно тук не можем да приложим аналогията с безконтекстните системи. В лема 4.5 обаче ние показахме, че тези четири правила са също изразими с правилата Δ_{FC}. Това ни дава основание окончателно да твърдим, че за всеки сегмент с извод по правилата Δ_{NK} съществува еквивалентен на него сегмент с извод по правилата Δ_{FC}.

$(ii) FC \to NK$

В обратната посока доказателството е аналогично. От теорема 3.3 следва, че всяка аксиома на FC е доказуема със средствата на NK. Използвайки взаимно еднозначното съответствие между правилата и формулите на контекста c_k и тези в безконтекстния случай, получаваме, че всяко от правилата XI_i е изразимо чрез Δ_{NK}. Както вече отбелязахме, останалите правила от Δ_{FC} са подмножество на Δ_{NK}. Следователно, за всеки сегмент с извод по правилата Δ_{FC} съществува еквивалентен на него сегмент с извод по правилата Δ_{NK}. С това лемата е доказана. $\diamondsuit$

Теорема 4.1. **(еквивалентност на FC и NK)** Ако $c_i = \langle L, A_i, \Delta_i \rangle$ е един примитивен контекст на системата $MCS = \langle \{c_i\}_{i \in I}, \Delta \rangle$, то многоконтекстните системи $MCS_{FC} = MCS(i|\langle L, A_i, \Delta_{FC} \rangle)$ и $MCS_{NK} = MCS(i|\langle L, A_i, \Delta_{NK} \rangle)$ са еквивалентни, т.е. $MCS_{FC} \simeq MCS_{NK}$.

Доказателство. ($\Rightarrow$) Нека $MCS_{FC} \vdash_{FC} \varphi$. Това означава, че съществува крайна редица, завършваща с φ, всеки член на която е или контекстна аксиома, или допускане, което се елиминира, или се получава от предишни по някое от правилата MP, Gen, PE или $XI_i, i = 1..12$. Преобразуваме тази редица в дърво на извод на φ от MCS_{FC}, съгласно лема 4.2, а полученото дърво го свиваме до скелет, според лема 4.3. Съгласно лема 4.6, заменяме дървото на извод на всеки сегмент, асоцииран с контекста c_i, с еквивалентно на него дърво на извод в NK. Разгъваме получения скелет според лема 4.4 и получаваме доказателство на φ от MCS_{NK}

($\Leftarrow$) Обратно: Нека $MCS_{NK} \vdash_{NK} \varphi$. Това означава, че съществува крайна редица, завършваща с φ, всеки член на която е или контекстна аксиома, или допускане, което се елиминира, или се получава от предишни по някое от правилата Δ_{NK}. Преобразуваме тази редица в дърво на извод на φ от MCS_{NK}, съгласно лема 4.2, а полученото дърво го свиваме до скелет, според лема 4.3. Съгласно лема 4.6, заменяме дървото на извод на всеки сегмент, асоцииран с контекста c_i, с еквивалентно на него дърво на извод в FC. Разгъваме получения скелет според лема 4.4 и получаваме доказателство на φ от MCS_{FC} $\diamondsuit$

4.2 Изводимост

Лема 4.7. Нека A е едно множество от формули на класическата логика от първи ред, такова че $A = \{P(a_i, b_i)\}_{i \in K}$, където a_i и b_i са индивидуални константи, P е двуместен предикатен символ, а K - множество от индекси. Тогава единственият литерал, който може да се изведе по правилата на логиката NK от множеството A и не принадлежи на A е $\top$.

Доказателство. Ясно е, че $\top$ е литерал изводим от всяко множество, дори от празното. Нека φ е произволен литерал, различен от $\top$, изводим от A по правилата на NK, т.е. $A \vdash_{NK} \varphi$. От теорема 3.3 имаме, че $A \vdash_{NK} \varphi \Leftrightarrow A \vdash_{FC} \varphi$, а от теорема 3.2 - $A \vdash_{FC} \varphi \Leftrightarrow A \models \varphi$. Следователно, всеки модел на A е модел и на φ. Тъй като φ е литерал, той има вида $P(x, y), \neg P(x, y), Q(x_1, ..., x_n)$ или $\neg Q(x_1, ..., x_n)$, където $P \neq Q$, а $x, y, x_1, ..., x_n$ са индивидуални променливи или константи. Да разгледаме всеки от случаите:

1. $\varphi = Q(x_1, ..., x_n)$

Разглеждаме интерпретация $I = \langle D, \rho \rangle$, при която $\rho(P) = \top$ (абсолютната истина), а $\rho(Q) = \bot$ (абсолютната лъжа). Ако $\psi = P(a_i, b_i) \in A$, то за произволна оценка V

$$V(\psi) = V(P(a_i, b_i)) = \rho(P)(V(a_i), V(b_i)) = true.$$

Следователно, I е модел на A. За произволна оценка V обаче

$$V(\varphi) = V(Q(x_1, ..., x_n)) = \rho(Q)(V(x_1), ..., V(x_n)) = false,$$

откъдето I не е модел на φ. Намерихме интерпретация, която е модел на A, но не е модел на φ. Това противоречие се дължи на допускането, че φ има вида $Q(x_1, ..., x_n)$.

2. $\varphi = \neg Q(x_1, ..., x_n)$

Този път разглеждаме интерпретация $I = \langle D, \rho \rangle$, при която $\rho(P) = \rho(Q) = \top$. Тогава за произволни V и i имаме

$$\begin{aligned} V(P(a_i, b_i)) = & \quad \rho(P)(V(a_i), V(b_i)) = true \\ V(\varphi) = & \quad V(\neg Q(x_1, ..., x_n)) = \neg V(Q(x_1, ..., x_n)) = \\ = & \quad \neg \rho(Q)(V(x_1), ..., V(x_n)) = \neg true = false \end{aligned}$$

Отново намерихме интерпретация, която е модел на A, но не е модел на φ. Следователно φ не е от вида $\neg Q(x_1, ..., x_n)$.

3. $\varphi = \neg P(x, y)$

Разглеждаме интерпретация $I = \langle D, \rho \rangle$, при която $\rho(P) = \top$. За произволни V и i имаме

$$V(P(x, y)) = \rho(P)(V(x), V(y)) = true \Rightarrow V(\neg P(x, y)) = false$$

Следователно φ няма вида $\neg P(x, y)$.

4. $\varphi = P(x, y)$, където x или y е индивидуална променлива

Нека за определеност x е индивидуална променлива. Случаят с y е аналогичен. Нека $c_t \in D$ и $c_f \in D$. Разглеждаме интерпретация $I = \langle D, \rho \rangle$, при която $\rho(P) = \{(c_t, c_t)\}, \rho(a_i) = c_t$ и $\rho(b_i) = c_t$ за $i \in K$. Разглеждаме оценка V в I, такава че $V(x) = c_f$ за всяка индивидуална променлива x. Тогава

$$\begin{aligned} V(P(a_i, b_i)) \quad & = \rho(P)(V(a_i), V(b_i)) = \rho(P)(\rho(a_i), \rho(b_i)) = \\ & = \rho(P)(c_t, c_t) = true \\ V(P(x, y)) \quad & = \rho(P)(V(x), V(y)) = \rho(P)(c_f, V(y)) = false \end{aligned}$$

Намерихме интерпретация, която е модел на A, но не е модел на φ, следователно φ не е от вида 4.

5. $\varphi = P(c, d)$, където c и d са индивидуални константи
Ако $c = a_i$ и $d = b_i$ за някое $i \in K$, тогава е ясно, че φ е изводимо от A. Да допуснем, че не съществува $i \in K$, за което едновременно $c = a_i$ и $d = b_i$. Тогава разглеждаме интерпретацията $I = \langle D, \rho \rangle$, при която $\rho(P) = \{(c_t, c_t), (c_t, c_f), (c_f, c_t)\}$, $\rho(a_i) = \rho(b_i) = c_t$ за тези a_i и b_i, които са различни от c и d, а за c и d $\rho(c) = \rho(d) = c_f$. В такъв случай за произволна оценка V в I имаме $V(P(a_i, b_i)) = \rho(P)(\rho(a_i), \rho(b_i)) = true$. Това е така, защото, както вече допуснахме по-горе, поне едно от a_i и b_i е различно от c и d, следователно поне един от аргументите на $\rho(P)$ има стойност c_t, а релацията $\rho(P)$ съдържа всички двойки, в които c_t участва поне веднъж. От друга страна $V(P(c, d)) = \rho(P)(\rho(c), \rho(d)) = \rho(P)(c_f, c_f)) = false$, защото $(c_f, c_f) \notin \rho(P)$. Следователно не всеки модел на A е модел на φ. Това противоречие се дължи на допускането, че двойката (c, d) не е измежду двойките аргументи на P във формулите от A.

Тъй като изчерпахме всички възможни случаи за φ, можем да твърдим, че единственият литерал, изводим от множеството A, който не принадлежи на това множество е абсолютната истина $\top$. С това лемата е доказана. $\diamondsuit$

Следствие. Ако $A = \{\top\} \cup \{P(a_i, b_i)\}_{i \in K}$, то $A \equiv Lit(Th(A))$.

Да означим за удобство с SF_k правилото SF, в което контекстът на заключението е фиксиран на k, т.е.

$$SF_k : \frac{\langle c_i, \alpha \rangle}{\langle c_k, ist(c_i, \alpha) \rangle},$$

където k е фиксирано, а i е произволно.

Лема 4.8. Всички атомарни формули в импортната част на един контекст c_k, освен $\langle c_k, \top \rangle$, са от вида $\langle c_k, ist(c_i, \alpha_i) \rangle$ и могат да се изведат като заключения на правило SF_k.

Доказателство. Нека разглежданата многоконтекстна система е $MCS = \langle \{c_i\}_{i \in I}, \Delta \rangle$ и c_k е един неин контекст. За определеност ще приемем, че $\Delta_k = \Delta_{NK}$. Нека $\varphi = \langle c_k, \alpha \rangle$ е една произволна формула от импортната част c_k, т.е. $\varphi \in Imp(c_k)$. Това означава, че тя се извежда от многоконтекстната система $MCS_0 = MCS(c_k | \langle L, \phi, \Delta_{NK} \rangle)$. Следователно, съществува дърво на извод на φ от MCS_0, чиито листа са само допускания, които по пътя до корена са елиминирани. Преобразуваме това дърво по следния начин:

Тръгваме от корена и разглеждаме всеки клон. Ако в даден клон не се използва правилото SF_k, тогава не променяме нищо по него. Ако е използвано правилото SF_k, то маркираме възела, който е заключение на това правило SF_k, което е най-близо до корена, и отсичаме поддървото с корен неговия наследник. Ясно е, че в този възел стои ist-формула. Ясно е също, че ако при отсичането премахнем някое правило за елиминиране на допускане, то с това правило ще отпадне и съответното допускане, защото то е листо на същото поддърво. Това означава, че в полученото дърво няма неелиминирани допускания. Следователно, листата на новополученото дърво са или елиминирани допускания, или ist-формули.

