ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Խաչատրյան Սուրեն Արտաշեսի

ՏՐԱՄԱԲԱՆԱԿԱՆ ԾՐԱԳՐԵՐԻ ՕՊՏԻՄԻԶԱՑՄԱՆ ՄԱՍԻՆ

Ե.13.04 – «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ
Ֆիզիկամաթեմատիկական գիտությունների թեկնածուի
գիտական աստիճանի հայցման ատենախոսության

Ս Ե Ղ Մ Ա Գ Ի Ր

Երևան – 2014

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Хачатрян Сурен Арташесович

ОБ ОПТИМИЗАЦИИ ЛОГИЧЕСКИХ ПРОГРАММ

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени кандидата
физико-математических наук по специальности
05.13.04 – “Математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей”

Ереван – 2014

Ատենախոսության թեման հաստատվել է Երևանի պետական համալսարանում

Գիտական ղեկավար՝	ֆիզ.մաթ. գիտ. դոկտոր	Ս. Ա. Նիգիյան
Պաշտոնական ընդդիմախոսներ՝	ֆիզ.մաթ. գիտ. դոկտոր	Ի. Դ. Զասլավսկի
	ֆիզ.մաթ. գիտ. թեկնածու	Հ. Ռ. Բոլիբեկյան
Առաջատար կազմակերպություն՝	Հայ-Ռուսական (Սլավոնական) համալսարան	

Պաշտպանությունը կայանալու է 2014թ. հուլիսի 10-ին, ժ. 16:00-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գործող 037 «Ինֆորմատիկա և հաշվողական համակարգեր» մասնագիտական խորհրդի նիստում հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակի 1:

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:

Սեղմագիրը առաքված է 2014թ. հունիսի 9-ին:

Մասնագիտական խորհրդի գիտական
քարտուղար, ֆիզ.մաթ. գիտ. դոկտոր Հ. Գ. Սարուխանյան

Тема диссертации утверждена в Ереванском государственном университете

Научный руководитель:	доктор физ.-мат. наук	С. А. Нигиян
Официальные оппоненты:	доктор физ.-мат. наук	И. Д. Заславский
	кандидат физ.-мат. наук	О. Р. Болибекян
Ведущая организация:	Российско-Армянский (Славянский) университет	

Защита состоится 10 июля 2014г. в 16:00 на заседании специализированного совета 037 «Информатика и вычислительные системы» Института проблем информатики и автоматизации НАН РА по адресу: 0014, г. Ереван, ул. П. Севака 1.

С диссертацией можно ознакомиться в библиотеке ИПИА НАН РА.

Автореферат разослан 9-го июня 2014г.

Ученый секретарь специализированного
совета, доктор физ.-мат. наук А. Г. Саруханян

ԱՇԽԱՏԱՆՔԻ ԸՆԴՀԱՆՈՒՐ ԲՆՈՒԹԱԳԻՐԸ

Թեմայի արդիականությունը: Աշխատանքը նվիրված է տրամաբանական ծրագրավորման խնդիրներին: Հետազոտության առարկա են հանդիսանում տրամաբանական ծրագրերը և հարցումները, իսկ դիտարկվող հիմնական խնդիրը՝ տրամաբանական ծրագրերի օպտիմիզացումը, այն իմաստով, որ ծրագրի և իր ցանկացած թույլատրելի հարցման ցանկացած SLD-ծառ դառնա վերջավոր: Այլ կերպ ասած, ծրագիրը դառնա որոշված (terminating) իր թույլատրելի հարցումների նկատմամբ:

Տրամաբանական ծրագրավորումը սկիզբ է առնում Ռ. Կովալսկիի[1] և Ա. Կոլմերոյի[2] աշխատանքներից, որտեղ առաջարկվում է Prolog լեզուն և դրվում դրա տեսական հիմքերը: Տրամաբանական ծրագրավորման համակարգերի հիմնական խնդիրն է պարզել արդյո՞ք հարցումը տրամաբանորեն հետևում է ծրագրից թե ոչ, և եթե այո, ապա հարցման փոփոխականների ո՞ր արժեքների դեպքում: Տրամաբանական ծրագրավորման համակարգերում որպես արտածման կանոն հիմնականում օգտագործվում է SLD-ռեզոլյուցիան (Selective Linear Definite clause resolution), որը նկարագրվել է Ռ. Կովալսկիի աշխատանքում: SLD-ռեզոլյուցիայի դերը տրամաբանական ծրագրավորման մեջ ամրապնդվում է [3] աշխատանքում, որտեղ սահմանվում են ծրագրերի տրամաբանական և անշարժ կետի սեմանտիկաները և ցույց է տրվում, որ այս սեմանտիկաները համընկնում են SLD-ռեզոլյուցիայի միջոցով տրվող պրոցեդուրային սեմանտիկայի հետ: Սակայն SLD-ռեզոլյուցիայի վրա հիմնված տրամաբանական ծրագրավորման համակարգերը մուտքում ստանալով ծրագիր և հարցում կարող են աշխատել անվերջ: Առաջանում է խնդիր՝ օպտիմիզացնել ծրագրերը այնպես, որ ծրագրի և իր ցանկացած թույլատրելի հարցման ցանկացած SLD-ծառ դառնա վերջավոր:

Հաշվի առնելով Չորչի թեորեմը, կարելի է ակնկալել, որ բավականաչափ հարուստ լեզուների համար դրված օպտիմիզացման խնդիրը կլինի անլուծելի: Իրոք՝ [4] աշխատանքից հետևում է, որ այն անլուծելի է: Հետևաբար բնական է ուսումնասիրել խնդիրը ծրագրերի որոշ դասերի համար:

Մ. Բեզեմի[5] աշխատանքում նկարագրված են ռեկուրենտ ծրագրերի և սահմանափակ հարցումների գաղափարները: Ցույց է տրվում, որ ռեկուրենտ ծրագրերը որոշված են սահմանափակ հարցումների նկատմամբ:

[1] R. A. Kowalski. “Predicate Logic as Programming Language”. In: *Proceedins of IFIP Congress.* North-Holland Publishing Company, 1974, p. 569–574.

[2] A. Colmerauer, H. Kanoui, P. Roussel, R. Pasero. *Un systéme de communication homme-machine en Français.* Tech. rep. Université de Aix-Marseille: Groupe de recherche en Intelligence Artificielle, 1973.

[3] K. R. Apt, M. H. van Emden. “Contributions to the Theory of Logic Programming”. *Journal of the ACM* 29.3 (1982), p. 841–862.

[4] С. А. Нигиян. “Интерпретатор ПРОЛОГа с точки зрения логической семантики”. *Программирование* 1994.2 (1994), p. 64–73.

[5] M. A. Bezem. “Strong Termination of Logic Programs”. *Journal of Logic Programming* №15 (1993), p. 79–97.

Մի շարք[1,2] աշխատանքներում նկարագրվում են ծրագրերի և հարցումների դասեր, որոնք ձևափոխությամբ հնարավոր է բերել ռեկուրենտ ծրագրերի և սահմանափակ հարցումների: Նշված աշխատանքներում դիտարկվող դասերը (այդ թվում նաև ռեկուրենտ ծրագրերն ու սահմանափակ հարցումները) բնութագրվում են դեկլարատիվ կերպով:

Այս աշխատանքում դիտարկվում է օպտիմիզացման խնդիրը ծրագրերի հետևյալ երեք կոնկրետ դասերի համար. փոփոխականներ չօգտագործող ծրագրեր, մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող ծրագրեր[3], և մոնադիկ (մեկից մեծ տեղայնությամբ ֆունկցիոնալ սիմվոլներ չօգտագործող և միայն մեկ տեղանի պրեդիկատային սիմվոլներ օգտագործող) ծրագրեր[4]: Տրվում են ձևափոխություններ, որոնցով այս դասերի ծրագրերն ու իրենց թույլատրելի հարցումները ձևափոխվում են ռեկուրենտ ծրագրերի և սահմանափակ հարցումների՝ օգտագործելով սահմանափակիչներով տրամաբանական ծրագրավորման (constraint logic programming) մեթոդները[5]: Արդյունքում, նրանց համար լուծվում է դրված օպտիմիզացման խնդիրը: Հարցումը թույլատրելի է ծրագրի համար, եթե այն օգտագործում է միայն այն պրեդիկատային սիմվոլները, որոնք օգտագործվում են այդ ծրագրում: Նկատենք, որ միայն թույլատրելի հարցումների դիտարկումը ընդհանրությունը չի խախտում, քանզի եթե հարցումը թույլատրելի չէ, ապա ակնհայտորեն այն տրամաբանորեն չի հետևում ծրագրից:

Աշխատանքի նպատակն ու խնդիրները: Աշխատանքի հիմնական նպատակն ու խնդիրները հետևյալն են.

