

ТЕМА 13

Семантична характеристика на логическите формули и програми

ВУЗ: Софийски Университет

Факултет: ФМИ

Специалност: Информатика

Конспект: 2004 г.

Автор: Иван Стамболийски

Използвана литература:

1. Записки по Логическо програмиране от Димитър Скордев

Съдържание:

13.1. Синтаксис и семантика на логическите формули. Изпълнимост и тъждествена вярност.....2

13.1. Синтаксис и семантика на логическите формули. Изпълнимост и тъждествена вярност

Дадена е една азбука, наречена *базисна*, която съдържа измежду своите знаци двете кръгли скоби и запетаята. Ще предполагаме също, че е фиксирано някое такова множество W от думи над базисната азбука, че префиксните изрази над W да имат еднозначен анализ. Нека W е обединение на множество X (*множество на променливите*), за което ще искаме да е безкрайно, множества $F_0, F_1, F_2, \dots$ (*множества съответно на нулместните, едноместните, двуместните и пр. функционални символи*), множества $P_0, P_1, P_2, \dots$ (*множества съответно на нулместните, едноместните, двуместните и пр. предикатни символи*) и непразно множество L (*множество на логическите символи*), като при това множествата X и L нямат общи елементи с никое от множествата F_n и P_n , а също и помежду си (други пресичания на множества измежду споменатите се допускат). Нулместните функционални символи ще наричаме още *константи*. Двойката от редиците $F_0, F_1, F_2, \dots$ и $P_0, P_1, P_2, \dots$ ще наричаме *сигнатура*.

Пример

Горното положение на нещата би могло да се осъществи например по следния начин, подходящ за разглеждане на чисто логическата част на езика Пролог в реализацията му Strawberry Prolog:

- а) базисната азбука съдържа поне главните и малките латински букви, цифрите, знака за подчертаване $_$, двете кръгли скоби и запетаята;
- б) множеството X се състои от думите, които започват с главна латинска буква или със знак за подчертаване и не съдържат други знаци освен латински букви, знак за подчертаване и цифри, като изключваме обаче еднобуквената дума, състояща се само от знак за подчертаване (анонимната променлива), и думите, започващи с **G**, следвано от споменатия знак (глобалните променливи);
- в) всяко от множествата F_n и P_n се състои от думи, които започват с малка латинска буква, не съдържат други знаци освен латински букви, знак за подчертаване и цифри и не са резервирани за вградени предикати на Strawberry Prolog;
- г) множеството L съдържа поне думата **not**, която е една от резервираните думи на Strawberry Prolog (всъщност даже и тя доста дълго време няма да ни е нужна);

За всяко множество C и всяко положително цяло число n ще означаваме с C^n множеството на всички наредени n -орки от елементи на C ; това множество се нарича *n -та декартова степен на C* (множеството C^1 ще отъждествяваме със самото множество C). Ако C и C' са някакви множества (не непременно различни помежду си), то за всяко положително цяло число n ще наричаме *n -местни операции от C към C'* изображенията на множеството C^n в множеството C' , а *нулместни операции от C към C'* ще наричаме елементите на C' . Множеството на всички n -местни операции от C към C' ще означаваме с $O_n(C, C')$. Под *n -местни операции* в дадено множество C ще разбираме n -местните операции от C към самото C , а под *n -местни предикати в C* – n -местните операции от C към двueleментното множество $\{0, 1\}$, състоящо се от числата 0 и 1 (когато имаме работа с предикати, интуитивно можем да тълкуваме числото 1 като "истина", а числото 0 – като "лъжа").

Когато C е дадено множество, а n е неотрицателно цяло число, ще наричаме *интерпретация в C на n -местните функционални символи* кое да е изображение на множеството F_n в множеството $O_n(C, C)$. C други думи, една интерпретация в C на n -местните функционални символи се определя като на всеки n -местен функционален символ се постави в съответствие някоя n -местна операция в C . Аналогично, под *интерпретация в C на n -местните предикатни символи* ще разбираме кое да е изображение на множеството P_n в множеството $O_n(C, \{0, 1\})$, т.е. една интерпретация в C на n -местните предикатни символи се определя като на всеки n -местен предикатен символ се постави в съответствие някой n -местен предикат в C .