Освен това, никъде в дървото не се прилага правило SF. Да допуснем, че някъде в него има произволно правило SF. Ако то е от вида SF_k, тогава ще излезе, че не сме премахнали най-близкото до корена правило. Ако има правило от вида $SF_l, l \neq k$, тогава ще напуснем контекста c_k. Но тъй като формулата φ е от c_k, то за да стигнем до нея, ще се наложи отново да влезнем в този контекст, а това може да стане единствено с правилото SF_k. Това отново противоречи на факта, че сме отсекли дървото от най-близкото до корена използване на SF_k.

Нека $S = \{\varphi_i | i = 1..n\}$ е множеството от всички маркирани ist-формули. В такъв случай, можем да разглеждаме полученото дърво, като дърво на извод на φ от $MCS' = MCS(c_k | \langle L, S, \Delta_{NK} \rangle)$, в което всички ist-формули по листата са поставени с аргумента, че са контекстни аксиоми на MCS'.

Да допуснем сега, че формулата φ е литерал. Както вече отбелязахме, в получения извод не участва правилото за смяна на контекста, т.е. използват се само правилата Δ_{NK} и всички формули са от един контекст. Да разгледаме множеството $X = \{\alpha_i | \langle c_k, \alpha_i \rangle \in S\}$. Тъй като между Δ_{NK} и правилата за извод на логиката NK има взаимно еднозначно съответствие, ако премахнем контекстната нотация на формулите в новото дърво на извод, ще получим извод на α от множеството X в класическата логика NK. (Тук ist ще го смятаме за най-обикновен двуместен предикатен символ, а формулите освен за такива, ще ги считаме и за терми.) Множеството X отговаря на условията на лема 4.7. Тогава е в сила и нейното твърдение, т.е. след като α е литерал, то $\alpha \in X$. От дефиницията на X следва, че $\varphi = \langle c_k, \alpha \rangle \in S$. Но S го дефинирахме като множество от ist-формули, които се получават по правилото SF_k. Следователно, произволен литерал, изводим в импортната част на един контекст и различен от $\top$, е ist-формула и е изводим по правилото SF_k. С това лемата е доказана. $\diamondsuit$

Теорема 4.2. (Локалност на противоречието) В импортната част на един контекст не може да се изведе противоречие.

Доказателство. Тази теорема е непосредствено следствие от лема 4.8, тъй като противоречието $\perp$ е атомарна формула. $\diamondsuit$

Теорема 4.2 представлява най-съществeният резултат, представен в тази дипломна работа. Тя гарантира пълната автономност на примитивните контексти. Изказана по-друг начин, тя гласи, че противоречието не може да се „внесе„ в никой контекст. Ако в един контекст се изведе противоречие, то това неминуемо се дължи на негова контекстна аксиома.

Едната възможност да се изведе противоречие в даден контекст е да са противоречиви част от неговите аксиоми, които не са мета-формули. Това съответства на противоречиви знания за света. Този случай е възможен при противоречива теория или при грешка в кодирането на теорията под формата на аксиоми. Въпреки че се случва, този вариант е сравнително рядък.

Съдейки по опита на хората, по-честа причина за заблуждения са неправилни знания за познанията на другите, погрешни очаквания за техните действия и т.н. В термините на многоконтекстните системи, това съответства на извеждане на противоречие от група формули, в които участват и мета-формули. Този вариант е по-вероятен, защото познанията на един контекст за друг, и въобще на един агент за друг, са много по-лабилни - тяхната истинност зависи от знанията и на двата агента.

В реална програмна реализация се очаква множеството контексти да се изменя динамично съобразно натрупаните знания и опит. При това, колкото по-близки до реалните проблеми на хората се решават, толкова по-динамична е средата. Оттук следва и по-голямата вероятност за грешка. Ако програмната системата не е осигурена против такива грешки, нейната ефективност ще е значително по-ниска. Теорема 4.2 ни гарантира, че в такива случаи, системата няма да се окаже напълно дезориентирана. Нейната стабилност ще бъде разклатена само в някакви тесни рамки; в останалите случаи тя ще запази своята работоспособност.

Да разгледаме още едно следствие от лема 4.8:

Теорема 4.3. В импортната част е в сила обратното правило за смяна на контекста

$$SB : \frac{\langle c_k, ist(c_l, \alpha)\rangle}{\langle c_l, \alpha\rangle}.$$

Доказателство. Тази теорема също е непосредствено следствие от лема 4.8, защото в нея доказахме, че ако в импортната част на един контекст изведем една формула от вида $\langle c_k, ist(c_l, \alpha)\rangle$, то тя се получава по правилото SF_k от формулата $\langle c_l, \alpha\rangle$. Следователно, винаги когато е вярна $\langle c_k, ist(c_l, \alpha)\rangle$ е вярна и формулата $\langle c_l, \alpha\rangle$. $\diamondsuit$

Свойства. В импортната част на един контекст, са в сила следните свойства на ist-формулите:

$$
\begin{array}{ll}
1.) & \langle c_k, ist(c_l, \alpha \wedge \beta) \leftrightarrow ist(c_l, \alpha) \wedge ist(c_l, \beta)\rangle \\
2.) & \langle c_k, ist(c_l, \alpha \vee \beta) \leftarrow ist(c_l, \alpha) \vee ist(c_l, \beta)\rangle \\
3.) & \langle c_k, ist(c_l, \neg\alpha) \wedge \neg ist(c_l, \bot) \leftrightarrow \neg ist(c_l, \alpha)\rangle \\
4.) & \langle c_k, ist(c_l, \alpha \rightarrow \beta) \leftrightarrow ist(c_l, \alpha) \rightarrow ist(c_l, \beta)\rangle
\end{array}
$$

Доказателство.

1.)

$(\rightarrow)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha \wedge \beta)\rangle}}{\langle c_l, \alpha \wedge \beta\rangle}\, SB}{\langle c_l, \alpha\rangle}\, AE}{\langle c_k, ist(c_l, \alpha)\rangle}\, SF \qquad \dfrac{\dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha \wedge \beta)\rangle}}{\langle c_l, \alpha \wedge \beta\rangle}\, SB}{\langle c_l, \beta\rangle}\, AE}{\langle c_k, ist(c_l, \beta)\rangle}\, SF}{\langle c_k, ist(c_l, \alpha) \wedge ist(c_l, \beta)\rangle}\, AI}{\langle c_k, ist(c_l, \alpha \wedge \beta) \rightarrow ist(c_l, \alpha) \wedge ist(c_l, \beta)\rangle}\, II\ 1
$$

$(\leftarrow)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha) \wedge ist(c_l, \beta)\rangle}}{\langle c_k, ist(c_l, \alpha)\rangle}\, AE}{\langle c_l, \alpha\rangle}\, SB \qquad \dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha) \wedge ist(c_l, \beta)\rangle}}{\langle c_k, ist(c_l, \beta)\rangle}\, AE}{\langle c_l, \beta\rangle}\, SB}{\langle c_l, \alpha \wedge \beta\rangle}\, AI}{\langle c_k, ist(c_l, \alpha \wedge \beta)\rangle}\, SF}{\langle c_k, ist(c_l, \alpha) \wedge ist(c_l, \beta) \rightarrow ist(c_l, \alpha \wedge \beta)\rangle}\, II\ 1
$$

2.)

$(\leftarrow)$

$$
\dfrac{\langle c_k, ist(c_l, \alpha) \vee ist(c_l, \beta)\rangle \qquad \dfrac{\dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha)\rangle}}{\langle c_l, \alpha\rangle}\, SB}{\langle c_l, \alpha \vee \beta\rangle}\, OI}{\langle c_k, ist(c_l, \alpha \vee \beta)\rangle}\, SF \qquad \dfrac{\dfrac{\dfrac{\overset{2}{\langle c_k, ist(c_l, \beta)\rangle}}{\langle c_l, \beta\rangle}\, SB}{\langle c_l, \alpha \vee \beta\rangle}\, OI}{\langle c_k, ist(c_l, \alpha \vee \beta)\rangle}\, SF}{\langle c_k, ist(c_l, \alpha \vee \beta)\rangle}\, OE\ 1,2
$$

($\rightarrow$) В тази посока твърдението не винаги е вярно. Ето един контрапример. Разглеждаме двуконтекстната система

$$
\begin{aligned}
MCS: \quad & A_k = \phi \\
& A_l = \{\alpha \vee \beta\}
\end{aligned}
$$

Очевидно от тази система се извежда $\langle c_l, \alpha \vee \beta \rangle$, защото това е контекстна аксиома на контекста c_l. От нея обаче по никакъв начин не може да се изведе нито $\langle c_l, \alpha \rangle$, нито $\langle c_l, \beta \rangle$.

3.)
($\rightarrow$)

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha)\rangle}}{\langle c_l, \alpha\rangle}\,SB
\qquad
\dfrac{\dfrac{\overset{2}{\langle c_k, ist(c_l, \neg\alpha) \wedge \neg ist(c_l, \perp)\rangle}}{\langle c_k, ist(c_l, \neg\alpha)\rangle}\,AE}{\langle c_l, \neg\alpha\rangle}\,SB
}{\dfrac{\langle c_l, \perp\rangle}{\langle c_k, ist(c_l, \perp)\rangle}\,SF}\,NE
\qquad
\dfrac{\overset{2}{\langle c_k, ist(c_l, \neg\alpha) \wedge \neg ist(c_l, \perp)\rangle}}{\langle c_k, \neg ist(c_l, \perp)\rangle}\,AE
}{\dfrac{\langle c_k, \perp\rangle}{\langle c_k, \neg ist(c_l, \alpha)\rangle}\,NI\ 1}\,NE
}{\langle c_k, ist(c_l, \neg\alpha) \wedge \neg ist(c_l, \perp) \rightarrow \neg ist(c_l, \alpha)\rangle}\,II\ 2
$$

($\leftarrow$) В теорема 4.2 доказахме, че в импортната част на един контекст не може да се изведе противоречие, т.е. $\langle c_k, \neg ist(c_l, \alpha)\rangle \notin Imp(c_k)$, за произволни l, k и α. Следователно импликацията в обратна посока е винаги изпълнена.