1. Հետազոտել տրամաբանական ծրագրերի օպտիմիզացման խնդիրը փոփոխականներ չօգտագործող ծրագրերի և այդ ծրագրերի թույլատրելի հարցումների համար:

2. Հետազոտել տրամաբանական ծրագրերի օպտիմիզացման խնդիրը մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող ծրագրերի և այդ ծրագրերի թույլատրելի հարցումների համար:

3. Հետազոտել տրամաբանական ծրագրերի օպտիմիզացման խնդիրը մոնադիկ ծրագրերի և այդ ծրագրերի թույլատրելի, փոփոխականներ չօգտագործող մոնադիկ հարցումների համար:

[1] D. Pedreschi, S. Ruggieri. "Bounded Nondeterminism of Logic Programs". In: *Proc. of the International Conference on Logic Programming.* MIT Pres, 1999, p. 350–364.

[2] D. Pedreschi, S. Ruggieri, J. G. Smaus. "Classes of Terminating Logic Programs". *Theory and Practice of Logic Programming* 2.3 (2002), p. 369–418.

[3] С. А. Нигиян, Л. О. Хачоян. "К проблеме Δ-эквивалентности логических программ". *Доклады национальной академми наук Армении* 99.2 (1999), p. 99–103.

[4] A. B. Matos. "Monadic Logic Programs and Functional Complexity". *Theoretical Computer Science* 176 (1997), p. 175–204.

[5] K. Marriot, P. J. Stuckey. *Programming with Constraints: An Introduction.* MIT Press, 1998.

Հետազոտության մեթոդները: Աշխատանքում օգտագործված հետազոտության մեթոդները ներառում են տրամաբանական ծրագրավորման, մաթեմատիկական տրամաբանության, ալգորիթմների տեսության և հանրահաշվի մեթոդները:

Արդյունքների նորությունը: Ատենախոսության հիմնական արդյունքները հետևյալն են.

1. Նկարագրվել է ծրագրի ձևափոխություն, որով ցանկացած փոփոխականներ չօգտագործող P ծրագիր փոխակերպվում է մեկ այլ՝ P' ծրագրի, այնպես որ P ծրագրի ցանկացած թույլատրելի G հարցման համար, G հարցումը տրամաբանորեն հետևում է P ծրագրից այն և միայն այն դեպքում, երբ այն տրամաբանորեն հետևում է P' ծրագրից, և ցույց է տրվում, որ (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է:

2. Նկարագրվել է ծրագրի ձևափոխություն, որով ցանկացած մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող P ծրագիր փոխակերպվում է մեկ այլ՝ P' ծրագրի, այնպես որ P ծրագրի ցանկացած թույլատրելի G հարցման համար, G հարցումը տրամաբանորեն հետևում է P ծրագրից այն և միայն այն դեպքում, երբ այն տրամաբանորեն հետևում է P' ծրագրից, և ցույց է տրվում, որ (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է:

3. Նկարագրվել են ծրագրի և հարցման ձևափոխություններ, որով ցանկացած P մոնադիկ ծրագիր և G թույլատրելի, փոփոխականներ չօգտագործող մոնադիկ հարցում փոխակերպվում է P' ծրագրի և G' հարցման, այնպես որ G հարցումը տրամաբանորեն հետևում է P ծրագրից այն և միայն այն դեպքում, երբ G' հարցումը տրամաբանորեն հետևում է P' ծրագրից, և ցույց է տրվում, որ (P', G') զույգի ցանկացած SLD-ծառ վերջավոր է: Նշենք, որ ձևափոխությամբ ստացված P' ծրագիրը նույնն է բոլոր դիտարկվող հարցումների համար:

Տեսական և կիրառական նշանակությունը: Աշխատանքում ստացված արդյունքներն ունեն ինչպես տեսական, այնպես էլ կիրառական նշանակություն: Դրանք կարող են օգտագործվել ինչպես տրամաբանական ծրագրավորման նոր համակարգերի ստեղծման ժամանակ, այնպես էլ արդեն գոյություն ունեցող համակարգերի համար:

Ստացված արդյունքների ապրոբացիան: Ատենախոսության հիմնական արդյունքները զեկուցվել են ԵՊՀ ծրագրավորման և ինֆորմացիոն տեխնոլոգիաների ամբիոնի սեմինարում, ԵՊՀ ինֆորմատիկայի և կիրառական մաթեմատիկայի ֆակուլտետի ընդհանուր սեմինարում և Computer Science and Information Technologies – 2011 (CSIT, Երևան, 2011), Computer Science and Information Technologies – 2013 (CSIT, Երևան, 2013) միջազգային գիտաժողովներում: Ատենախոսության հիմնական արդյունքներն ընդգրկված են հրատարակված չորս աշխատանքներում:

Աշխատանքի կառուցվածքն ու ծավալը: Ատենախոսությունը բաղկացած է ներածությունից, չորս գլուխներից, եզրակացությունից և օգտագործված գրականության ցանկից (46 անուն): Ատենախոսության ծավալը 80 էջ է:

ԱՇԽԱՏԱՆՔԻ ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆԸ

Ներածությունում նկարագրվում է ատենախոսության հետազոտության հիմնական նպատակն ու խնդիրները, հիմնավորվում է ատենախոսության թեմայի արդիականությունը և նորությունը: Համառոտ կերպով ներկայացվում է աշխատանքի բովանդակությունը:

Գլուխ 1-ում բերվում են անհրաժեշտ սահմանումներ և արդյունքներ մաթեմատիկական տրամաբանության և տրամաբանական ծրագրավորման վերաբերյալ, որոնք հիմնականում վերցված են [1,2] աշխատանքներից: Առաջին գլուխը բաղկացած է չորս բաժիններից:

1.1 բաժնում ներմուծվում են անհրաժեշտ տեղեկություններ առաջին կարգի պրեդիկատների տրամաբանությունից:

Ֆիքսենք հետևյալ երեք իրար հետ չհատվող հաշվելի բազմությունները՝ $\mathbb{X}$, Φ, և Π, որտեղ

- $\mathbb{X}$-ը փոփոխականների բազմությունն է,
- Φ-ն ֆունկցիոնալ սիմվոլների բազմությունն է, համապատասխանեցված ամեն մի ֆունկցիոնալ սիմվոլին բնական թիվ՝ նրա տեղայնությունը, ընդ որում ցանկացած $n \geq 0$ համար Φ-ում առկա n տեղանի ֆունկցիոնալ սիմվոլների բազմությունը հաշվելի է: Զրո տեղանի ֆունկցիոնալ սիմվոլները կոչվում են նաև հաստատուններ:
- Π-ն պրեդիկատային սիմվոլների բազմությունն է, համապատասխանեցված ամեն մի պրեդիկատային սիմվոլին բնական թիվ՝ նրա տեղայնությունը, ընդ որում ցանկացած $n \geq 0$ համար Π-ում առկա n տեղանի պրեդիկատային սիմվոլների բազմությունը հաշվելի է:

Հետագայում, ասելով փոփոխական կհասկանանք փոփոխական $\mathbb{X}$ բազմությունից, ֆունկցիոնալ սիմվոլ՝ Φ բազմությունից, և պրեդիկատային սիմվոլ՝ Π բազմությունից:

Սահմանում 1.1.1 (**Թերմ**): *Թերմերը որոշվում են հետևյալ կանոններով.*

- *յուրաքանչյուր փոփոխական թերմ է,*
- *յուրաքանչյուր հաստատուն թերմ է,*
- *եթե f-ը n տեղանի ֆունկցիոնալ սիմվոլ է, որտեղ $n > 0$, իսկ $t_1, ..., t_n$-ը թերմեր են, ապա $f(t_1, ..., t_n)$-ը թերմ է:*

[1] K. R. Apt. "Introduction to Logic Programming". In: *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics.* Ed. by J. van Leeuwen. Elsevier and MIT press, 1990, p. 493–574.

[2] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 1984.