Казваме, че е дадена една *структура*, когато е дадено едно непразно множество C и за всяко неотрицателно цяло число n са избрани по една интерпретация в C на n -местните функционални символи и по една интерпретация в C на n -местните предикатни символи. Самата структура може да се разглежда като наредена тройка с пръв член непразното множество C , втори член редица $j_0, j_1, j_2, \dots$ и трети член редица $p_0, p_1, p_2, \dots$, където за всяко неотрицателно цяло число n j_n и p_n са съответно интерпретация в C на n -местните функционални символи и интерпретация в C на n -местните

предикатни символи.Първият член на така описаната тройка (т.е. множеството C) се нарича *носител* на структурата.

Def

Термове са

- 1) константите
- 2) променливите
- 3) всички думи от вида $f(T_1, T_2, \dots, T_n)$, където n е положително цяло число, f е n -местен функционален символ и $T_1, T_2, \dots, T_n$ са термове

Def

Затворени термове (основни термове) са константите и всички думи от вида $f(T_1, T_2, \dots, T_n)$, където n е положително цяло число, f е n -местен функционален символ и $T_1, T_2, \dots, T_n$ са затворени термове.

Всички затворени термове са термове, а обратното не е вярно.

Def

Атомарни формули са нулместните предикатни символи и всички думи от вида $p(T_1, T_2, \dots, T_n)$, където n е положително цяло число, p е n -местен предикатен символ и $T_1, T_2, \dots, T_n$ са термове.

Def

Затворени атомарни формули (основни атомарни формули) – нулместните предикатни символи и всички думи от вида $p(T_1, T_2, \dots, T_n)$, където n е положително цяло число, p е n -местен предикатен символ и $T_1, T_2, \dots, T_n$ са затворени термове.

Всички затворени атомарни формули са атомарни формули. Ясно е, че всички термове и всички атомарни формули са префиксни изрази над множеството W , но има и много други префиксни изрази над това множество, (напр. изразите от вида $X(T_1, T_2, \dots, T_n)$, където X е променлива, n е положително цяло число и $T_1, T_2, \dots, T_n$ са термове).

Това, че термовете и атомарните формули са префиксни изрази над множеството W , заедно с еднозначния анализ на префиксните изрази над W осигурява еднозначното прочитане на термовете и на атомарните формули в следния смисъл: трите случая в дефиницията за терм се изключват взаимно, като в третия от тях числото n , функционалният символ f и термовете $T_1, T_2, \dots, T_n$ се определят еднозначно по дадения терм, а също двата случая в дефиницията за атомарна формула се изключват взаимно и във втория от тях числото n , предикатният символ p и термовете $T_1, T_2, \dots, T_n$ се определят еднозначно по дадената атомарна формула.

За всеки терм T ще дефинираме индуктивно едно множество $VAR(T)$, което ще наричаме *множество на променливите на T* . А именно:

- а) при $c \in F_0$ полагаме $VAR(c) = \emptyset$;
- б) при $X \in X$ полагаме $VAR(X) = \{X\}$;
- в) когато n е положително цяло число, $f \in F_n$ и $T_1, T_2, \dots, T_n$ са термове, полагаме

$$VAR(f(T_1, T_2, \dots, T_n)) = \bigcup_{i=1}^n VAR(T_i).$$

Така изказаната дефиниция определя еднозначно множеството $VAR(T)$ за произволен терм T благодарение на еднозначното прочитане на термовете.

Като използваме току-що дадената дефиниция и индукция, съобразена с дефиницията за терм, веднага се убеждаваме, че множеството на променливите на който да е терм е крайно подмножество на множеството X . Пак въз основа на горната дефиниция, но чрез индукция, съобразена с дефиницията за затворен терм, показваме, че множеството на променливите на който да е затворен терм е празно. Обратното също е вярно - ако множеството на променливите на един терм е празно, то този терм е затворен. В това можем да се убедим, като с помощта на горната дефиниция и на индукция, съобразена с дефиницията за терм, докажем за произволен терм, че той е затворен или множеството на променливите му има поне един елемент.

За всяка атомарна формула A дефинираме множество $\text{VAR}(A)$, което ще наричаме *множество на променливите на A* . Дефиницията е следната: когато $p \in P_0$, полагаме $\text{VAR}(p) = \emptyset$, а когато n е положително цяло число, $p \in P_n$ и $T_1, T_2, \dots, T_n$ са термове, полагаме

$$\text{VAR}(p(T_1, T_2, \dots, T_n)) = \bigcup_{i=1}^n \text{VAR}(T_i).$$

Множеството на променливите на коя да е атомарна формула е крайно подмножество на множеството X и че една атомарна формула е затворена точно тогава, когато множеството на нейните променливи е празно.