Това свойство отразява друг аспект на противоречието в многоконтекстните системи. Първо, въвеждането на символа $\perp$ в синтаксиса увеличава неговата изразителна сила. То ни дава възможност да третираме противоречието като обект и да изказваме твърдения за него. Такова е твърдението $\neg ist(c_l, \perp)$, което отразява убеждението, че контекстът c_l е непротиворечив.

Второ, свойството ни дава отговор на въпроса кога бихме могли да изведем формули от вида $\neg ist(c_l, \alpha)$ (досега показахме, че в един контекст се внасят само формули от вида $ist(c_l, \alpha)$). Това става в случаите, когато един контекст се „доверява напълно„ на друг, т.е. приеме контекстната аксиома, че вторият контекст е непротиворечив.

Това естествено е нож с две остриета. От една страна то ни позволява да извеждаме твърдения за неща, които *не са изпълнени* в даден контекст. От друга, наличието на противоречие във втория контекст, автоматично довежда до противоречие в първия:

$$\dfrac{\dfrac{\langle c_l, \bot\rangle}{\langle c_k, ist(c_l, \bot)\rangle}\,SF \qquad \langle c_k, \neg ist(c_l, \bot)\rangle}{\langle c_k, \bot\rangle}\,NE$$

В този случай противоречието в контескта c_k се дължи на неговото неправилно убеждение, че контекстът c_l е непротиворечив.

4.)
$(\rightarrow)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha \rightarrow \beta)\rangle}}{\langle c_l, \alpha \rightarrow \beta\rangle}\,SB \qquad \dfrac{\overset{2}{\langle c_k, ist(c_l, \alpha)\rangle}}{\langle c_l, \alpha\rangle}\,SB}{\langle c_l, \beta\rangle}\,IE}{\langle c_k, ist(c_l, \beta)\rangle}\,SF}{\langle c_k, ist(c_l, \alpha) \rightarrow ist(c_l, \beta)\rangle}\,II\ 2}{\langle c_k, ist(c_l, \alpha \rightarrow \beta) \rightarrow (ist(c_l, \alpha) \rightarrow ist(c_l, \beta))\rangle}\,II\ 1$$

$(\leftarrow)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{1}{\langle c_k, ist(c_l, \alpha) \rightarrow ist(c_l, \beta)\rangle} \qquad \dfrac{\overset{2}{\langle c_l, \alpha\rangle}}{\langle c_k, ist(c_l, \alpha)\rangle}\,SF}{\langle c_k, ist(c_l, \beta)\rangle}\,IE}{\langle c_l, \beta\rangle}\,SB}{\langle c_l, \alpha \rightarrow \beta\rangle}\,II\ 2}{\langle c_k, ist(c_l, \alpha \rightarrow \beta)\rangle}\,SF}{\langle c_k, (ist(c_l, \alpha) \rightarrow ist(c_l, \beta)) \rightarrow ist(c_l, \alpha \rightarrow \beta)\rangle}\,II\ 1$$

$\diamondsuit$

4.3 Коректност и пълнота

Както знаем, в безконтекстните системи имаме пълнота и коректност, т.е. синтактичната и семантичната изводимост са еквивалентни. Да видим дали това е така при многоконтекстните системи. Тъй като вече доказахме еквивалентността на изводите по правилата Δ_{FC} и Δ_{NK}, сега ще сравняваме само изводимостта по правилата Δ_{FC} със семантичната изводимост.

Теорема 4.4. Всяка формула, която е изводима от една многоконтекстна система MCS с правила $\Delta_i = \Delta_{FC}, i \in I$, е и семантично изводима от тази многоконтекстна система, т.е.

$$MCS \vdash_{FC} \varphi \Rightarrow MCS \models \varphi.$$

Доказателство. Нека $MCS \vdash_{FC} \varphi$. Тогава съществува крайна редица, която е доказателство на φ от MCS. Съгласно лема 4.2 съществува крайно дърво на извод на φ от краен брой предпоставки $\varphi_1, ..., \varphi_n$. Нека $J = \langle D, \rho \rangle$ е един произволен модел на предпоставките $\varphi_1, ..., \varphi_n$. Ще докажем, че J е модел и на φ. За целта е достатъчно да покажем, че за всяко от правилата SF и Δ_{FC}, ако J е модел на неговите предпоставки, то J е модел и на заключението му. Нека V е една произволна оценка в J. Тогава за всяка предпоставка φ_i имаме $V(\varphi_i) = true$. Проверяваме правилата:

$$MP : \frac{\langle c_k, \alpha \rangle \qquad \langle c_k, \alpha \to \beta \rangle}{\langle c_k, \beta \rangle}$$

Имаме $V(\langle c_k, \alpha \rangle) = true$ и $V(\langle c_k, \alpha \to \beta \rangle) = true$. Но $V(\langle c_k, \alpha \to \beta \rangle) = true \Leftrightarrow V(\langle c_k, \alpha \rangle) = false$ или $V(\langle c_k, \beta \rangle) = true$. Тъй като $V(\langle c_k, \alpha \rangle) = true$, остава $V(\langle c_k, \beta \rangle) = true$.

$$Gen : \frac{\langle c_k, \alpha \rangle}{\langle c_k, \forall x \alpha \rangle}$$

Да допуснем, че J е модел на α, но не е модел на $\forall x \alpha$. Тогава съществува оценка V в J, за която $V(\langle c_k, \forall x \alpha \rangle) = false$, следователно съществува $a \in D$, такова че $V_a^x(\langle c_k, \alpha \rangle) = false$, откъдето съществува оценка $W = V_a^x$, за която $W(\langle c_k, \alpha \rangle) = false$ - противоречие. Следователно, всеки модел на α е модел и на $\forall x \alpha$.

$$SF : \frac{\langle c_k, \alpha \rangle}{\langle c_l, ist(c_k, \alpha) \rangle}$$

По дефиниция $V(\langle c_l, ist(c_k, \alpha) \rangle) = V(\langle c_k, \alpha \rangle)$

$$PE : \quad \frac{\begin{array}{c}[\langle c_k, \alpha \rangle] \\ \langle c_k, \beta \rangle\end{array}}{\langle c_k, \alpha \to \beta \rangle}$$

Ако $V(\langle c_k, \alpha \rangle) = false$, тогава по дефиниция $V(\langle c_k, \alpha \to \beta\alpha \rangle) = true$. Ако $V(\langle c_k, \alpha \rangle) = true$, тогава е в сила индуктивното предположение, че $V(\langle c_k, \beta \rangle) = true$, откъдето по дефиниция. Следователно, независимо от стойността на допускането $\langle c_k, \alpha \rangle$, $V(\langle c_k, \alpha \to \beta\alpha \rangle) = true$

$$XI_i : \frac{\langle c_k, \top \rangle}{Ax_i}$$

В класическата логика вече е проверено, че всяка интерпретация е модел на всяка аксиома.

От фактите по-горе следва, че ако стойността на всички предпоставки при оценката V е истина, то стойноността на заключението φ при тази оценка е също истина. Но тази оценка я избрахме произволна. Следователно, J е модел на φ. Интерпретацията J обаче също беше произволно избран модел на предпоставките. Така получаваме, че всеки модел на предпоставките $\varphi_1, ..., \varphi_n$ е модел и на φ. Остава да отбележим, че всеки модел на MCS е модел на предпоставките $\varphi_1, ..., \varphi_n$, тъй като това са контекстни аксиоми. Дори и част от тях да принадлежат на противоречиви контексти, по дефиниция стойността им при произволна оценка в произволна интерпретация е винаги $true$. $\diamondsuit$

Следствие. (Коректност) Всяка формула, която е синтактично изводима от една многоконтекстна система MCS с контексти, чиито правила са подмножества на Δ_{FC} или на Δ_{NK}, е и семантично изводима от нея.

Доказателство. Доказателството е непосредствено следствие от теореми 4.1 и 4.4. $\diamondsuit$

За съжаление в многоконтекстните системи нямаме пълнота. Това лесно се вижда от следния пример:

$$\begin{aligned} MCS: \quad & A_k = \{\neg\varphi\} \\ & A_l = \phi \end{aligned}$$

Нека J е произволен модел на контекстните аксиоми на MCS, т.е. на $\neg\varphi$. Следователно за произволна оценка V в J имаме $V(\langle c_k, \neg\varphi \rangle) = true$. Оттук

$$\begin{aligned} V(\langle c_k, \neg\varphi \rangle) = true \quad & \Leftrightarrow V(\langle c_k, \varphi \rangle) = false \Leftrightarrow \\ & \Leftrightarrow V(\langle c_l, ist(c_k, \varphi) \rangle) = false \Leftrightarrow \\ & \Leftrightarrow V(\langle c_l, \neg ist(c_k, \varphi) \rangle) = true \end{aligned}$$

Това означава, че всеки модел на контекстните аксиоми на MCS е модел на $\langle c_l, \neg ist(c_k, \varphi) \rangle$, следователно $MCS \models \langle c_l, \neg ist(c_k, \varphi) \rangle$. Но в контекста

c_l не можем да изведем $\neg ist(c_k, \varphi)$, защото нямаме правило, по което да го изведем. Имаме правило само за $ist(c_k, \varphi)$ (строгото доказателство, че $\langle c_l, \neg ist(c_k, \varphi)\rangle$ не може да се изведе от такава многоконтекстна система бе направено в раздел 4.2). Налице е пример, в който имаме семантична, но не и синтактична изводимост.

Липсата на пълнота е доста сериозен недостатък, но в случая тя донякъде се компенсира с други две качества. Първо, благодарение на нея получаваме едно много важно свойство на многоконтекстните системи - локалността на противоречието. Нещо повече:

Теорема 4.5. В общия случай, ако в една многоконтекстна система MCS имаме пълнота на извода, тогава тази многоконтекстна система губи свойството локалност на противоречието.

Доказателство. В общия случай между контекстите на една многоконтекстна система има и противоречиви. Нека c_k е един противоречив контекст, в който се извеждат формулите α и $\neg\alpha$. Нека c_l е произволен друг контекст, J е произволен модел на MCS, а V е произволна оценка в J. Тогава:

$$
\begin{aligned}
V(\langle c_k, \alpha\rangle) = true \quad & \Rightarrow V(\langle c_l, ist(c_k, \alpha)\rangle) = true \\
V(\langle c_k, \neg\alpha\rangle) = true \quad & \Rightarrow V(\langle c_k, \alpha\rangle) = false \Rightarrow \\
& \Rightarrow V(\langle c_l, ist(c_k, \alpha)\rangle) = false \Rightarrow \\
& \Rightarrow V(\langle c_l, \neg ist(c_k, \alpha)\rangle) = true
\end{aligned}
$$

т.е. в контекста c_l семантично се извеждат както $ist(c_k, \alpha)$, така и $\neg ist(c_k, \alpha)$. Следователно c_l е противоречив. Тъй като c_l беше произволно избран следва, че противоречието в един контекст води до противоречие във всички контексти. Следователно, при такава многоконтекстна система нямаме локалност на противоречието. $\diamondsuit$

Тази теорема ни изправя пред перспективата да избираме само едно от две много желани качества: пълнота на извода и локалност на противоречието. Очевидно нито един от двата варианта не е достатъчно добър. Може би най-добрият начин за разрешаване на този проблем, е да се търси преформулировка, при която да се атакува директно самата дилема. Засега този въпрос остава открит. В дипломната работа сме се спрели на варианта с локалност на противоречието.