Թերմերը հաճախ կնշանակենք $t, \tau, \ldots$ տառերով՝ գուցե որոշ ինդեքսներով: Թերմերի $< t_1, \ldots, t_n >$ n-յակներն էլ, որտեղ $n \geq 0$, կնշանակենք $\bar{t}$-ով:

Սահմանում 1.1.2 (**Բանաձև**): *Բանաձևերը որոշվում են հետևյալ կանոններով.*

- *յուրաքանչյուր զրո տեղանի պրեդիկատային սիմվոլ բանաձև է,*
- *եթե p-ն n տեղանի պրեդիկատային սիմվոլ է, որտեղ $n > 0$, իսկ $t_1, ..., t_n$-ը թերմեր, ապա $p(t_1, ..., t_n)$-ը բանաձև է,*
- *եթե F_1-ն և F_2-ն բանաձևեր են, ապա $(\neg F_1)$-ը, $(F_1 \wedge F_2)$-ն և $(F_1 \vee F_2)$-ն նույնպես բանաձևեր են,*
- *եթե F-ը բանաձև է, իսկ x-ը փոփոխական, ապա $(\forall x F)$-ը և $(\exists x F)$-ը նույնպես բանաձևեր են:*

Առաջին երկու կանոններով ստացվող բանաձևերը կոչվում են *ատոմար բանաձևեր* կամ *ատոմներ*: Բանաձևերում փոփոխականների ազատ և կապված մուտքերի, ինչպես նաև, ազատ և կապված փոփոխաականների գաղափարները սահմանվում են ավանդական եղանակով: Բանաձևը, որը չունի ազատ փոփոխականներ կոչվում է *փակ*:

Սահմանենք քվանտորների և տրամաբանական կապերի հետևյալ առաջնայնությունը (նվազման կարգով).

$$\forall, \exists, \neg, \wedge, \vee$$

և պայմանավորվենք սղել փակագծերը, եթե երկիմաստություն չի առաջանում: $\forall x_1 \ldots \forall x_n F$ տեսքի բանաձևը, որտեղ $x_1, ..., x_n$-ը F բանաձևի բոլոր ազատ փոփոխականներն են, կրճատ կնշանակենք $\forall F$-ով: Համգունորեն $\exists x_1 \ldots \exists x_n F$ տեսքի բանաձևը կնշանակենք $\exists F$-ով:

Թերմերը և ատոմները, որոնք չեն պարունակում փոփոխականներ, կոչվում են *հիմնական*: Բոլոր հիմնական թերմերի բազմությունը անվանվում է *Հերբրանի ունիվերսում*, որը կնշանակենք $\mathbb{U}$-ով: Բոլոր հիմնական ատոմների բազմությունը անվանվում է *Հերբրանի հիմք*, որն էլ կնշանակենք $\mathbb{B}$-ով:

Սահմանենք հիմնական թերմերի խորության՝ $depth : \mathbb{U} \rightarrow \mathbb{N}$ ֆունկցիան հետևյալ կերպ.

$$depth(f\ (\tau_1, \ldots, \tau_n)) = \begin{cases} 1 + max\{depth(\tau_i) | i = 1, \ldots n\}, & \text{եթե n>0} \\ 0, & \text{եթե n=0} \end{cases}$$

Օրինակ, $depth\,(s\,(s\,(0))) = 1 + depth\,(s\,(0)) = 1 + 1 + depth\,(0) = 1 + 1 + 0 = 2.$

Սահմանում 1.1.3 (**Հերբրանի ինտերպրետացիա**): *Հերբրանի ինտերպրետացիան՝ $\mathcal{I}$, սահմանվում է հետևյալ կերպ.*

- *տարրկայական բազմությունը $\mathbb{U}$-ն է,*

- *յուրաքանչյուր c հաստատունի համապատասխանում է հենց ինքը՝ $c^{\mathcal{I}} = c$,*
  յուրաքանչյուր $n > 0$ տեղանի f ֆունկցիոնալ սիմվոլին համապատասխանում է $f^{\mathcal{I}} : \mathbb{U}^n \to \mathbb{U}$ արտապատկերում, որը $\langle t_1, ..., t_n \rangle$ n-յակին համապատասխանեցնում է $f(t_1, ..., t_n)$ թերմը, բոլոր $t_1, ..., t_n \in \mathbb{U}$ թերմերի համար:

- *յուրաքանչյուր զրո տեղանի q պրեդիկատային սիմվոլին համապատասխանեցնում է $q^{\mathcal{I}} \in \{\text{True}, \text{False}\}$ ճշտության արժեք,*
  յուրաքանչյուր $n > 0$ տեղանի p պրեդիկատային սիմվոլին համապատասխանեցնում է $p^{\mathcal{I}} : \mathbb{U}^n \to \{\text{True}, \text{False}\}$ արտապատկերում:

Այսուհետ ասելով ինտերպրետացիա կհասկանանք հենց Հերբրանի ինտերպրետացիան:

Փակ բանաձևի արժեքը ինտերպրետացիայի վրա սահմանվում է ավանդական եղանակով: Ինտերպրետացիան, որի վրա F փակ բանաձևն ընդունում է True արժեք, կոչվում է F բանաձևի մոդել: Կասենք, որ ինտերպրետացիան հանդիսանում է W փակ բանաձևերի ոչ դատարկ բազմության մոդել, եթե այն հանդիսանում է W բազմության յուրաքանչյուր բանաձևի մոդել:

Սահմանում 1.1.5 (**Տրամաբանական հետևանք**): *F փակ բանաձևը հանդիսանում է W փակ բանաձևերի ոչ դատարկ բազմության տրամաբանական հետևանք (կնշանակենք $W \models F$), եթե W-ի ցանկացած մոդել հանդիսանում է F-ի մոդել: Այդ դեպքում կասենք նաև, որ F-ը տրամաբանորեն հետևում է W-ից:*

1.2 բաժնում ներմուծվում են անհրաժեշտ տեղեկություններ տրամաբանական ծրագրավորումից:

Լիտերալը ատոմ է (դրական լիտերալ) կամ ատոմի ժխտում (բացասական լիտերալ): $\forall(L_1 \vee ... \vee L_n)$ տեսքի բանաձևը, որտեղ $n > 0$ և $L_1, ..., L_n$-ը լիտերալներ են, կոչվում է *դիզյունկտ*: Այն դիզյունկտը, որը պարունակում է առավելագույնը մեկ դրական լիտերալ կոչվում է *Հորնի դիզյունկտ*: Հորնի դիզյունկտը, որն ունի ճիշտ մեկ դրական լիտերալ կոչվում է *ծրագրի նախադասություն* (կարճ՝ *նախադասություն*): $\forall(A \vee \neg B_1 \vee ... \vee \neg B_m)$ ծրագրի նախադասությունը կգրենք հետևյալ կերպ.

$$A \leftarrow B_1, ..., B_m,$$

որտեղ $m \geq 0$, $A, B_1, ..., B_m$ ատոմներ են: A ատոմը կոչվում է նախադասության *գլուխ*, իսկ $B_1, ..., B_m$-ը՝ *մարմին*: $m = 0$ դեպքում այն կոչվում է *փաստ*, որը կնշանակենք A-ով, իսկ $m > 0$ դեպքում՝ *կանոն*:

Սահմանում 1.2.1 (**Տրամաբանական ծրագիր**): *Տրամաբանական ծրագիրը նախադասությունների ոչ դատարկ վերջավոր բազմություն է:*

Սահմանում 1.2.2 (**Հարցում**): *Ոչ դատարկ հարցումը $\exists(C_1 \wedge ... \wedge C_k)$ տեսքի բանաձև է, որտեղ $k > 0$, $C_1, ..., C_k$ ատոմներ են: Այն կգրենք որպես*

$$\leftarrow C_1, ..., C_k :$$

$k = 0$ դեպքում այն անվանվում է դատարկ հարցում, որն էլ կգրենք որպես $\leftarrow$:

Դիցուք P-ն տրամաբանական ծրագիր է: P-ում օգտագործվող ֆունկցիոնալ սիմվոլների բազմությունը նշանակենք Φ_P-ով, իսկ պրեդիկատային սիմվոլների բազմությունը՝ Π_P-ով:

Սահմանում 1.2.3 (Թույլատրելի հարցում): *P տրամաբանական ծրագրի համար, հարցումը կանվանենք թույլատրելի, եթե այն օգտագործում է պրեդիկատային սիմվոլներ միայն Π_P բազմությունից: P ծրագրի թույլատրելի հարցումների բազմությունը կնշանակենք $\Delta(P)$-ով:*

Սահմանում 1.2.4 (Տեղադրություն): *Տեղադրությունն իրենից ներկայացնում է զույգերի վերջավոր $\{x_1/t_1, ..., x_n/t_n\}$ բազմություն, որտեղ $n \geq 0$, $x_1, ..., x_n$-ը իրարից տարբեր փոփոխականներ են, իսկ $t_1, ..., t_n$-ը՝ այնպիսի թերմեր, որ $x_i \neq t_i$, $i = 1, \ldots, n$:*