Дадена е една структура S с носител C , на която вторият и третият член са двете редици от интерпретации $j_0, j_1, j_2, \dots$ и $p_0, p_1, p_2, \dots$. За всеки затворен терм T дефинираме елемент T^S на множеството C , който ще наричаме *стойност на T в S* . Дефиницията е индуктивна: при $c \in F_0$ полагаме $c^S = j_0(c)$, а когато n е положително цяло число, $f \in F_n$ и $T_1, T_2, \dots, T_n$ са затворени термове, полагаме

$$f(T_1, T_2, \dots, T_n)^S = j_n(f)(T_1^S, T_2^S, \dots, T_n^S).$$

Разбира се еднозначността на тази дефиниция произтича от еднозначното прочитане на термовете. Сега ще дефинираме кога една затворена атомарна формула се счита *вярна в S* . Когато A е затворена атомарна формула, улавяме се да пишем $S \models A$, за да изразим, че A е вярна в S (за да изразим противното, ще пишем $S \not\models A$). При това улавяне дефиницията е следната: ако $p \in P_0$, считаме, че $S \models p$ точно тогава, когато $p_0(p) = 1$, а ако n е положително цяло число, $p \in P_n$ и $T_1, T_2, \dots, T_n$ са затворени термове, считаме, че $S \models p(T_1, T_2, \dots, T_n)$ точно тогава, когато

$$p_n(p)(T_1^S, T_2^S, \dots, T_n^S) = 1.$$

(по повод на еднозначността на тази дефиниция следва разбира се да се позовем на еднозначното прочитане на атомарните формули).

Def

Ако S е дадена структура и C е нейният носител, то ще наричаме *оценка в S на променливите* всяко изображение на множеството X в множеството C .

Да предположим сега, че освен структурата S е дадена и една оценка v в S на променливите. Тогава за всеки терм T ще дефинираме елемент $T^{S,v}$ на C , който ще наричаме *стойност на T в S при оценка v* . Дефиницията е пак индуктивна: при $c \in F_0$ полагаме $c^{S,v} = j_0(c)$, при $X \in X$ полагаме $X^{S,v} = v(X)$, а когато n е положително цяло число, $f \in F_n$ и $T_1, T_2, \dots, T_n$ са термове, полагаме

$$f(T_1, T_2, \dots, T_n)^{S,v} = j_n(f)(T_1^{S,v}, T_2^{S,v}, \dots, T_n^{S,v}).$$

Ще дефинираме и кога една произволна атомарна формула A е *вярна в S при оценка v* , което ще означаваме с $S, v \models A$ (противното ще означаваме с $S, v \not\models A$). Дефиницията е следната: ако $p \in P_0$, считаме, че $S, v \models p$ точно тогава, когато $p_0(p) = 1$, а ако n е положително цяло число, $p \in P_n$ и $T_1, T_2, \dots, T_n$ са термове, считаме, че $S, v \models p(T_1, T_2, \dots, T_n)$ точно тогава, когато

$$p_n(p)(T_1^{S,v}, T_2^{S,v}, \dots, T_n^{S,v}) = 1$$

(разбира се и тук е уместна бележка във връзка с еднозначността на дадените дефиниции).

Доказва се (с индукция по дефиницията на затворен терм), че за всеки затворен терм T имаме равенството $T^{S,v} = T^S$ при произволна оценка v в S на променливите. Следователно стойността на затворен терм в дадена структура е частен случай на стойност на терм в дадена структура при дадена оценка на променливите. Оттук веднага получаваме също, че за произволна затворена атомарна формула A и произволна оценка v в S на променливите условието $S, v \models A$ е равносилно с условието $S \models A$.

Лема за базисна роля на променливите на един терм

Нека S е дадена структура. Тогава за произволен терм T имаме, че ако две оценки v и v' в S на променливите съвпадат върху множеството $\text{VAR}(T)$, то е в сила равенството $T^{S,v} = T^{S,v'}$.