Липсата на пълнота се донякъде се компенсира и от следната

Теорема 4.6. (частична пълнота) Изводът в рамките на един контекст е пълен.

Доказателство. Това е следствие от теоремата за пълнота в класическата логика. Разглеждайки ist като обикновен предикатен символ, без да му предаваме специален смисъл (т.е. $ist \in \mathcal{PR}$), можем да се освободим от многоконтекстната нотация на контекстните аксиоми на избрания контекст, както направихме това при доказателството на лема 4.6. Нека за определеност избраният контекст да е c_k, а A_k са неговите аксиоми. Ще получим множество $\tilde{A}_k$ от формули на класическата логика.

Ако разгледаме дефинициите за интерпретация и стойност на формула в един контекст, без специалното значение на ist, ще видим, че те са същите като съответните дефиниции в класическата логика. Това означава, че съществува биекция между моделите на $\tilde{A}_k$ и моделите на контекста c_k. Всеки модел на многоконтекстната система е модел на c_k, а за всеки модел на c_k има съответен модел на $\tilde{A}_k$. Оттук следва, че за всяка формула $\langle c_k, \alpha \rangle$, която е семантично изводима в c_k, съответната формула α е семантично изводима от $\tilde{A}_k$.

От теорема 3.2 имаме, че за всяка семантично изводима формула от едно множество, съществува нейно синтактично доказателство от същото множество. В лема 4.6 показахме, че съществува биекция между синтаксиса, както и между правилата за извод на един контекст и тези на класическата логика. Това означава, че в крайна сметка, за всяка формула, която е семантично изводима в c_k, може да се намери нейно синтактично доказателство, т.е. имаме пълнота в рамките на контекста. С това теоремата е доказана.

Забележка. Изводът

$$\frac{\langle c_k, \alpha \rangle}{\langle c_k, ist(c_k, \alpha) \rangle}$$

не го считаме за извод в контекста c_k, защото е направен с правилото за смяна на контекста SF, а не с правило от Δ_k.

$\diamondsuit$

Глава 5

Многоконтекстни системи със съставни контексти

В глава 4 разгледахме свойствата на примитивните контексти. Те изграждат основата на многоконтекстните системи, но само с тях, тези системи биха били непълни. Звеното, което липсва, е средство за динамично сформиране на нови контексти на базата на примитивните. В тази глава разглеждаме едно разширение на описаните до момента многоконтекстни системи, включващо и *съставни* контексти.

5.1 Дефиниции

5.1.1 Синтаксис

Искаме да въведем средство за сформиране на нови контексти, които да имат същата структура, както примитивните - език, контекстни аксиоми и правила за извод. Както вече казахме, в дипломната работа разглеждаме контексти с един и същ език. Тъй като един контекст се определя най-вече от неговите контекстни аксиоми, а те преставляват множество от формули, логичен подход е да се въведат операции за сформиране на нови контексти на базата на тези множества. Ще въведем следните четири контекстни операции (тук само ще ги дефинираме, а по-подробно ще ги разгледаме в раздел 5.3):

Ако имаме два контекста $c_1 = \langle L, A_1, \Delta_1 \rangle$ и $c_2 = \langle L, A_2, \Delta_2 \rangle$ в многоконтекстната система MCS, то

- $c_{1*} = c_1^* = \langle L, \{\langle c_1, \alpha \rangle | \langle c_1, \alpha \rangle \in Lit(Mcth(MCS))\}, \Delta_1 \rangle$
- $c_{1\cup 2} = c_1 \cup c_2 = \langle L, A_1 \cup A_2, \Delta_1 \rangle$
- $c_{1\cap 2} = c_1 \cap c_2 = \langle L, A_1 \cap A_2, \Delta_1 \rangle$
- $c_{1\setminus 2} = c_1 \setminus c_2 = \langle L, A_1 \setminus A_2, \Delta_1 \rangle$,

където $1^*, 1 \cup 2, 1 \cap 2$ и $1 \setminus 2$ са индексите на новосформираните контексти.

Така дефинирани, операциите ни показват как се конструират съставните контексти. За да ги използваме в синтаксиса на многоконтекстните системи, трябва да изразим техния смисъл със синтактични средства. Формално това може да стане с добавянето към Δ на следните четири правила за извод със смяна на контекста:

$$CE : \frac{\langle c_k, \alpha \rangle \quad is\_literal(\alpha)}{\langle c_k^*, \alpha \rangle \in A_{k^*}}$$

$$CU : \frac{\langle c_k, \alpha \rangle \in A_k}{\langle c_k \cup c_l, \alpha \rangle \in A_{k \cup l}} \qquad CU : \frac{\langle c_l, \alpha \rangle \in A_k}{\langle c_k \cup c_l, \alpha \rangle \in A_{k \cup l}}$$

$$CI : \frac{\langle c_k, \alpha \rangle \in A_k \quad \langle c_l, \alpha \rangle \in A_l}{\langle c_k \cap c_l, \alpha \rangle \in A_{k \cap l}} \qquad CD : \frac{\langle c_k, \alpha \rangle \in A_k \quad \langle c_l, \alpha \rangle \notin A_l}{\langle c_k \backslash c_l, \alpha \rangle \in A_{k \backslash l}}$$

Буквалният смисъл на правилото CE е „Ако сме извели $\langle c_k, \alpha \rangle$ и α е литерал, тогава α е аксиома на контекста c_{k^*}„. Тук се налага естественото ограничение, изводът на $\langle c_k, \alpha \rangle$ да не зависи от никакви допускания, т.е. да е безусловен. Смисълът на останалите правила е: „Ако предпоставките на правилото са контекстни аксиоми, то заключението е контекстна аксиома на новосформирания контекст„.

За да включим съставните контексти пълноценно в многоконтекстните системи се налагат и някои изменения и допълнения на част от дефинициите в раздел 2.1:

Нека $\mathcal{A}$ е крайна азбука, несъдържаща символите $\langle, \rangle, (,), \cup, \cap, \backslash, *$ и запетая. Нека имаме едно множество от индекси I и $\mathcal{A}_i$ за $i \in I$ са крайни азбуки, несъдържащи символите $\langle, \rangle, (,), \neg, \vee, \wedge, \rightarrow, \forall, \exists, \perp, \top$ и запетая. Нека още множествата $\mathcal{VAR}_i$, $\mathcal{IND}_i$, $\mathcal{CONST}_i$, $\mathcal{F}_i$, $\mathcal{PR}_i$, $\mathcal{CC}$ и $\mathcal{CV}$ са дефинираните в началото на раздел 2.1.

Дефиниция 5.1. *Контекстен израз* ще наричаме
1. c_i за всяко $i \in I$.
2. всяко $v \in \mathcal{CV}$.
3. w^*, където w е контекстен израз.
4. $w_1 \cup w_2$, където w_1 и w_2 са контекстни изрази.
5. $w_1 \cap w_2$, където w_1 и w_2 са контекстни изрази.
6. $w_1 \backslash w_2$, където w_1 и w_2 са контекстни изрази.

С $\mathcal{CX}$ ще означаваме множеството от контекстни изрази, а с $\mathcal{CE}$ - множеството от тези контекстни изрази, в които не участват контекстни променливи. Дефинициите от раздел 2.1 остават същите, но като се вземе предвид новото множество $\mathcal{CX}$.

На практика смисълът на множеството $\mathcal{CX}$ е един и същ и в раздел 2.1 и тук. Това е множеството от всички контекстни изрази, но този термин придобива смисъл едва сега. В раздел 2.1 контекстните изрази са само контекстни константи или променливи, защото там няма въведени контекстни операции. Аналогично, множеството $\mathcal{CE}$ при примитивните контексти се състои само от контекстни константи.

5.1.2 Семантика

Като имаме предвид, че

$$
\begin{aligned}
c_{1^*} = c_1^* = \langle L, \{\langle c_1, \alpha\rangle | \langle c_1, \alpha\rangle \in Lit(Mcth(MCS))\}, \Delta_1\rangle \\
c_{1\cup 2} = c_1 \cup c_2 = \langle L, A_1 \cup A_2, \Delta_1\rangle \\
c_{1\cap 2} = c_1 \cap c_2 = \langle L, A_1 \cap A_2, \Delta_1\rangle \\
c_{1\backslash 2} = c_1 \backslash c_2 = \langle L, A_1 \backslash A_2, \Delta_1\rangle
\end{aligned}
$$

дефинициите за интерпретация, оценка и стойност на терм и на формула се обобщават с малки изменения:

Нека имаме едно множество $D : |D| > 1$, което ще наричаме *проблемна област* или *област на компетентност*, и една функция $\rho : \mathcal{MCL} \to D$, която ще наричаме *смислова функция*. Нека ρ е такава, че ако $e \in \mathcal{CE}$, то

- ако $a \in \mathcal{CONST}$, то $\rho(\langle e, a\rangle) \in D$
- ако $f \in \mathcal{F}$, $ar(f) = n$, то $\rho(\langle e, f\rangle) : D^n \to D$
- ако $P \in \mathcal{PR}$, $ar(P) = n$, то $\rho(\langle e, P\rangle) \subseteq D^n$
- ако $p \in \mathcal{VAR}$, то $\rho(\langle e, p\rangle) \in \{true, false\}$
- $\rho(\langle e, \perp\rangle) = false, \rho(\langle e, \top\rangle) = true$

Интерпретация на многоконтекстна системата система MCS отново ще наричаме наредената двойка $J = \langle D, \rho\rangle$.

Дефиниция 5.2. *Оценка* V в J ще наричаме всяка функция $V : \mathcal{CE} \times \mathcal{IND} \to D$ и $V : \mathcal{CX} \to \mathcal{CE}$.