$\{x_1/t_1, ..., x_n/t_n\}$ տեղադրությունը կանվանենք $x_1, \ldots, x_n$ փոփոխականների տեղադրություն: Այն կոչվում է *հիմնական*, եթե $t_1, ..., t_n$ թերմերը հիմնական են: Եթե $\theta = \{x_1/t_1, ..., x_n/t_n\}$ տեղադրություն է, իսկ A-ն՝ ատոմ, ապա $A\theta$-ով կնշանակենք այն ատոմը, որը ստացվում է A-ից $x_1, ..., x_n$ փոփոխականների միաժամանակյա փոխարինումով համապատասխանաբար $t_1, ..., t_n$ թերմերով: $A\theta$ ատոմը կոչվում է A ատոմի նմուշ: Նշանակենք $ground(A)$-ով A ատոմի բոլոր հիմնական նմուշների բազմությունը:

Դիցուք $\theta = \{x_1/t_1, ..., x_n/t_n\}$ և $\sigma = \{y_1/r_1, ..., y_m/r_m\}$ տեղադրություններ են: θ և σ տեղադրությունների $\theta\sigma$ *կոմպոզիցիան* տեղադրություն է, որը ստացվում է

$$\{x_1/t_1\sigma, ..., x_n/t_n\sigma, y_1/r_1, ..., y_m/r_m\}$$

բազմությունից հեռացնելով $x_i/t_i\sigma$ զույգերը ($i = 1, ..., n$), որոնց համար $x_i = t_i\sigma$ և y_j/r_j զույգերը ($j = 1, ..., m$), որոնց համար $y_j \in \{x_1, ..., x_n\}$:

A և B ատոմները կոչվում են *ունիֆակացվող*, եթե գոյություն ունի θ տեղադրություն այնպես, որ $A\theta = B\theta$: θ տեղադրությունը այդ դեպքում կոչվում է A և B ատոմների *ունիֆիկատոր*: Եթե θ ունիֆիկատորն այնպիսին է, որ A և B ատոմների ցանկացած σ ունիֆակատորի համար գոյություն ունի γ տեղադրություն այնպես, որ $\sigma = \theta\gamma$, ապա θ-ն կոչվում է A և B ատոմների *ամենաընդհանուր ունիֆիակատոր* և նշանակվում $\mathrm{mgu}(A, B)$-ով:

A ատոմում օգտագործվող փոփոխականների բազմությունը նշանակենք $Var(A)$-ով: $A \leftarrow B_1, ..., B_m$, $m \geq 0$ տեսքի S նախադասության համար, սահմանենք այդ նախադասության մեջ օգտագործվող փոփոխականների բազմությունը՝ $Var(S)$, հետևյալ կերպ.

$$Var(S) = Var(A) \cup Var(B_1) \cup \cdots \cup Var(B_m):$$

$\leftarrow C_1, ..., C_k$, $k > 0$ տեսքի G հարցման մեջ օգտագործվող փոփոխականների բազմությունն էլ կնշանակենք $Var(G)$-ով, և կսահմանենք հետևյալ կերպ.

$$Var(G) = Var(C_1) \cup \cdots \cup Var(C_k) :$$

1.3 բաժնում ներմուծվում են SLD-ռեզոլյուցիայի, SLD-արտածման, SLD-ծառի և որոշվածության գաղափարները:

Սահմանում 1.3.1 (**SLD-ռեզոլյուցիա**): *Դիցուք G-ն $\leftarrow C_1, \ldots, C_k$ ոչ դատարկ հարցումն է, իսկ S-ը $A \leftarrow B_1, \ldots, B_m$, $m \geq 0$ նախադասությունը: Այդ դեպքում, կասենք որ Q հարցումը ստացվել է G հարցումից և S նախադասությունից SLD-ռեզոլյուցիայի միջոցով, եթե*

- $Var(G) \cap Var(S) = \varnothing$,
- *C_i և A ատոմները ունիֆիկացվող են, որտեղ C_i-ն ատոմ է G-ից, որը կոչվում է ընտրված ատոմ,* $1 \leq i \leq k$,
- *Q-ն $\leftarrow C_1\theta, \ldots, C_{i-1}\theta, B_1\theta, \ldots, B_m\theta, C_{i+1}\theta, \ldots, C_k\theta$ հարցումն է, որտեղ $\theta = \mathrm{mgu}(C_i, A)$:*

C_i ատոմի ընտրությունը կատարվում է այսպես կոչված *ընտրման ֆունկցիայի* միջոցով: Q-ն կոչվում է G հարցման և S նախադասության SLD-ռեզոլվենտ:

Սահմանում 1.3.2 (**SLD-արտածում**): *Դիցուք P-ն ծրագիր է, իսկ G-ն ոչ դատարկ հարցում: (P, G) զույգից SLD-արտածում անվանվում է այն հարցումների վերջավոր կամ անվերջ $G_0, G_1, G_2 \ldots$ հաջորդականությունը, որ $G_0 = G$ և յուրաքանչյուր G_{i+1} հանդիսանում է G_i հարցման և P ծրագրի որևէ նախադասության SLD-ռեզոլվենտ ($i \geq 0$):*

Սահմանում 1.3.3 (**SLD-ծառ**): *Դիցուք P-ն ծրագիր է, իսկ G-ն ոչ դատարկ հարցում: (P, G) զույգի SLD-ծառ անվանվում է այն ծառը, որը բավարարում է հետևյալ պայմաններին՝*

- *ծառի ամեն մի հանգույց հարցում է (հնարավոր է նաև դատարկ),*
- *ծառի արմատը G հարցումն է,*
- *եթե $\leftarrow C_1, \ldots, C_k$ ոչ դատարկ հարցումը ծառի որևէ հանգույց է, իսկ C_i-ն՝ ընտրված ատոմ ($1 \leq i \leq k$), ապա P ծրագրի յուրաքանչյուր $A \leftarrow B_1, \ldots, B_m$, $m \geq 0$ նախադասության համար, որի գլուխը՝ A-ն և C_i-ն ունիֆիկացվող են, $\leftarrow C_1, \ldots, C_k$ հանգույցը ունի զավակ՝ $\leftarrow C_1\theta, \ldots, C_{i-1}\theta, B_1\theta, \ldots, B_m\theta, C_{i+1}\theta, \ldots, C_k\theta$, որտեղ $\theta = \mathrm{mgu}(C_i, A)$:*
- *հանգույցները, որոնք դատարկ հարցումներ են չունեն զավակներ:*

Նշենք, որ SLD-ծառի յուրաքանչյուր ճյուղ իրենից ներկայացնում է SLD-արտածում: Նշենք նաև, որ նույն ծրագրին և հարցմանը կարող են համապատասխանել տարբեր SLD-ծառեր՝ կախված ընտրման ֆունկցիայից:

Սահմանում 1.3.4 (**Որոշվածություն**): *P ծրագիրը կանվանենք որոշված $\mathcal{G}$ ոչ դատարկ հարցումների բազմության նկատմամբ, եթե ցանկացած $G \in \mathcal{G}$ հարցման համար, (P, G) զույգի ցանկացած SLD-ծառ վերջավոր է:*

1.4 բաժնում ներմուծվում են ատոմի մակարդակի, սահմանափակ ատոմի, ռեկուրենտ ծրագրերի և սահմանափակ հարցումների գաղափարները:

Սահմանում 1.4.1 (**Ատոմի մակարդակ**): *Ատոմի մակարդակը՝ $||$, ֆունկցիա է, որը ցանկացած հիմնական ատոմի համապատասխանության մեջ է դնում որևէ բնական թիվ՝ $|| : \mathbb{B} \to \mathbb{N}$: A հիմնական ատոմի մակարդակը կնշանակենք $|A|$:*

Սահմանում 1.4.2 (**Սահմանափակ ատոմ**): *A ատոմը կոչվում է սահմանափակ (ըստ $||$ ֆունկցիայի), եթե գոյություն ունի $k \in \mathbb{N}$, այնպես որ ցանկացած $A_0 \in ground(A)$ ատոմի համար $|A_0| \leq k$: Եթե A ատոմը սահմանափակ է, ապա*

$$|A| = max\{|A_0| \mid A_0 \in ground(A)\} :$$

Սահմանում 1.4.3 (**Սահմանափակ հարցում**): *$\leftarrow C_1, \ldots, C_k$ հարցումը կոչվում է սահմանափակ (ըստ $||$ ֆունկցիայի), եթե ամեն մի C_i ատոմ սահմանափակ է (ըստ $||$ ֆունկցիայի), $i = 1, \ldots, k$:*