Доказателство:

Ще използваме индукция, съобразена с дефиницията за терм. Ако разглежданият терм е константа или

променлива, проверката на твърдението е непосредствена. Нека сега термът T да има вида $f(T_1, T_2, \dots, T_n)$, където n е положително цяло число, f е n -местен функционален символ и $T_1, T_2, \dots, T_n$ са термове, притежаващи доказаното свойство. Тогава и термът T ще има това свойство, защото ако две оценки v и v' в S на променливите съвпадат върху множеството $\text{VAR}(T)$, то при $i=1, \dots, n$ те ще съвпадат също върху неговото подмножество $\text{VAR}(T_i)$ и следователно ще имаме равенството $T_i^{S,v} = T_i^{S,v'}$, а оттук получаваме

$$T^{S,v} = j_n(f)(T_1^{S,v}, T_2^{S,v}, \dots, T_n^{S,v}) = j_n(f)(T_1^{S,v'}, T_2^{S,v'}, \dots, T_n^{S,v'}) = T^{S,v'}.$$

Като следствие от горната лема лесно се получава и подобно твърдение за атомарни формули, а именно:

Лема за базисна роля на променливите на една атомарна формула

Нека S е дадена структура. Тогава за произволна атомарна формула A имаме, че ако две оценки v и v' в S на променливите съвпадат върху множеството $\text{VAR}(A)$, то условието $S, v \models A$ е равносилно с условието $S, v' \models A$.

За една атомарна формула A ще казваме, че е *тъждествено вярна* в дадена структура S , ако A е вярна в S при всяка оценка в S на променливите.

Ясно е, че ако A е затворена атомарна формула, то A е тъждествено вярна в S точно тогава, когато A е вярна в S . По тази причина за произволна атомарна формула A се улавяме да пишем $S \models A$, за да изразим, че A е тъждествено вярна в S .

Хорнова цел (за краткост *цел*) ще наричаме всяка крайна редица от атомарни формули (вкл. едночленните редици и празната редица). Една цел ще наричаме *затворена*, ако всички нейни членове са затворени атомарни формули (тази дефиниция, приложена към празната цел, дава, че тя е затворена по тривиални съображения). За една затворена цел ще казваме, че е *изпълнена* в дадена структура S , ако всички членове на тази цел са верни в S (пак по тривиални съображения празната цел е изпълнена във всяка структура). За една каква да е (не непременно затворена) цел ще казваме, че е *изпълнена* в дадена структура S при дадена оценка v в S на променливите, ако всички членове на тази цел са верни в S при оценката v . За една цел ще казваме, че е *изпълнима* в дадена структура S , ако тази цел е изпълнена в S при някоя оценка на променливите. От предходния въпрос знаем, че ако A е затворена атомарна формула, то условието A да е вярна в дадена структура S при дадена оценка на променливите е равносилно с условието A да е вярна в S . По тази причина една затворена цел е изпълнима в дадена структура S точно тогава, когато тази цел е изпълнена в S .

Една непразна цел често ще я означаваме, като на самостоятелен ред напишем въпросителен знак, следван от тире, и след това последователните членове на целта със запетаи между тях и с точка на края. Празната цел ще означаваме със знака $[]$.

Пример

?- $d(a(X)), d(a(b(X)))$.

?- $d(s(e)), p(e, e)$.

Всяка наредена двойка, на която първият член е атомарна формула, а вторият е хорнова цел, ще наричаме *хорнова клауза* (за краткост *клауза*). Първия член на такава наредена двойка ще наричаме *глава* на клаузата (или нейно *заклучение*), а втория - нейно *тяло* (или нейна *предпоставка*). Прието е още клауза с празно тяло да се нарича *факт*, а клауза с непразно тяло - *правило*. Интуитивно можем да тълкуваме една клауза като твърдение, че ако са верни всички членове на нейното тяло, то е вярна и нейната глава, като в случай на клауза с празно тяло това всъщност е твърдение, че е вярна главата на клаузата. Точната дефиниция в подобен дух, с която ще си служим, е следната: ако S е дадена структура, то за една клауза ще казваме, че е *тъждествено вярна* в S , ако главата на тази клауза е вярна в S при всяка оценка на променливите, при която в S е изпълнено тялото на клаузата. За клауза с празно тяло разбира се това е равносилно с изискването атомарната формула, която е глава на клаузата, да бъде тъждествено вярна в S . Да отбележим още, че клауза, на която тялото не е изпълнимо в дадена структура S , по тривиални съображения трябва да се счита тъждествено вярна в S .