Дефиниция 5.3. *Стойност на терм* t *при оценка* V дефинираме по следния начин:

1. ако $t = \langle e, a\rangle, a \in \mathcal{CONST}$, то $V(t) = \rho(\langle e, a\rangle)$
2. ако $t = \langle e, x\rangle, x \in \mathcal{IND}$, то $V(t) = V(\langle e, x\rangle)$
3. ако $t = \langle e, f(t_1, ..., t_n)\rangle, f \in \mathcal{F}$, $ar(f) = n$,
то $V(t) = \rho(\langle e, f\rangle)(V(\langle e, t_1\rangle), ..., V(\langle e, t_n\rangle))$
4. ако $t = \langle w, h\rangle, w \in \mathcal{CX}$, то $V(t) = V(\langle V(w), h\rangle)$

Дефиниция 5.4. *Стойност на формула* φ *при оценка* V също дефинираме индуктивно:

1. $V(\langle e, \perp\rangle) = false$ $, V(\langle e, \top\rangle) = true$
2. ако $\varphi = \langle e, p\rangle, p \in \mathcal{VAR}$, то $V(\varphi) = \rho(\langle e, p\rangle)$
3. ако $\varphi = \langle e, P(t_1, ..., t_n)\rangle, P \in \mathcal{PR}$, $ar(P) = n$,
то $V(\varphi) = true \Leftrightarrow (V(\langle e, t_1\rangle), ..., V(\langle e, t_n\rangle)) \in \rho(\langle e, P\rangle)$

4. ако $\varphi = \langle e, \neg\alpha\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle e, \alpha\rangle) = false$.

5. ако $\varphi = \langle e, \alpha \vee \beta\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle e, \alpha\rangle) = true$ или $V(\langle e, \beta\rangle) = true$

6. ако $\varphi = \langle e, \alpha \wedge \beta\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle e, \alpha\rangle) = true$ и $V(\langle e, \beta\rangle) = true$

7. ако $\varphi = \langle e, \alpha \rightarrow \beta\rangle$, то $V(\varphi) = true \Leftrightarrow V(\langle e, \alpha\rangle) = false$ или $V(\langle e, \beta\rangle) = true$

8. ако $\varphi = \langle e, (\forall x)\alpha\rangle, x \in \mathcal{IND}$, образуваме нова оценка $V_a^x(y) = a$, ако $y = x$ и $V_a^x(y) = V(y)$, ако $y \neq x$. Тогава $V(\varphi) = true \Leftrightarrow$ за всяко $a \in D$ $V_a^x(\langle e, \alpha\rangle) = true$

9. ако $\varphi = \langle e, (\exists x)\alpha\rangle, x \in \mathcal{IND}$, образуваме нова оценка $V_a^x(y) = a$, ако $y = x$ и $V_a^x(y) = V(y)$, ако $y \neq x$. Тогава $V(\varphi) = true \Leftrightarrow$ съществува $a \in D$ $V_a^x(\langle e, \alpha\rangle) = true$

10. ако $\varphi = \langle e, (\forall v)\alpha\rangle, v \in \mathcal{CV}$, образуваме нова оценка $V_c^v(y) = c$, ако $y = v$ и $V_c^v(y) = V(y)$, ако $y \neq v$. Тогава $V(\varphi) = true \Leftrightarrow$ за всяко $c \in \mathcal{CE}$ $V_c^v(\langle c, \alpha\rangle) = true$

11. ако $\varphi = \langle e, (\exists v)\alpha\rangle, v \in \mathcal{CV}$, образуваме нова оценка $V_c^v(y) = c$, ако $y = v$ и $V_c^v(y) = V(y)$, ако $y \neq v$. Тогава $V(\varphi) = true \Leftrightarrow$ съществува $c \in \mathcal{CE}$ $V_c^v(\langle c, \alpha\rangle) = true$

12. ако $\varphi = \langle e, ist(e_1, \alpha)\rangle$, то $V(\varphi) = V(\langle e_1, \alpha\rangle)$

13. ако $\varphi = \langle w, \alpha\rangle, w \in \mathcal{CX}$, то $V(\varphi) = V(\langle V(w), \alpha\rangle)$

Тук навсякъде $e \in \mathcal{CE}$ и $w \in \mathcal{CX}$.

Останалите дефиниции от раздел 2.3 остават същите.

5.2 Свойства

Тъй като направихме разширение на основните дефиниции, е редно отново да прегледаме основните резултати:

Теорема 5.1. (еквивалентност на FC и NK) Ако $c_i = \langle L, A_i, \Delta_i\rangle$ е един произволен контекст на системата $MCS = \langle\{c_i\}_{i\in I}, \Delta\rangle$, то многоконтекстните системи $MCS_{FC} = MCS(i|\langle L, A_i, \Delta_{FC}\rangle)$ и $MCS_{NK} = MCS(i|\langle L, A_i, \Delta_{NK}\rangle)$ са еквивалентни, т.е. $MCS_{FC} \simeq MCS_{NK}$.

Доказателство. От доказателството на теорема 4.1 имаме, че групите правила за извод $\Delta_{FC} \cup \Delta$ и $\Delta_{NK} \cup \Delta$ са равномощни. Сега към всяка от тези две групи добавяме едни и същи четири нови правила CE, CU, CI и CD. Следователно получените нови групи правила са също равномощни, т.е. FC-изводимостта е еквивалентна на NK-изводимостта и при съставните контексти. $\diamondsuit$

Съставните контексти се конструират от примитивните чрез модификация на контекстните аксиоми и присвояване на правила за извод. В импортната част на кой да е контекст, аксиомите нямат никакво значение, а лема 4.8 е доказана за произволно подмножество на Δ_{FC} и Δ_{NK}.

Следователно, и за съставните контексти са изпълнени тази лема и нейните две следствия: теоремата за локалност на противоречието и теорема 4.3, заедно със свойствата на ist-формулите.

Тук обаче има един факт, на който трябва да обърнем по-специално внимание. Ако c_k е един противоречив контекст, тогава автоматично и контекстът $c_{k \cup l}$ е противоречив. От друга страна теоремата за локалност на противоречието (теорема 4.2) твърди, че то не може да се внесе от един контекст в друг. Къде е недоразумението?

Недуразумение няма. Теоремата за локалност на противоречието продължава да е вярна, защото ако премахнем аксиомите на $c_{k \cup l}$, в него наистина няма да можем да изведем противоречие. В случая контекстът $c_{k \cup l}$ е противоречив, защото така е конструиран. Точно такъв е смисълът и на теорема 4.2 - „Ако в един контекст се изведе противоречие, „виновник„ за това е някоя негова аксиома„. В този пример, виновни са същите аксиоми, от които се извежда противоречие и в контекста c_k.

Теорема 5.2. (Коректност) Всяка формула, която е синтактично изводима от една многоконтекстна система MCS с контексти, чиито правила са подмножества на Δ_{FC} или на Δ_{NK}, е и семантично изводима от нея.

Доказателство. Доказателството на тази теорема е продължение на доказателството на теорема 4.4. Трябва да проверим верността на твърдението за новите 4 правила, при произволна интерпретация J и оценка V:

$$CE : \frac{\langle c_k, \alpha \rangle \quad is\_literal(\alpha)}{\langle c_{k^*}, \alpha \rangle \in A_{k^*}}$$

Тъй като формулата $\langle c_k, \alpha \rangle$ е изведена безусловно, тя принадлежи на $Mcth(MCS)$. В такъв случай, ако тя е литерал ($is\_literal(\alpha)$), ще принадлежи на множеството $\{\langle c_k, \alpha \rangle | \langle c_k, \alpha \rangle \in Lit(Mcth(MCS))\}$, което по дефиниция е A_{k^*}. Следователно, тя се явява контекстна аксиома на c_{k^*}. Тъй като J е модел на всички контекстни аксиоми, $V(\langle c_{k^*}, \alpha \rangle) = true$ за произволна оценка V в J.

Останалите правила отразяват множествените операции обединение, сечение и разлика. Стойността на техните заключения за произволна оценка V е винаги $true$, защото те са аксиоми на новосформираните контексти, а както вече споменахме, J е модел на всички контекстни аксиоми. $\diamondsuit$

Естествено, след като нямаме пълнота на извода при примитивните контексти, не можем да очакваме такава при съставните. Контрапримерът в раздел 4.3 е в сила за произволни контексти. Въпреки това, теореми 4.5 и 4.6 важат и за съставни контексти, защото в техните доказателства никъде не се използва факта, че контекстите са примитивни.

5.3 Операции с контексти

5.3.1 Затваряне

Дефиниция 5.5. Унарната операция

$$c_k^* = \langle L, \{\langle c_k, \alpha\rangle | \langle c_k, \alpha\rangle \in Lit(Mcth(MCS))\}, \Delta_k\rangle$$

над контекста $c_k = \langle L, A_k, \Delta_k\rangle$ ще наричаме *затваряне (капсулиране)*.

Затварянето би могло да се разглежда като вид обобщаване на контекст, защото е очевидно следното

Свойство 5.1 $c_k^* \subseteq c_k$.

Въпреки това тук го разглеждаме отделно, защото то е интересно от практическа гледна точка.

В някои случаи е полезно някои контексти да бъдат *затворени* относно възможната им композиция с други контексти. Прилагането на операцията затваряне върху един контекст прави този контекст достъпен за другите само в термините на неговите логически следствия. Резултатът от тази операция е множество от литерали - прости факти. По този начин се скриват както аксиомите на контекста, така и неговите общи познания за света.

От практическа гледна точка, тази операция моделира ситуацията, при която някой изисква само директни отговори на въпросите си, без да се интересува от начина, по който те се получават. В определени области, като военното и банковото дело, такова поведение е дори желателно.

Да разгледаме един пример. Да предположим, че един математик току-що е открил формулата за пресмятане на обема на прав кръгов цилиндър и не желае да я сподели с никого от страх, че някой ще си присвои заслугата за откриването ѝ. Затова хората, които използват съдове за вода с такава форма, отиват при него и го молят да им каже колко вода събира техният съд. Той прави съответните измервания, след това изчислява обема и им казва резултата. Тази ситуация може да се моделира по следния начин:

$$A_i = \{h(cyl, 25), r(cyl, 2), h(P, X) \wedge r(P, Y) \rightarrow vol(P, 3.14 * X * Y * Y)\}$$

$$\dfrac{\dfrac{\dfrac{\langle c_i, h(cyl,25)\rangle \quad \langle c_i, r(cyl,2)\rangle}{\langle c_i, h(cyl,25) \wedge r(cyl,2)\rangle}AI \qquad \langle c_i, h(P,X) \wedge r(P,Y) \rightarrow vol(P, 3.14 * X * Y * Y)\rangle}{\langle c_i, vol(cyl, 628)\rangle}IE}{\langle c_i^*, vol(cyl, 628)\rangle}CE$$

Тук контекстът c_i представлява познанията на математика, докато c_i^* е тази част от тях, която той казва на хората. Забележете, че формулата $h(P,X) \wedge r(P,Y) \to vol(P, 3.14 * X * Y * Y)$ принадлежи на контекста c_i, но не принадлежи на c_i^*, защото не е прост факт (литерал).