Սահմանում 1.4.4 (**Ռեկուրենտ նախադասություն**): *$A \leftarrow B_1, \ldots, B_m$, $m \geq 0$ նախադասությունը կոչվում է ռեկուրենտ (ըստ $||$ ֆունկցիայի), եթե այդ նախադասության փոփոխականների ցանկացած θ հիմնական տեղադրության համար, $|A\theta| > |B_i\theta|$ բոլոր $i = 1, \ldots, m$ համար:*

Սահմանում 1.4.5 (**Ռեկուրենտ ծրագիր**): *Ծրագիրը կոչվում է ռեկուրենտ (ըստ $||$ ֆունկցիայի), եթե նրա բոլոր նախադասությունները ռեկուրենտ են (ըստ $||$ ֆունկցիայի):*

Թեորեմ 1.4.2:[1] *Ռեկուրենտ ծրագրի և սահմանափակ հարցման զույգից ցանկացած SLD-արտածում վերջավոր է:*

Գլուխ 2-ում ուսումնասիրվում են փոփոխականներ չօգտագործող ծրագրերը: Ցույց է տրվում, որ ընդհանուր առմամբ փոփոխականներ չօգտագործող ծրագրերը իրենց թույլատրելի հարցումների նկատմամբ որոշված չեն: Նկարագրվում է ձևափոխության ալգորիթմ, որով ցանկացած փոփոխականներ չօգտագործող P ծրագիր փոխակերպվում է մեկ այլ P' ծրագրի: Ապացուցվում է, որ P ծրագրի ցանկացած թույլատրելի G հարցման համար, $P \models G$ այն և միայն այն դեպքում, երբ $P' \models G$: Ցույց է տրվում, որ (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է: Երկրորդ գլուխը բաղկացած է չորս բաժիններից:

[1]M. A. Bezem. “Strong Termination of Logic Programs”. *Journal of Logic Programming* №15 (1993), p. 79–97.

2.1 բաժնում տրվում են անհրաժեշտ սահմանումներ և օգտագործված արդյունքներ: Ուսումնասիրվում են փոփոխականներ չօգտագործող ծրագրերի որոշվածությունը իրենց թույլատրելի հարցումների նկատմամբ: Նկատենք, որ փոփոխականներ չօգտագործող ծրագրերի թույլատրելի հարցումները կարող են օգտագործել փոփոխականներ: Ցույց է տրվում, որ ընդհանուր առմամբ փոփոխականներ չօգտագործող ծրագրերը իրենց թույլատրելի հարցումների նկատմամբ որոշված չեն:

2.2 բաժնում նկարագրվում է ձևափոխության ալգորիթմ, որով ցանկացած փոփոխականներ չօգտագործող ծրագիր փոխակերպվում է մեկ այլ՝ արդեն փոփոխականներով ծրագրի, որն որոշված է սկզբնական ծրագրի թույլատրելի հարցումների նկատմամբ:

Սահմանում 2.2.1 (ծրագրի ձևափոխություն PT1): *Դիցուք P-ն փոփոխականներ չօգտագործող տրամաբանական ծրագիր է: P ծրագրից ստացվում է P' ծրագիրը հետևյալ կերպ.*

- *ամեն մի S՝*

$$p(\overline{\tau}) \leftarrow p_1(\overline{\tau_1}), \ldots, p_m(\overline{\tau_m}) \in P,$$

  $m \geq 0$, *նախադասություն փոխարինվում է S'՝*

$$p^T(\overline{\tau}, s(z)) \leftarrow p_1^T(\overline{\tau_1}, z), \ldots, p_m^T(\overline{\tau_m}, z)$$

  նախադասությամբ,

- *ամեն մի $p \in \Pi_P$ պրեդիկատային սիմվոլի համար P' ծրագրում ավելացվում է*

$$p(x_1, \ldots, x_k) \leftarrow p^T(x_1, \ldots, x_k, s^h(0))$$

  կանոնը,

որտեղ $h = \overline{\overline{P}}$, 0-ն հաստատուն է, z-ը փոփոխական է, $x_1, \ldots, x_k$ փոփոխականները իրարից տարբեր են, s-ը մեկ տեղանի ֆունկցիոնալ սիմվոլ է, և $p^T, p_i^T \notin \Pi_P, (1 \leq i \leq m)$:

2.3 բաժնում ցույց է տրվում, որ սկզբնական և PT1 ձևափոխության արդյունքում ստացված ծրագրերի տրամաբանական (հետևաբար նաև պրոցեդուրային) սեմանտիկաները սկզբնական ծրագրի թույլատրելի հարցումների նկատմամբ համընկնում են:

Թեորեմ 2.3.1 (սեմանտիկայի պահպանում): *Դիցուք P-ն փոփոխականներ չօգտագործող տրամաբանական ծրագիր է, P'-ը P ծրագրից PT1 ձևափոխության արդյունքում ստացված ծրագիրը: Այդ դեպքում, ցանկացած $G \in \Delta(P)$ հարցման համար, $P \models G$ այն և միայն այն դեպքում, երբ $P' \models G$:*

Թեորեմ 2.3.1-ը ապացուցվում է Լեմմա 2.3.1 – 2.3.6-ի միջոցով:

2.4 բաժնում ցույց է տրվում, որ PT1 ձևափոխությունը ծառայում է իր նպատակին: Այսինքն, PT1 ձևափոխության արդյունքում ստացված ցանկացած ծրագիր սկզբնական ծրագրի թույլատրելի հարցումների նկատմամբ որոշված է:

Թեորեմ 2.4.1 (որոշվածության): *Դիցուք P-ն փոփոխականներ չօգտագործող տրամաբանական ծրագիր է, P'-ը P ծրագրից PT1 ձևափոխության արդյունքում ստացված ծրագիրը: Այդ դեպքում, ցանկացած $G \in \Delta(P)$ հարցման համար, (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է:*

Թեորեմ 2.4.1-ը ապացուցվում է հետևյալ ձևով: Քանի որ SLD-ծառերի յուրաքանչյուր հանգույցի զավակների քանակը վերջավոր է, ապա համաձայն Կոնիգի (König) լեմմայի, որպեսզի ցույց տանք (P', G) զույգի SLD-ծառի վերջավոր լինելը բավական է ցույց տալ, որ (P', G) զույգից ցանկացած SLD-արտածում (SLD-ծառի ճյուղ) վերջավոր է: (P', G) զույգից ցանկացած SLD-արտածման վերջավոր լինելն էլ ցույց տալու համար, համաձայն Թեորեմ 1.4.2-ի, բավական է ցույց տալ, որ P' ծրագիրը ռեկուրենտ է, իսկ G հարցումը՝ սահմանափակ ըստ որևէ ատոմի մակարդակի՝ $||$ ֆունկցիայի: A հիմնական ատոմի համար սահմանվում է $||$ ֆունկցիան հետևյալ կերպ՝

- $|A| = depth(\tau_0)$, եթե $A = p^T(\overline{\tau}, \tau_0)$, որտեղ $p^T \in \Pi_{P'}$ և $p^T \notin \Pi_P$,
- $|A| = h + 1$, մնացած դեպքերում, որտեղ $h = \overline{\overline{P}}$:

Ցույց է տրվում, որ ըստ վերը սահմանված $||$ ֆունկցիայի P' ծրագիրը ռեկուրենտ է, իսկ G հարցումը սահմանափակ:

Գլուխ 3-ում ուսումնասիրվում են մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող տրամաբանական ծրագրերը: Բերվում են այս գլխում օգտագործվող սահմանումներն ու արդյունքները: Ցույց է տրվում, որ ընդհանուր առմամբ մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող ծրագրերը իրենց թույլատրելի հարցումների նկատմամբ որոշված չեն: Նկարագրվում է ձևափոխություն, որով ցանկացած մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող P ծրագիր փոխակերպվում է մեկ այլ P' ծրագրի: Ապացուցվում է, որ P ծրագրի ցանկացած թույլատրելի G հարցման համար, $P \models G$ այն և միայն այն դեպքում, երբ $P' \models G$: Ցույց է տրվում, որ (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է: Երրորդ գլուխը բաղկացած է չորս բաժիններից:

3.1 բաժնում տրվում են անհրաժեշտ սահմանումներ և օգտագործված արդյունքներ: Ցույց է տրվում, որ ընդհանուր առմամբ մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող ծրագրերը իրենց թույլատրելի հարցումների նկատմամբ որոշված չեն:

Կասենք, որ A ատոմը *նախորդում* է B ատոմին և կնշանակենք $A \lesssim B$, եթե գոյություն ունի այնպիսի θ տեղադրություն, որ $A\theta = B$: Հեշտ է տեսնել, որ $\lesssim$ հարաբերությունը ռեֆլեքսիվ և տրանզիտիվ է:

Սահմանում 3.1.1 (Կոնգրուենտություն): *Կասենք, որ A և B ատոմները* կոնգրուենտ *են և կնշանակենք $A \equiv B$, եթե $A \lesssim B$ և $B \lesssim A$:*

Հեշտ է տեսնել, որ $\equiv$ հարաբերությունը համարժեքության հարաբերություն է: Կարելի է ցույց տալ, որ նախորդման և հետևաբար նաև կոնգրուենտության հարաբերությունները ալգորիթմորեն լուծելի են: Այս գլխում կոնգրուենտ թերմերը չենք տարբերի:

3.2 բաժնում նկարագրվում է ձևափոխության ալգորիթմ, որով ցանկացած մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող ծրագիր փոխակերպվում է մեկ այլ ծրագրի, որը որոշված է սկզբնական ծրագրի թույլատրելի հարցումների նկատմամբ:

Մինչ ձևափոխության նկարագրությունը ներմուծենք որոշ գաղափարներ: $H(P,S)$-ով նշանակենք P ծրագրի S նախադասության գլխի իրարից տարբեր նմուշների բազմությունը, որոնք օգտագործում են հաստատուններ (զրո տեղանի ֆունկցիոնալ սիմվոլներ) միայն P ծրագրից: Այլ կերպ ասած, $A \in H(P,S)$ այն և միայն այն դեպքում, երբ գոյություն ունի θ տեղադրություն, որն օգտագործում է հաստատուններ միայն P ծրագրից, այնպես որ $B\theta = A$, որտեղ B-ն S նախադասության գլուխն է:

Լեմմա 3.2.1: *Դիցուք P-ն մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող ծրագիր է: Այդ դեպքում, P ծրագրի ցանկացած S նախադասության համար տեղի ունի*

$$\overline{\overline{H(P,S)}} = \begin{cases} \prod_{j=1}^{k}(l+j), & \text{եթե } k>0 \\ 1, & \text{եթե } k=0 \end{cases}$$

որտեղ k-ն S նախադասության գլխի իրարից տարբեր փոփոխականների քանակն է, իսկ l-ը P ծրագրում օգտագործվող իրարից տարբեր հաստատունների քանակը:

$P = \{S_1, S_2, ..., S_n\}$, $n \geq 1$ ծրագրի համար սահմանենք $H(P)$ ատոմների բազմությունը հետևյալ կերպ.

$$H(P) = H(P,S_1) \cup H(P,S_2) \cup \cdots \cup H(P,S_n):$$

Պարզ է, որ

$$\overline{\overline{H(P)}} \leq \sum_{i=1}^{n} \overline{\overline{H(P,S_i)}}:$$

Սահմանում 3.2.1 (ծրագրի ձևափոխություն PT2): *Դիցուք P-ն մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող տրամաբանական ծրագիր է: P ծրագրից ստացվում է P' ծրագիրը հետևյալ կերպ.*

- *ամեն մի S՝*

  $$p(\bar{t}) \leftarrow p_1(\overline{t_1}), \ldots, p_m(\overline{t_m}) \in P,$$

  $m \geq 0$, նախադասություն փոխարինվում է S'՝

  $$p^T(\bar{t}, s(z)) \leftarrow p_1^T(\overline{t_1}, z), \ldots, p_m^T(\overline{t_m}, z)$$

  նախադասությամբ,

- *ամեն մի $p \in \Pi_P$ պրեդիկատային սիմվոլի համար P' ծրագրում ավելացվում է*

  $$p(x_1, \ldots, x_k) \leftarrow p^T(x_1, \ldots, x_k, s^h(0))$$

  կանոնը,

որտեղ $h = \overline{\overline{\overline{H(P)}}}$, 0*-ն հաստատուն է,* z*-ը փոփոխական է,* $z \notin Var(S)$, $x_1, \ldots, x_k$ *փոփոխականները իրարից տարբեր են,* s*-ը մեկ տեղանի ֆունկցիոնալ սիմվոլ է, և* $p^T, p_i^T \notin \Pi_P, (1 \leq i \leq m)$*:*

3.3 բաժնում ցույց է տրվում, որ սկզբնական և PT2 ձևափոխության արդյունքում ստացված ծրագրերի տրամաբանական (հետևաբար նաև պրոցեդուրային) սեմանտիկաները սկզբնական ծրագրի թույլատրելի հարցումների նկատմամբ համընկնում են:

Թեորեմ 3.3.1 (**սեմանտիկայի պահպանման**): *Դիցուք* P*-ն մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող տրամաբանական ծրագիր է,* P'*-ը* P *ծրագրից PT2 ձևափոխության արդյունքում ստացված ծրագիրը: Այդ դեպքում, ցանկացած* $G \in \Delta(P)$ *հարցման համար,* $P \models G$ *այն և միայն այն դեպքում, երբ* $P' \models G$*:*

Թեորեմ 3.3.1-ը ապացուցվում է Լեմմա 3.3.1 – 3.3.6-ի միջոցով:

3.4 բաժնում ցույց է տրվում, որ PT2 ձևափոխությունը ծառայում է իր նպատակին: Այսինքն, PT2 ձևափոխության արդյունքում ստացված ցանկացած ծրագիր սկզբնական ծրագրի թույլատրելի հարցումների նկատմամբ որոշված է:

Թեորեմ 3.4.1 (**որոշվածության**): *Դիցուք* P*-ն մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող տրամաբանական ծրագիր է,* P'*-ը* P *ծրագրից PT2 ձևափոխության արդյունքում ստացված ծրագիրը: Այդ դեպքում, ցանկացած* $G \in \Delta(P)$ *հարցման համար,* (P', G) *զույգի ցանկացած SLD-ծառ վերջավոր է:*

Այս թեորեմը ապացուցվում է նման եղանակով ինչպես Թեորեմ 2.4.1-ը:

Գլուխ 4-ում ուսումնասիրվում են մոնադիկ ծրագրերը և հարցումները: Ներմուծվում են այս գլխում օգտագործվող սահմանումներն ու արդյուքները: Ցույց է տրվում, որ ընդհանուր առմամբ մոնադիկ ծրագրերը իրենց թույլատրելի, փոփոխականներ չօգտագործող մոնադիկ հարցումների նկատմամբ որոշված չեն: Նկարագրվում են ծրագրի և հարցման ձևափոխություններ, որով ցանկացած P մոնադիկ ծրագիր և G թույլատրելի, փոփոխականներ չօգտագործող մոնադիկ հարցում փոխակերպվում է մեկ այլ P' ծրագրի և G' հարցման: Ապացուցվում է, $P \models G$ այն և միայն այն դեպքում, երբ $P' \models G'$: Ցույց է տրվում, որ (P', G') զույգի ցանկացած SLD-ծառ վերջավոր է: Նշենք, որ ձևափոխությամբ ստացված P' ծրագիրը նույնն է բոլոր դիտարկվող հարցումների համար: Չորրորդ գլուխը բաղկացած է հինգ բաժիններից:

4.1 բաժնում տրվում են անհրաժեշտ սահմանումներ և օգտագործված արդյունքներ: Ցույց է տրվում, որ ընդհանուր առմամբ մոնադիկ ծրագրերը իրենց թույլատրելի, փոփոխականներ չօգտագործող մոնադիկ հարցումների նկատմամբ որոշված չեն:

Ծրագիրը (հարցումը) կոչվում է *մոնադիկ*, եթե այն չի օգտագործում ֆունկցիոնալ սիմվոլներ մեկից մեծ տեղայնությամբ, և օգտագործում է միայն մեկ տեղանի պրեդիկատային սիմվոլներ:

4.2 բաժնում սահմանվում են կանոնական մոնադիկ ծրագրերի և հարցումների գաղափարները, և ցույց տրվում, որ ցանկացած մոնադիկ ծրագիր և հարցում կարելի է ձևափոխել կանոնական մոնադիկ ծրագրի և հարցման այնպես, որ սկզբնական հարցումը հետևում է սկզբնական ծրագրից այն և միայն այն դեպքում, երբ ձևափոխված հարցումը հետևում է ձևափոխված ծրագրից:

Սահմանում 4.2.1 (**Կանոնական ծրագիր**): *Մոնադիկ ծրագիրը կանվանենք կանոնական, եթե*