Една клауза се нарича *затворена*, ако са затворени както нейното тяло, така и нейната глава. За такава клауза често ще пропускаме думата "тъждествено" в термина "тъждествено вярна в S "; изискването за тъждествена вярност на клаузата в S в този случай е равносилно с това, че ако тялото на клаузата е

изпълнено в S , то главата на клаузата е вярна в S (разбира се, в специалния случай, когато тялото на въпросната клауза е празно, горното изискване от своя страна е равносилно с това главата на клаузата да е вярна в S , а пък в случая, когато тялото на клаузата не е изпълнено в S , изискването е удовлетворено по тривиални съображения).

Една клауза обикновено ще я означаваме, като на самостоятелен ред напишем главата на клаузата, следвана от точка в случай, че клаузата е факт, и следвана от знака $:-$ и след това последователните членове на тялото на клаузата със запетаи между тях и с точка на края в случай, че клаузата е правило.

Пример

$p(s(X), a(s(b(X))))$.

$p(s(X), a(b(X)))$.

$p(a(X), a(Y)) :- p(X, Y)$.

$p(b(X), b(Y)) :- p(X, Y)$.

$p(s(X), s(Y)) :- p(X, Y)$.

$d(s(e))$.

$d(Y) :- p(X, Y), d(X)$.

Множество на променливите на една цел G ще наричаме обединението на множествата на променливите на нейните членове; ще го означаваме с $VAR(G)$. *Множество на променливите на една клауза C* с глава H и тяло B ще наричаме обединението на множествата $VAR(H)$ и $VAR(B)$; ще го означаваме с $VAR(C)$. Веднага се съобразява, че множеството на променливите на всяка цел и на всяка клауза е крайно. Очевидно една цел е затворена точно тогава, когато множеството на нейните променливи е празно, и аналогично твърдение е в сила за клаузите.

Всяка крайна редица от хорнови клаузи (вкл. едночленните редици и празната редица) ще наричаме *хорнова програма* (за краткост *програма*). Например редицата, съставена от седемте клаузи в горния пример, представлява една хорнова програма. Една непразна хорнова програма обикновено ще я записваме като напишем на последователни отделни редове клаузите, които са нейните последователни членове (така както да речем са записани седемте клаузи в горния пример).

Модел за дадена хорнова програма P ще наричаме такава структура S , че всички клаузи, които са членове на редицата P , са тъждествено верни в S . Лесно се съобразява, че за всяка хорнова програма P и всяко непразно множество S съществува структура с носител S , която е модел за P . Една такава структура можем да получим, ако за интерпретация в S на всеки функционален символ (ако има такива) изберем произволна операция в S със съответния брой аргументи, за интерпретация на всеки нулместен предикатен символ (ако има такива) вземем числото 1, а за интерпретация на всеки друг предикатен символ - предикат в S , който има съответния брой аргументи и е тъждествено равен на 1. За някои хорнови програми съществуват и значително по-интересни модели. Все пак има и случаи, когато една хорнова програма няма други модели освен тривиалните, които описахме преди малко. Например ако сигнатурата е такава, че липсват функционални символи, множеството на едноместните предикатни символи се състои само от еднобуквената дума p , а при n , различно от 1, липсват n -местни предикатни символи, то за едночленната програма

$p(X)$.
единствените модели са такива структури, в които предикатният символ p се интерпретира чрез едноместен предикат, тъждествено равен на 1.

Когато искаме с методите на логическото програмиране да изследваме дали дадена цел е изпълнима в дадена конкретна структура S_0 , обикновено записваме подходящи краен брой свойства на S_0 посредством хорнови клаузи и след това се опитваме да решим въпроса въз основа на информацията, съдържаща се в тези клаузи. Споменатите краен брой клаузи, взети в някаква последователност, съставляват хорнова програма, за която структурата S_0 е модел. За тази програма обаче ще съществуват и други модели освен S_0 и може да се случи в някои от тях дадената цел да е изпълнима, а в други да не е. За една цел ще казваме, че е *изпълнима при* дадена програма P , ако тази цел е изпълнима във всеки модел за P . Когато програмата P е получена от дадена структура S_0 по описания преди малко начин (тъй че S_0 да е модел за P), ясно е че изпълнимостта на една цел при програмата P ще гарантира изпълнимост на тази цел и в структурата S_0 (разбира се би могло някой път да се случи една цел да е

изпълнима в S_0 и без да е изпълнима при програмата P).