5.3.2 Специализиране

Специализирането е единият от двата основни вида операции с контекстите. Принципът, който се използва тук, е че колкото повече знаем за едно нещо, толкова по-конкретни (специализирани) са знанията ни за него. Като представител на този вид ще разгледаме операцията обединение.

Дефиниция 5.6. Бинарната операция $c_1 \cup c_2 = \langle L, A_1 \cup A_2, \Delta_1 \rangle$ над контекстите $c_1 = \langle L, A_1, \Delta_1 \rangle$ и $c_2 = \langle L, A_2, \Delta_2 \rangle$ ще наричаме *обединение на* c_1 *и* c_2.

Забележете, че операцията обединение не е комутативна. Тя запазва правилата за извод на първия от двата контекста.

Колкото повече знания имаме за дадена област толкова повече неща можем да изведем за нея. Ето пример за използването на обединението:

Двама приятели, да ги означим с $Mister$ 1 и $Mister$ 2, имат трети общ приятел $John$. $Mister$ 1 познава съпругата $Mary$ на $John$, но не познава сина му $Brian$, а $Mister$ 2 - точно обратното - знае, че $Brian$ е син на $John$, но не познава съпругата му. Да видим как при тези познания можем да конструираме контекст, в който да се изведе, че $Brian$ е син на $Mary$. За краткост ще означим $Mister$ 1 с c_1, $Mister$ 2 с c_2, а останалите с техните инициали.

$$
\begin{aligned}
A_1 = \quad & \{son(b,j), son(X,Y) \wedge spouse(Z,Y) \to son(X,Z), \\
& \ wife(X,Y) \to spouse(X,Y)\} \\
A_2 = \quad & \{wife(m,j)\}
\end{aligned}
$$

$$
\dfrac{
 \dfrac{
   \dfrac{\langle c_1, son(b,j)\rangle}{\langle c_1\cup c_2, son(b,j)\rangle}CU
   \quad
   \dfrac{\dfrac{\langle c_2, wife(m,j)\rangle}{\langle c_1\cup c_2, wife(m,j)\rangle}CU \quad \dfrac{\langle c_1, wife(X,Y)\to spouse(X,Y)\rangle}{\langle c_1\cup c_2, wife(X,Y)\to spouse(X,Y)\rangle}CU}{\langle c_1\cup c_2, spouse(m,j)\rangle}IE
 }{\langle c_1\cup c_2, son(b,j)\wedge spouse(m,j)\rangle}AI
 \quad
 \dfrac{\langle c_1, son(X,Y)\wedge spouse(Z,Y)\to son(X,Z)\rangle}{\langle c_1\cup c_2, son(X,Y)\wedge spouse(Z,Y)\to son(X,Z)\rangle}CU
}{\langle c_1\cup c_2, son(b,m)\rangle}IE
$$

Дефиниция 5.7. Бинарната операция $c_1 \cup c_2^* = \langle L, A_1 \cup A_2^*, \Delta_1\rangle$ над контекстите $c_1 = \langle L, A_1, \Delta_1\rangle$ и $c_2 = \langle L, A_2, \Delta_2\rangle$ ще наричаме *абсорбция (поглъщане) на c_2 от c_1.*

Този частен случай на специализирането е комбинация от обединение и затваряне. Той е характерен с това, че запазва принципа на локалността на извода. Към аксиомите на c_1 се добавят факти, които обаче са били изведени в контекста c_2 с правилата Δ_2.

В случаите, когато Δ_1 и Δ_2 са равномощни е валидно следното свойство, което е пряко следствие от свойство 5.1:

Свойство 5.2 $c_1 \cup c_2^* \subseteq c_1 \cup c_2$.

Когато Δ_1 и Δ_2 не са равномощни обаче, това горното свойство не е в сила. Това личи от следния контрапример:

$$\begin{aligned} A_1 = \quad & \{\psi \to \chi\} \\ A_2 = \quad & \{\varphi_1 \vee \varphi_2 \\ & \varphi_1 \to \psi \\ & \varphi_2 \to \psi\} \end{aligned}$$

Да допуснем, че правилата на контекста c_2 са тези на класическото предикатно смятане, а контекстът c_2 може да разсъждава само с конюнкции и с правилото MP, т.е. $\Delta_2 = \Delta_{FC}$, а $\Delta_1 = \{MP, AI, AE\}$. Тогава само с помощта на Δ_1 от $A_1 \cup A_2$ не може да се изведе ψ, защото в това множество няма правила обработващи логическата операция *или*. Това означава, че в контекста $c_1 \cup c_2$ не може да се изведе χ.

От друга страна формулата ψ се извежда в контекста c_2 и при това тя е атомарна. Оттук следва, че тя се извежда в контекста c_2^*, откъдето се извежда и в $c_1 \cup c_2^*$. В същия контекст се извежда и $\psi \to \chi$. Тук вече с помощта на правилата Δ_1 може да се изведе χ.

5.3.3 Обобщаване

Другият основен вид операции с контекстите е обобщаването. Неговият принцип е обратен на специализирането - колкото по-малко знаем за едно нещо, толкова по-общо ни изглежда то. Като представител на този вид ще разгледаме операцията сечение.

Дефиниция 5.7. Бинарната операция $c_1 \cap c_2 = \langle L, A_1 \cap A_2, \Delta_1\rangle$ над контекстите $c_1 = \langle L, A_1, \Delta_1\rangle$ и $c_2 = \langle L, A_2, \Delta_2\rangle$ ще наричаме *сечение на c_1 и c_2.*

Също като предишните бинарни операции и тази не е комутативна.

Да разгледаме един пример от света на кубчетата, илюстриращ смисъла на сечението на контексти. Нека имаме четири кубчета a, b, c и d, които можем да разполагаме едно върху друго в произволен ред. Тяхната позиция се описва с предиката $on(X, Y)$, който означава, че кубчето X е поставено върху кубчето Y.

Дадено ни е описанието на две ситуации c_1 и c_2. В първата кубчетата са наредени отдолу-нагоре в последователност a, b, c, d, а във втората - d, a, b, c. Разполагаме и със знания за термина $abv(X, Y)$, който означава, че кубчето X се намира някъде над (може и не непосредствено върху) кубчето Y. Общото в двете ситуации е, че c лежи на b, а b лежи на a. Да видим как в сечението на двата контекста c_1 и c_2, обединено със знанията за термина $abv(X, Y)$, можем да изведем, че c е над a.

$$
\begin{aligned}
A_1 = \quad & \{on(b,a), on(c,b), on(d,c)\} \\
A_2 = \quad & \{on(a,d), on(b,a), on(c,b)\} \\
A_{rel} = \quad & \{on(X,Y) \rightarrow abv(X,Y), \\
& on(X,Y) \wedge abv(Y,Z) \rightarrow abv(X,Z)\}
\end{aligned}
$$

$$\langle c_1, on(b,a)\rangle \langle c_2, on(b,a)\rangle$$
$$\text{————————————}CI$$
$$\langle c_1, on(c,b)\rangle \langle c_2, on(c,b)\rangle \quad \langle c_1 \cap c_2, on(b,a)\rangle \qquad \langle rel, on(X,Y) \rightarrow abv(X,Y)\rangle$$
$$\text{—————————}CI\text{——————————}CU\text{———————————————}CU$$
$$\langle c_1 \cap c_2, on(c,b)\rangle \quad \langle (c_1 \cap c_2) \cup rel, on(b,a)\rangle \quad \langle (c_1 \cap c_2) \cup rel, on(X,Y) \rightarrow abv(X,Y)\rangle$$
$$\text{—————————}CU\text{————————————————}IE$$
$$\langle (c_1 \cap c_2) \cup rel, on(c,b)\rangle \quad \langle (c_1 \cap c_2) \cup rel, abv(b,a)\rangle \quad \langle rel, on(X,Y) \wedge abv(Y,Z) \rightarrow abv(X,Z)\rangle$$
$$\text{————————————————————}AI\text{————————————————}CU$$
$$\langle (c_1 \cap c_2) \cup rel, on(c,b) \wedge abv(b,a)\rangle \qquad \langle (c_1 \cap c_2) \cup rel, on(X,Y) \wedge abv(Y,Z) \rightarrow abv(X,Z)\rangle$$
$$\text{——————————————————————————————————————}IE$$
$$\langle (c_1 \cap c_2) \cup rel, abv(c,a)\rangle$$

Последната контекстна операция, която ще разгледаме, е контекстанта разлика:

Дефиниция 5.8. Бинарната операция $c_1 \backslash c_2 = \langle L, A_1 \backslash A_2, \Delta_1\rangle$ над контекстите $c_1 = \langle L, A_1, \Delta_1\rangle$ и $c_2 = \langle L, A_2, \Delta_2\rangle$ ще наричаме *разлика на c_1 и c_2*.

С нейна помощ можем да установяваме различията в два контекста. Ето как става това с описаните ситуации в примера по-горе:

$$\langle c_1, on(d,c)\rangle \qquad\qquad \langle c_2, on(a,d)\rangle$$
$$\text{—————————}CD \qquad \text{—————————}CD$$
$$\langle c_1 \backslash c_2, on(d,c)\rangle \qquad\qquad \langle c_2 \backslash c_1, on(a,d)\rangle$$
$$\text{——————————}CU \qquad \text{——————————}CU$$
$$\langle c_1 \backslash c_2 \cup c_2 \backslash c_1, on(d,c)\rangle \qquad \langle c_1 \backslash c_2 \cup c_2 \backslash c_1, on(a,d)\rangle$$
$$\text{——————————————————————————}AI$$
$$\langle c_1 \backslash c_2 \cup c_2 \backslash c_1, on(a,d) \wedge on(d,c)\rangle$$

5.4 Йерархия на контекстите

С помощта на обобщенията и специализациите се образува естествена йерархия на контекстите. На върха на тази йерархия се намира празният контекст, защото той е най-общ (не може да се обобщи). По-надолу следва затварянето на сечението на всички контексти, а след него самото сечение на всички контексти. По-нататък се редуват затварянията на сеченията на всички $n-1$-орки, $n-2$-орки и т.н. двойки контексти, и самите сечения.

След тях идват затварянията на примитивните контексти и самите примитивни контексти. Оттук дървото на йерархия се разклонява още по-силно, защото следват обобщенията: първо по двойки, след това по тройки и т.н. Между тях се намират и получените чрез абсорбция съставни контексти.

Разбира се, това е само приблизителна структура на йерархичното дърво. В общия случай, точната е трудно да се определи, тъй като тя зависи както от броя на контекстите и техните аксиоми, така и от формулите, които могат да се изведат от тях. Във всички случаи обаче е налице композиция на огромен брой съставни контексти от сравнително малък брой примитивни.