- 0-*ն միակ հաստատունն է, որն օգտագործվում է այդ ծրագրում,*
- *այդ ծրագրի կանոնները չեն օգտագործում հիմնական թերմեր,*
- *այդ ծրագրի ցանկացած կանոն օգտագործում է ճիշտ մեկ փոփոխական: Հարմարության համար ենթադրենք, որ ծրագրի բոլոր կանոններում օգտագործվում է միևնույն փոփոխականը, և նշանակենք այն* x*-ով,*
- *այդ ծրագրի կանոններում օգտագործվող ցանկացած թերմ* x *փոփոխականն է կամ* $f(x)$ *թերմը, որտեղ* f*-ը ինչ որ ֆունկցիոնալ սիմվոլ է,*
- *այդ ծրագրի փաստերում օգտագործվող ցանկացած թերմ* 0 *հաստատունն է կամ* x *փոփոխականը:*

Սահմանում 4.2.2 (**Կանոնական հարցում**): *Մոնադիկ հարցումը կանվանենք կանոնական, եթե* 0-*ն միակ հաստատունն է, որն օգտագործվում է այդ հարցման մեջ:*

Նկարագրվում են ձևափոխություններ, որոնցով ցանկացած մոնադիկ ծրագիր և հարցում ձևափոխվում է կանոնական մոնադիկ ծրագրի և կանոնական հարցման: Ապացուցվում է հաջորդիվ թեորեմը:

Թեորեմ 4.2.1: *Ցանկացած* P *մոնադիկ ծրագիր և* $G \in \Delta(P)$ *մոնադիկ հարցում կարելի է ձևափոխել* $\tilde{P}$ *կանոնական մոնադիկ ծրագրի և* $\tilde{G}$ *կանոնական մոնադիկ հարցման, այնպես որ* $P \models G$ *այն և միայն այն դեպքում, երբ* $\tilde{P} \models \tilde{G}$*:*

Հաշվի առնելով Թեորեմ 4.2.1-ը, ընդհանրությունը չենք խախտի, եթե հետագա բաժիններում դիտարկենք միայն կանոնական մոնադիկ ծրագրերը և այդ ծրագրերի թույլատրելի կանոնական մոնադիկ հարցումները: Այսուհետ ասելով մոնադիկ ծրագիր կհասկանանք կանոնական մոնադիկ ծրագիր, և ասելով մոնադիկ հարցում կհասկանանք կանոնական մոնադիկ հարցում:

4.3 բաժնում նկարագրվում են ծրագրի և հարցման ձևափոխություններ, որով ցանկացած մոնադիկ (կանոնական) ծրագիր և փոփոխականներ չօգտագործող մոնադիկ (կանոնական) հարցում փոխակերպվում է մեկ այլ ծրագրի և հարցման:

Մինչ ձևափոխության նկարագրությունը ներմուծենք որոշ գաղափարներ: P մոնադիկ ծրագրի և G՝ $\leftarrow p_1(\tau_1), \ldots, p_k(\tau_k)$ փոփոխականներ չօգտագործող մոնադիկ հարցման համար սահմանենք $constraint(P, G)$ հետևյալ կերպ՝

$$constraint\,(P, G) = \overline{\overline{P}} * \sum_{i=0}^{d-1} (\overline{\overline{\Phi_{P \cup \{G\}}}})^i,$$

որտեղ $d = l + 2^{\overline{\overline{\Pi_P}}}$, $l = max\{depth(\tau_i) \mid i = 1, \ldots, k\}$:

Սահմանում 4.3.1 (ծրագրի ձևափոխություն PT3): *Դիցուք P-ն մոնադիկ տրամաբանական ծրագիր է: P ծրագրից ստացվում է P' ծրագիրը հետևյալ կերպ. ամեն մի S՝*

$$p(t) \leftarrow p_1(t_1), \ldots, p_m(t_m) \in P,$$

$m \geq 0$, նախադասություն փոխարինվում է S'՝

$$p^T(t, s(z)) \leftarrow p_1^T(t_1, z), \ldots, p_m^T(t_m, z)$$

նախադասությամբ, որտեղ z-ը փոփոխական է, $z \notin Var(S)$, s-ը մեկ տեղանի ֆունկցիոնալ սիմվոլ է, $s \notin \Phi_P$, և $p^T, p_i^T \notin \Pi_P, (1 \leq i \leq m)$:

Սահմանում 4.3.2 (հարցման ձևափոխություն GT): *P մոնադիկ ծրագրի և G՝*

$$\leftarrow p_1(\tau_1), \ldots, p_k(\tau_k),$$

փոփոխականներ չօգտագործող հարցում փոխարինվում է G'՝

$$\leftarrow p_1^T(\tau_1, s^h(0)), \ldots, p_k^T(\tau_k, s^h(0)),$$

հարցումով, որտեղ $h = constraint(P, G)$:

4.4 բաժնում ցույց է տրվում, որ սկզբնական ծրագրի և սկզբնական հարցման տրամաբանական (հետևաբար նաև պրոցեդուրային) սեմանտիկան համընկնում է PT3 ձևափոխության արդյունքում ստացված ծրագրի և GT ձևափոխության արդյունքում ստացված հարցման տրամաբանական սեմանտիկայի հետ:

Թեորեմ 4.4.1 (սեմանտիկայի պահպանման): *Դիցուք P-ն մոնադիկ տրամաբանական ծրագիր է, $G \in \Delta(P)$ փոփոխականներ չօգտագործող մոնադիկ հարցում է, P'-ը P ծրագրից PT3 ձևափոխության արդյունքում ստացված ծրագիրն է, իսկ G'-ը G հարցումից GT ձևափոխության արդյունքում ստացված հարցումը: Այդ դեպքում, $P \models G$ այն և միայն այն դեպքում, երբ $P' \models G'$:*

Թեորեմ 3.3.1-ը ապացուցվում է Լեմմա 4.4.1 – 4.4.7-ի միջոցով:

4.5 բաժնում ցույց է տրվում, որ PT3 և GT ծրագրի և հարցման ձևափոխությունները ծառայում են իրենց նպատակին: Այսինքն, PT3 ձևափոխության արդյունքում ստացված ցանկացած ծրագիր որոշված է GT ձևափոխության արդյունքում ստացված հարցման նկատմամբ:

Թեորեմ 4.5.1 (որոշվածության): *Դիցուք P-ն մոնադիկ տրամաբանական ծրագիր է, $G \in \Delta(P)$ փոփոխականներ չօգտագործող մոնադիկ հարցում է, P'-ը P ծրագրից PT3 ձևափոխության արդյունքում ստացված ծրագիրը, իսկ G'-ը G հարցումից GT ձևափոխության արդյունքում ստացված հարցումը: Այդ դեպքում, (P', G') զույգի ցանկացած SLD-ծառ վերջավոր է:*

Այս թեորեմը ապացուցվում է նման եղանակով ինչպես Թեորեմ 2.4.1-ը:

ԱՇԽԱՏԱՆՔԻ ՀԻՄՆԱԿԱՆ ԱՐԴՅՈՒՆՔՆԵՐԸ

Ատենախոսության շրջանակներում կատարված հետազոտությունները բերել են հետևյալ արդյունքների:

1. Նկարագրվել է ծրագրի ձևափոխություն, որով ցանկացած փոփոխականներ չօգտագործող P ծրագիր փոխակերպվում է մեկ այլ՝ P' ծրագրի, այնպես որ P ծրագրի ցանկացած թույլատրելի G հարցման համար, G հարցումը տրամաբանորեն հետևում է P ծրագրից այն և միայն այն դեպքում, երբ այն տրամաբանորեն հետևում է P' ծրագրից, և ցույց է տրվում, որ (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է [1]:

2. Նկարագրվել է ծրագրի ձևափոխություն, որով ցանկացած մեկ և ավել տեղանի ֆունկցիոնալ սիմվոլներ չօգտագործող P ծրագիր փոխակերպվում է մեկ այլ՝ P' ծրագրի, այնպես որ P ծրագրի ցանկացած թույլատրելի G հարցման համար, G հարցումը տրամաբանորեն հետևում է P ծրագրից այն և միայն այն դեպքում, երբ այն տրամաբանորեն հետևում է P' ծրագրից, և ցույց է տրվում, որ (P', G) զույգի ցանկացած SLD-ծառ վերջավոր է [2,3]:

3. Նկարագրվել են ծրագրի և հարցման ձևափոխություններ, որով ցանկացած P մոնադիկ ծրագիր և G թույլատրելի, փոփոխականներ չօգտագործող մոնադիկ հարցում փոխակերպվում է P' ծրագրի և G' հարցման, այնպես որ G հարցումը տրամաբանորեն հետևում է P ծրագրից այն և միայն այն դեպքում, երբ G' հարցումը տրամաբանորեն հետևում է P' ծրագրից, և ցույց է տրվում, որ (P', G') զույգի ցանկացած SLD-ծառ վերջավոր է: Նշենք, որ ձևափոխությամբ ստացված P' ծրագիրը նույնն է բոլոր դիտարկվող հարցումների համար [4]:

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ՇՐՋԱՆԱԿՆԵՐՈՒՄ ՀՐԱՏԱՐԱԿՎԱԾ ԱՇԽԱՏԱՆՔՆԵՐԻ ՑԱՆԿԸ

1. Khachatryan S. A. On the Optimization of Variable-Free Logic Programs. In *Proceedings of CSIT'2011 Conference*, pages 50–51. Gitutyun, 2011.