Глава 6

Примерна реализация на езика PROLOG

6.1 Описание на програмата MCSI.PL

Най-напред ще отбележим, че програмата MCSI.PL, приложена в края на дипломната работа, не следва точно въведения синтаксис на многоконтекстните системи. Тя представлява един мета-интерпретатор на езика PROLOG и, следователно, формулите, с които работи са клаузи на Хорн, а изводът се извършва по метода на резолюцията с участие на предположението за затворения свят - ако нещо не може да се изведе от базата знания, тогава то не е вярно. Това определя и разликите с теоретичния модел, въведен в тази работа.

Една от основните разлики е смисълът на ist-формулите. Операционната семантика на една формула $ist(context, goal)$, която в програмата се записва `goal in context`, се състои в опит за удовлетворяване на целта $goal$ в контекста $context$. Ако този опит успее, успява и целта $goal\ in\ context$, иначе - целта не може да се удовлетвори. В този случай би се удовлетворила целта $not(goal)\ in\ context$. На езика на многоконтекстните системи това означава, че имаме синтактична изводимост на $\neg ist(c_k, \alpha)$, когато в контекста c_k не се знае нищо за α. Точно тук се крие причината за разликите със синтактичната изводимост в многоконтекстните системи.

Написването на програма, която да следва ефективно синтаксиса на многоконтекстните системи е по същество достатъчно трудна задача и е извън целите на настоящия труд. Програмата, представена тук, има за цел единствено да демонстрира някои от идеите за работа с много контексти.

Както вече споменахме, програмата MCSI.PL представлява един мета-интерпретатор на езика PROLOG. Целите се задават заедно с името на контекста, в който ще се удовлетворяват. След това предикатът $solve$

се опитва да ги удовлетвори. Този предикат представлява съществената част от програмата.

Предикатът $solve\ (Goal, CurrentContext, OriginalContext, R)$ прави опит да удовлетвори целта $Goal$ в контекста $CurrentContext$. В аргумента $OriginalContext$ се съдържа информация за първоначалния контекст, от който е започнало решаването на даден проблем, а аргументът R предава служебна информация, с която се реализира отрязването на дървото на извода при срещане на предиката ! (cut). Когато се стигне до !, първият път R приема стойност $true$ и се прави опит да се удовлетвори целта. Ако този опит успее - това е отговорът. В противен случай R приема стойност cut, която се връща нагоре по дървото на извода и указва на останалите предикати да не търсят алтернативни варианти.

Преди да се пристъпи към оценяването на общия случай, се разглеждат някои частни случаи. Две такива цели са $current\_context(X)$, която свързва X с името на текущия контекст, и $original\_context(X)$, която свързва аргумента X с името на контекста, от който е започнала дадена фаза на решаване на проблема (обикновено, контекста, в който е зададена първоначалната цел).

Следват случаите със смяна на контекста $G\ in\ C$ и $G\ in\_o\ C$. Случаят $G\ in\ C$ представлява смяна на текущия и на първоначалния контекст с контекста C. Той се използва в случаи, в които искаме да отбележим нова фаза в решаването на даден проблем и да запазим името на контекста, от който е започнала тази фаза. Случаят $G\ in\_o\ C$ се използва за смяна само на текущия контекст, без да променяме първоначалния.

По-нататък се оценяват $call$, not и останалите системни предикати. Без тях възможностите на програмите биха били силно ограничени. Следва разглеждане на случаите на конюнкция и дизюнкция на цели, при които първо се оценява първата цел и при необходимост се оценява и втората. Накрая, след като се е оказало, че текущата цел не е изключение, тя се разглежда в общия вид $solve(G, C, OC, R)$, като базата знания се претърсва за клаузи с подходящи глави и, ако се намерят такива, се оценяват техните тела.

Претърсването на базата знания се осъществява с помощта на предикатите $atomic\_clause$ и $compoud\_clause$. Най-напред се разглежда контекста на целта и, ако той се окаже съставен, се раздробява на примитивни с помощта на $compound\_clause$. Когато се стигне до примитивен контекст, $atomic\_clause$ намира съответната клауза в базата знания. Тук под примитивен контекст се има предвид контекст, който си има име, дори с това име да е асоцииран контекстен израз.

Клаузите на различните контексти се съхраняват в базата знания под формата на клаузи от вида $ctx(C, H) : -B$. Тук C е името на контекста, на който принадлежи клаузата, H е нейната глава, а B е тялото ѝ. Когато въпросната клауза е факт, тя се записва под формата $ctx(C, H)$, което е еквивалентно на $ctx(C, H) : -true$. Освен този вид клаузи се използват и клаузи от вида $include(CG, CS)$. Те указват, че контекстът CG е обобщение на контекста CS, т.е. съдържа всички негови клаузи. С този предикат може да се въведе йерархия на контекстите.

Предикатът $loadfile$ зарежда многоконтекстната програмата и я привежда във вида описан по-горе. Той позволява на потребителя да напише програмата си на обикновен текстов редактор с по-удобен за човека синтаксис, след което да я запише на файл и да я използва с мета-интерпретатора. При желание, базата знание може да се изчисти с предиката $reset$.

И накрая, с цел улеснение на синтаксиса и премахването на излишни скоби, в началото на програмата са дефинирани различни оператори.

6.2 Работа с програмата и примери

Преди всичко, програмата MCSI.PL трябва да се зареди и консултира с базата знания:

```
?- reconsult('MCSI.pl').
```

След това може да се зареди и консултира самата потребителска програма. Преди всяко ново зареждане е добре да се изчисти базата знания. Това става с предиката

```
?- reset.
```

Самото зареждане на потребителската програма става с предиката

```
?- loadfile('prog.pl').
```

Едновременно с прочитането на файла, мета-интерпретаторът преобразува потребителската програма от нейния синтаксис в своето вътрешно представяне. В процеса на преобразуване, той извежда на екрана текущия контекст и текущата аксиома, която обработва. Накрая, ако няма синтактични грешки, се извежда yes.

Синтаксисът е изключително прост. Описанието на всеки контекст започва с ключовата дума $context(name)$, където $name$ е името на контекста. Следва секцията за обявяване наследяването на аксиомите на други контексти. Това става с предикати от вида $include(name1)$. Техният смисъл е, че всички аксиоми на контекста $name1$ стават аксиоми и на $name$. Самите аксиоми се намират между ключовите думи $begin$ и end. Накрая, програмата трябва да завърши с $endfile$.

Забележка. След всяка формула и всяка ключова дума се поставя точка.

Ето един пример:

```
;;; file exam1.pl:  Example 1
;;;-----------------------------
context(pet).
    include(cat).
    include(dog).
    include(mouse).
begin.
    food(canned_food).
end.

context(cat).
begin.
    food(mouse).
end.

context(dog).
begin.
    food(meat).
end.

endfile.
```

Изходът при компилирането му изглежда така:

```
?- loadfile(' /DW/exam.pl').
CONTEXT:pet
include(cat)
include(dog)
include(mouse)
food(canned_food)

CONTEXT:cat
food(mouse)

CONTEXT:dog
food(meat)

CONTEXT:mouse
food(cheese)

yes
```

Сега вече можем да задаваме въпроси:

```
?- food(cheese) in mouse.

yes
```

```
?- food(X) in cat.
X = mouse ?  ;
no

?- food(X) in pet.
X = canned_food ?  ;
X = mouse ?  ;
X = meat ?  ;
X = cheese ?  ;
no
```

Да разгледаме един пример с композиция на контексти. Ще използваме операцията „*„, която може да се нарече доминиращо обединение. В случая $c_1 * c_2$ означава, че към клаузите на c_1 се добавят тези на c_2, които не се отнасят до термини от c_1 (главите им не са глави на клаузи на c_1).

```
;;; file exam2.pl:  Example 2
;;;----------------------------
context(world).
begin.
    likes(X,Y):-man(X),woman(Y).
    woman(X):-unmarried_girl(X).
    man(peter).
    man(brian).
    woman(maria).
    unmarried_girl(annie).
end.

context(group1).
begin.
    likes(X,Y):-man(X),unmarried_girl(Y).
end.

context(group).
    include(group1*world).
begin.
end.

endfile.
```

В аксиомите на контекста *world* са изразени фактите, че обикновено мъжете харесват жените и че неомъжените момичета са жени, както и са цитирани имената на някои мъже, жени и неомъжени момичета. Естествено, в този контекст имаме:

```
?- likes(X,Y) in world.
X = peter
Y = maria ?  ;

X = peter
Y = annie ?  ;

X = brian
Y = maria ?  ;

X = brian
Y = annie ?  ;

no
```

В клаузата в контекста $group1$ сме заявили, че една група мъже харесват само неомъжените млади момичета. Затова в контекста $group = group1 * world$ имаме само:

```
?- likes(X,Y) in world.
X = peter
Y = annie ?  ;

X = brian
Y = annie ?  ;

no
```

Следващият пример ни показва как можем да скрием вътрешната реализация на някоио предикати. В него се използва операцията, означена тук с „and„, която представлява сечение на изводимите факти от два контекста.

```
;;; file exam3.pl:  Example 3
;;;------------------------------
context(list).
begin.
   rev(X,Y):-rev(X,[],Y).
   rev([E|X],Acc,Y):-rev(X,[E|Acc],Y).
   rev([],Y,Y).
end.

context(exprt).
begin.
   H:-functor(H,F,N),N<3.
end.
```

```
context(lst).
    include(list and exprt).
begin.
end.

endfile.


?- rev([a, b, c], X) in What.
X = [c, b, a]
What = list ?  ;

X = _1
What = exprt ?  ;

X = [c, b, a]
What = lst ?  ;

no

?- rev([a, b, c], [d], X) in list.
X = [c, b, a, d] ?  ;
no

?- rev([a, b, c], [d], X) in lst.
no
```

Накрая ще демонстрираме един пример, от който става ясно, че в някои случаи само информацията за текущия контекст не е достатъчна. Искаме в контекста *animal* да изразим твърденията, че ако едно животно има 2 крила, то може да лети, и че ако животното има 2 крака, тогава то ходи по земята. Информацията за това, че човекът има два крака е логично да се намира в контекста *human*, а фактът че птиците имат 2 крила - в контекста *bird*. За да получим достъп до тези факти се налага използването на предиката *original_context*, който да ни покаже първоначалния контекст на задачата:

```
;;; file exam4.pl:  Example 4
;;;-------------------------------
context(animal).
begin.
    mode(fly):-original_context(X),no_of_wings(2) in X.
    mode(run):-original_context(X),no_of_legs(2) in X.
end.
```

```
context(bird).
begin.
    no_of_wings(2).
end.

context(human).
begin.
    no_of_legs(2).
end.

endfile.


?- mode(X) in bird+animal.
X = fly ?  ;
no

?- mode(X) in human+animal.
X = run ?  ;
no
```

Глава 7

Обобщение и заключение

В тази дипломна работа разгледахме един логически апарат за формализиране на някои аспекти на човешкото мислене: локалност и схематичност на разсъжденията, локалност на противоречието, динамично конструиране на описания на нови ситуации.

Локалността и схематичността на разсъжденията се моделират с въвеждането на термина „контекст„. На контекста в тази работа гледахме като на теория, породена от неговите аксиоми и затворена относно правилата му за извод. Аксиомите на един контекст представляват описание на един аспект от познанията ни за света, а правилата са методите ни на разсъждение, свързания с тези познания. По този начин в контекста са отразени не само познанията ни за света, но и методи за тяхното получаване (механизъм за извод).

Схематичността на разсъжденията се гарантира и от ограничения набор правила за смяна на контекста - единствено общо правило за смяна на контекста и четири конструктивни за съставните контексти. Докато контекстните правила за извод определят възможностите на даден контекст, правилата за смяна на контекста придават облика на многоконтекстната система, като структура от независими контексти.

Окончателната стъпка към осигуряване на независимостта на контекстите е направена с доказателството на теоремата за локалност на противоречието. Тази теорема ни гарантира, че наличието на противоречиви контексти в една многоконтекстна система не оказва влияние на другите. Така се моделира толерантността към грешки - едно твърде ценно човешко качество.

В дипломната работа разгледахме и начини за сформиране на описания на нови ситуации въз основа на няколко стари. Това става с помощта на контекстните операции, с чиято помощ се конструират съставните контексти. С представения набор такива операции - затваряне и класовете обобщаване и специализиране - въз основа на сравнително малък брой примитивни контексти, могат да се конструират огромен брой съставни.

С тяхна помощ се образува и естествена йерархия на контекстите в една многоконтекстна система.

Изложеното в дипломната работа не може и не претендира да бъде завършен научен труд. По-скоро, тя полага основите, очертава някои проблеми и дава идеи за насоки на по-нататъшни изследвания.

Най-важният проблем е липсата на пълнота на извода и по-специално, дилемата пълнота/локалност на противоречието. Нито едно от нейните решения не е достатъчно добро, защото искаме да разполагаме и с двете качества. Това означава, че трябва да се атакува самата дилема. Би могло да се намери преформулировка на нещата, при която дилемата да се избегне.

Друг проблем, споменат в дипломната работа, е свързан с езиците на контекстите. Добре би било да се намери средство за приобщаването към една многоконтекстна система на контексти с различни езици. Това не би увеличило изразителната сила на системите, но би ги направило по-естествени и по-близки до реалността.

От практическа гледна точка възниква проблем с ефективното реализиране на механизма за синтактичен извод на многоконтекстите системи. Методът на резолюцията, така както е използван в езика *Prolog*, не е подходящ, защото при него операционната семантика на предиката *ist* се различава от желаната. Освен това, в този метод не са предвидени средства за разсъждения с допускания.

Една интересна идея за бъдещи разширения е включването на апарата на тризначната логика. На мисълта за това ни навежда вътрешната тризначност на предиката $ist(c, \varphi)$. Много по-естествено е да приемем, че той има стойност $true$, ако φ е вярно в c, $false$ - ако в c е вярно $\neg\varphi$, и $1/2$ - ако стойността на φ е неизвестна в c. Така бихме могли да се преборим с противоречията. Ако в даден контекст c се извеждат φ и $\neg\varphi$, бихме могли да го разделим на два подконтекста c_1 и c_2, в единия от които се извежда φ, а в другия - $\neg\varphi$. Тогава в c - тяхното обединение - вместо φ и $\neg\varphi$, можем да положим, че се извежда $V(\varphi) = 1/2$.

В заключение, едно очевидно пряко приложение на представения в дипломната работа модел е в изграждането на декларативни бази и банки от данни. В последните години се правят опити за изграждане на много големи бази данни (Very Large Data Bases), описващи човешките познания в различни области. Прилагането на този модел се улеснява от факта, че неговите структурни единици, контекстите, са относително независими, което дава възможност за лесно разпаралеляване и използване в разпределени бази от данни. Актуалността на последните се гарантира и от бума в развитиетно на глобалните компютърни мрежи, като Internet.

Апендикс

Листинг на програмата *MCSI.PL*

```
;;; Operators
;;;-------------------------------------------------------------------
:- op(101,yfx,in).   % Context switching, incl.  original_context
:- op(101,yfx,ino).  % Context switching; saves original_context
:- op(51,xfy, and).  % Logical and

:- op(52,xfy, or).   % Logical or
:- op(50,xfy, but).  % Logical difference
:- op(31,xfy,+).   % sum

:- op(31,xfy,*).   % union
:- op(30,yfx,\).   % subtraction
:- op(21,xfy,&).   % intersection (only heads)
:- op(21,xfy,&&).  % intersection (full clauses)

;;; Clear the knowledge base and consult a file
;;;-------------------------------------------------------------------

reset :- !, retractall(ctx(_,_)), retractall(include(_,_)).

loadfile(X) :- atomic(X), see(X), seen, see(X), comp, seen.

comp :- read(S), check(S).
check(endfile) :- !.
check(context(C)) :- write('CONTEXT:'), write(C), nl,
            do_include(C), comp(C), comp.

do_include(C) :- read(S), check_include(C,S).
check_include(C,begin) :- !.
check_include(C,include(S)) :- write(include(S)), nl,
            assertz(include(C,S)), do_include(C).
```

```
comp(C) :- read(S), check(C,S).
check(C,end) :- !, nl.
check(C,(H :- B)) :- !, write((H :- B)), nl,
            assertz((ctx(C,H) :- B)), comp(C).
check(C, H) :- !, write(H), nl,
            assertz(ctx(C,H)), comp(C).

;;; The main part of the solver.
;;; Solves a problem in given context, or
;;; finds a primitive context in which given logical formula is true.
;;;-------------------------------------------------------------------

conunct((A, B)).
disunct((A; B)).

system(A) :- A t̄rue, not(disunct(A)), not(conunct(A)),
            not((clause(A,B);clause(A, true))),
            functor (A, B, C), B n̄ot, prolog_system_predicate(B,C).

Goal in Context :- solve(Goal,Context,Context,R), R = true.
Goal in_o Context :- solve(Goal,Context,Context,R), R = true.

solve(true,_,_,true) :- !.
solve(!,_,_,R) :- !, (R is true; R is cut).

solve(current_context(X),X,_,true) :- !.
solve(original_context(X),_,X,true) :- !.

solve(G in C,_,OC,true) :- !, solve(G,C,C,R), R = true.
solve(G in_o C,_,OC,true) :- !, solve(G,C,OC,R), R = true.
solve(call(A),CC,OC,true) :- !, solve(A,C,OC,R), R = true.
solve(not(G),C,OC,_) :- solve(G,C,OC,R), R = true, !, fail.
solve(not(_),C,_,true) :- !.
solve(G,_,_,true) :- system(G),!,call(G).
solve((A,B),C,OC,R) :- !, solve(A,C,OC,R1), (R1 = true, solve(B,C,OC,R2),
            (R2 = true, R is true;
            R2 = cut, !, R is cut);
            R1 = cut, !, R is cut).
solve((A;B),C,OC,R) :- !, (solve(A,C,OC,R1), (R1 = true, R is true;
            R1 = cut, !, R is cut);
      solve(B,C,OC,R1), (R1 = true, R is true;
            R1 = cut, !, R is cut)).
solve(G,C,OC,R) :- compound_clause(C,G,Body),solve(Body,C,OC,R1),
            (R1 = true, R is true;
```

```
          R1 = cut, !, fail).

;;; Searches for clauses that belong to a given primitive context
;;;-----------------------------------------------------------------
atomic_clause(Ctx,H,B):-clause(ctx(Ctx,H),B).
atomic_clause(Ctx,H,B):-include(Ctx,CtxN),compound_clause(CtxN,H,B).

;;; Deals with the clauses of context expressions.
;;;----------------------------------------------------------------
compound_clause(Ctx,H,B):-var(Ctx),!,atomic_clause(Ctx,H,B).
compound_clause(Ctx,H,B):-atomic(Ctx),!,atomic_clause(Ctx,H,B).

compound_clause(enc(C),H,(H in_o C)).
compound_clause(C1+C2,H,B) :- !,(compound_clause(C1,H,B);
          compound_clause(C2,H,B)).
compound_clause(C1&C2,H,B) :- !, compound_clause(C1,H,B),
          compound_clause(C2,H,_).
compound_clause(C1&&C2,H,B) :- !, compound_clause(C1,H,B),
          compound_clause(C2,H,B).
compound_clause(C1\ C2,H,B) :- !, compound_clause(C1,H,B),
          not(compound_clause(C2,H,_)).
compound_clause(C1*C2,H,B) :- !, compound_clause(C1+C2 \ C1,H,B).
compound_clause(C1 and C2,H,(H in_o C1, H in_o C2)) :- !.
compound_clause(C1 or C2,H,(H in_o C1; H in_o C2)) :- !.
compound_clause(C1 but C2,H,(H in_o C1, not(H) in_o C2)) :- !.
```

Библиография

[1] Dichev Ch., *'Context formation in logic programming setting'*, Proceedings of TAINN'95, Turkey, June 1995, pp. 207-218

[2] Dichev Ch., *'A framework for a contextual reasoning'*, 1995

[3] Gentzen G., *'Untersuchungen uber das logishe Schliessen'*, Matematische Zeitschrift, vol. 39, 1935, pp. 176-210, 405-443

[4] Giunchiglia F., *'Contextual reasoning'*, Proceedings of the IJCAI'93 Workshop on "Using knowledge in context", pp. 39-50

[5] Giunchiglia F., Serafini L., *'Multilanguage hierarchical logics, or: how we can do without modal logics'*, Artificial Intelligence, vol 65, 1994, pp. 29-70

[6] Guha R. V., *'Contexts: A formalization and some applications'*, Stanford Ph.D thesis, 1991

[7] McCarthy J., *'Notes on formalizing context'*, Proceedings of Thirteenth International Joint Conference on Artificial Intelligence (IJCAI'93), France, 1993, pp. 550-560

[8] Prawitz D., *'Ideas and results in proof theory'*,

[9] Тинчев Т., Вакарелов Д., *Лекции по математическа логика*, ФМИ, СУ „Св. Климент Охридски„, 1992-93