2. Khachatryan S. A. On the Optimization of Functional Symbol-Free Logic Programs. In *Proceedings of CSIT'2013 Conference*, pages 27–29. Gitutyun, 2013.

3. Khachatryan S. A. On Termination of Functional Symbol-Free Logic Programs. *Proceedings of the Yerevan State University. Physical and Mathematical Sciences*, 2013(3):49–56, 2013.

4. Khachatryan S. A. On Optimization of Monadic Logic Programs. *Proceedings of the Yerevan State University. Physical and Mathematical Sciences*, 2014(1):40–47, 2014.

РЕЗЮМЕ

Хачатрян Сурен Арташесович

“Об оптимизации логических программ”

В данной работе исследуются проблемы логического программирования. Предметом исследования являются логические программы и запросы, а рассматриваемая основная проблема – оптимизация логических программ в том смысле, что любое SLD-дерево программы и любого допустимого запроса этой программы становится конечным. Другими словами, исследуется проблема оптимизации логических программ с помощью которой программа становится завершаемой по отношению к ее допустимым запросам.

Логическое программирование организовано вокруг языка Пролог, созданного А. Кольмероэ на основе процедурной интерпретации предложений Хорна, предложенной Р. Ковальским. По сути, логические программы – это базы знаний. Делается запрос к системе логического программирования, на который она должна ответить на основе программы. Система логического программирования должна ответить, является ли запрос логическим следствием программы или нет, и, если да, то при каких значениях переменных, использованных в запросе. В системах логического программирования в качестве правила вывода в основном используется SLD-резолюция (Selective Linear Definite clause resolution). Но для некоторых программ и запросов системы логического программирования, основанные на SLD-резолюции, могут работать бесконечно. Возникает проблема, оптимизации программы таким образом, чтобы любое SLD-дерево программы и любого допустимого запроса этой программы стало конечным.

В общем случае, проблема оптимизации неразрешима. Таким образом, представляется целесообразным изучить специальные случаи логических программ.

В рамках данной работы проблема оптимизации рассматривается для следующих видов программ: логических программ, не использующих переменные, логических программ не использующих функциональные символы местности > 1 и монадических программ (программы, которые не используют функциональные символы местности > 1, и используют только предитакные символы местности 1).

М. Безем ввел понятия так называемых рекурентных программ и ограниченных запросов и показал, что рекурентные программы завершаемы по отношению к ограниченным запросам.

В работе представлены трансформации, при которых программы вышеуказанного типа и их допустимые запросы преобразовываются в рекуррентные программы и ограниченные запросы с использованием методов ограниченного логического программирования (constraint logic programming). В результате, решается проблема оптимизации для данных видов программ. Запрос допустим для программы, если

он использует только предикатные символы, которые используются в этой программе. Следует отметить, что рассматривая только допустимые запросы, мы не нарушаем общности, так как если запрос не является допустимым для программы, то, очевидно, он не является логическим следствием этой программы.

Основной целью диссертационной работы является:

1. Исследовать проблему оптимизации для логических программ, не использующих переменные, и их допустимых запросов.

2. Исследовать проблему оптимизации для логических программ, не использующих функциональные символы местности > 1, и их допустимых запросов.

3. Исследовать проблему оптимизации для монадических логических программ и их допустимых запросов.

В результате исследований, проведенных в данной работе, были получены следующие основные результаты:

1. Описывается трансформация программы, при которой любая программа P, не использующая переменные, преобразуется в программу P', так что для любого допустимого запроса G программы P, запрос G является логическим следствием программы P тогда и только тогда, когда G является логическим следствием программы P', а также любое SLD-дерево программы P' и запроса G - конечно.

2. Описывается трансформация программы, при которой любая программа P, не использующая функциональные символы местности > 1, преобразуется в программу P', так что для любого допустимого запроса G программы P, запрос G является логическим следствием программы P тогда и только тогда, когда G является логическим следствием программы P', а также любое SLD-дерево программы P' и запроса G - конечно.

3. Описываются трансформации программы и запроса, при которых любая монадическая программа P и запрос G, не содержащий переменные, преобразуются в программу P' и запрос G', так что запрос G является логическим следствием программы P тогда и только тогда, когда запрос G' является логическим следствием программы P', а также любое SLD-дерево программы P' и запроса G' - конечно. Следует отметить, что программа P', полученная в результате трансформации, одна и та же для всех рассматриваемых запросов.

ABSTRACT

Suren A. Khachatryan

“On Optimization of Logic Programs”

The thesis is devoted to the study of some problems of logic programming. The objects of study are logic programs and goals. The main problem studied is the optimization of logic programs in the sense that any SLD-tree of a program and any permitted goal of this program become finite. In other words, the optimization problem is studied by which a program becomes terminating with respect to its permitted goals.

Logic programming is organised around Prolog language, which was created by A. Colmerauer, based on R. Kowalski’s procedural interpretation of Horn clauses. Logic programs are in essence knowledge bases. The logic programming system is queried, to which it should answer based on the knowledge in the program. The logic programming system should answer whether a goal is a logical consequence of a logic program or not and, if yes, for which values of the variables used in the goal it happens. In a logic programming system SLD-resolution (Selective Linear Definite clause resolution) is basically used as an inference rule. But for some programs and goals logic programming systems based on SLD-resolution can work infinitely. A problem arises, namely to optimize programs such that any SLD-tree of a program and any permitted goal of this program will become finite.

In general, the optimization problem is unsolvable. Thus, it seems appropriate to study specific cases of logic programs.

In the framework of this thesis the optimization problem is considered for the following three program classes: variable-free logic programs (programs which do not use variables), functional symbol-free logic programs (programs which do not use functional symbols of arity > 1), and monadic programs (programs which do not use functional symbols of arity > 1 and use only predicate symbols of arity 1).

M. Bezem introduced the concepts of the so-called recurrent programs and bounded goals. It is shown that recurrent programs are terminating with respect to bounded goals.

In the thesis transformations are introduced by which programs of the above-mentioned classes and their permitted goals are transformed into recurrent programs and bounded goals by the use of constraint logic programming methods. As a result, the optimization problem is resolved for these classes. For a program, a goal is permitted if it uses only predicate symbols which are used in that program. It is to be noted that considering only permitted goals does not break the generality because if a goal is not permitted for a program it obviously is not a logical consequence of that program.

The main objectives of the research are:

1. To research the optimization problem for variable-free logic programs and their permitted goals.

2. To research the optimization problem for functional symbol-free logic programs and their permitted goals.

3. To research the optimization problem for monadic logic programs and their permitted, variable-free monadic goals.

The research conducted within the framework of the thesis has produced the following main results:

1. Program transformation is introduced by which any variable-free program P is transformed into P' program, such that for any permitted goal G of P, the goal G is a logical consequence of the program P if and only if it is a logical consequence of the program P' and any SLD-tree of P' and G is finite.

2. Program transformation is introduced by which any functional symbol-free program P is transformed into P' program, such that for any permitted goal G of P, the goal G is a logical consequence of the program P if and only if it is a logical consequence of the program P' and any SLD-tree of P' and G is finite.

3. Program and goal transformations are introduced by which any monadic program P and any variable-free monadic goal G are transformed into P' and G', such that the goal G is a logical consequence of the program P if and only if the goal G' is a logical consequence of the program P' and any SLD-tree of P' and G' is finite. It is to be noted that transformed program P' is the same for all considered goals.

Ծավալը՝ 24 էջ: Տպաքանակը՝ 100:
ՀՀ ԳԱԱ ԻԱՊԻ կոմպյուտերային պոլիգրաֆիայի լաբորատորիա:
Երևան, Պ. Սևակի 1