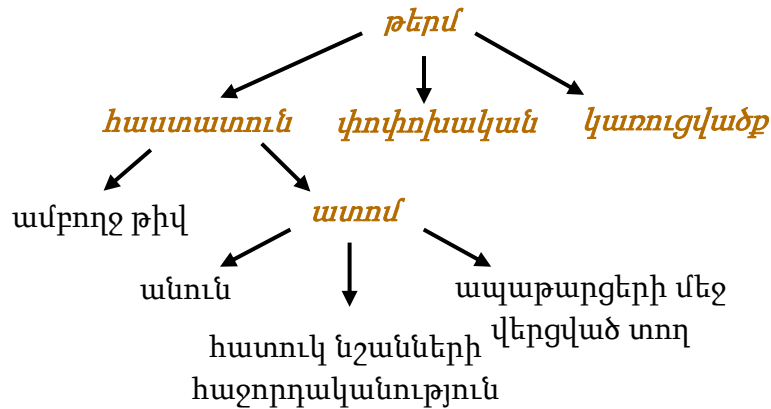


2. Շարահյուսություն

Prolog ծրագրերը կառուցվում են թերմերից: **Թերմը** սահմանվում է հետևյալ կերպ.



Նկ. 2.1

2.1. Հաստատուններ, փոփոխականներ, կառուցվածքներ

2.1.1. Հաստատուններ

Գոյություն ունեն երկու տեսակի հաստատուններ՝ *ամբողջ թվեր* և *ատոմներ*:

Prolog լեզվում **ամբողջ թվեր են** համարվում տասական համակարգում ներկայացված և մեքենայական բառով սահմանափակված ամբողջ թվերը, որոնց նկատմամբ կիրառվում են թվաբանական և համեմատության հիմնական գործողությունները: Դրա հետ մեկտեղ Prolog-ը հնարավորություն է տալիս մոդելավորելու աշխատանքը մեքենայական բառով չսահմանափակված ամբողջ թվերի հետ թվաբանական և համեմատության գործողությունները իրականացնող ծրագրերի ստեղծման միջոցով:

Ատոմներ են՝

- տառերից, թվանշաններից և ընդգծման նշանից կառուցված, փոքրատառով սկսվող հաջորդականությունները: Օրինակ՝

alpha, aLIST01, a_LIST_01:

- Հատուկ նշաններից կառուցված հաջորդականությունները: Օրինակ՝

?-, :-, +, ->:

- Ապաթարցերի մեջ վերցված կամայական նշանների հաջորդականությունները: Օրինակ՝

'123', 'Alpha', '1+2' և այլն:

2.1.2. Փոփոխականներ

Փոփոխականը տառերից, թվանշաններից և ընդգծման նշանից կառուցված, մեծատառով կամ ընդգծման նշանով սկսվող հաջորդականություն է: Օրինակ՝

Alpha, WWW, _a_PLUS_b:

Միայն ընդգծման նշանից բաղկացած փոփոխականը կոչվում է *անանուն փոփոխական*: Ի տարբերություն սովորական փոփոխականի՝ անանուն փոփոխականի արժեքը չի կոնկրետանում: Օրինակ, եթե անհրաժեշտ է պարզել՝ հավանում է արդյոք որևէ մեկը **john**-ին, բավարար է գրել.

?- likes(_, john).

Համակարգը այս հարցմանը կտա **yes** կամ **no** պատասխան: Ի տարբերություն դրան, եթե հարցումը ձևակերպվի

?- likes(X, john).

տեսքով, ապա համակարգը հաջորդաբար կարտահանի **X**-ի բոլոր հնարավոր արժեքները:

Եթե անանուն փոփոխականը հարցման մեջ հանդիպում է մեկ անգամից ավելի, ապա յուրաքանչյուր հանդիպում դիտարկվում է որպես առանձին փոփոխական: Օրինակ՝

?- likes(_, john), likes(_, mary).

հարցման պատասխանը կլինի **yes** այն և միայն այն դեպքում, երբ կա երկու օբյեկտ, որոնցից մեկը հավանում է **john**-ին, իսկ մյուսը՝ **mary**-ին: Ի տարբերություն դրան, եթե հարցումը ձևակերպվի հետևյալ տեսքով՝

?- likes(X, john), likes(X, mary).

ապա պատասխանում կարտաբերվեն բոլոր այն օբյեկտները, որոնք միաժամանակ հավանում են և՛ **john**-ին, և՛ **mary**-ին:

2.1.3. Կառուցվածքներ

Կառուցվածքը առանձին բաղադրիչներից բաղկացած միասնական և անվանակոչված օբյեկտ է: Բաղադրիչների դերում կարող են հանդես գալ հաստատուններ, փոփոխականներ և այլ կառուցվածքներ: Կառուցվածքը հնարավորություն է տալիս հստակեցնելու օբյեկտի իմաստը՝ խմբավորելով տվյալները այս կամ այն եղանակով:

Օրինակ՝

has(john, programming_in_prolog).

փաստից պարզ չէ, թե **john**-ի ունեցած օբյեկտը զիրք է, այլ ոչ թե, օրինակ, ամսագիր: Ներմուծելով **book** անունով կառուցվածք, որն ունի գրքի հեղինակին և վերնագիրը նշող բաղադրիչներ, այս փաստը կարող ենք ներկայացնել ավելի հասկանալի տեսքով.

has(john, book(clocks_in_melish, programming_in_prolog)).

book կառուցվածքի բաղադրիչների իմաստն ավելի թափանցիկ դարձնելու համար կարող ենք գրել.

has(john, book(author(clocks_in_melish), title(programming_in_prolog))).

Փոփոխականներ պարունակող կառուցվածքները կարող են հանդիպել հարցումներում: Օրինակ՝

?-has(john, book(author(clocks_in_melish), X)).

?-has(john, book(author(clocks_in_melish), _)).

?-has(X, book(author(clocks_in_melish), Y)).

Ընդհանուր դեպքում կառուցվածքը սահմանվում է հետևյալ կերպ.

ֆունկտոր (բաղադրիչների ցուցակ),

որտեղ **ֆունկտոր**-ը *ատում* է, իսկ **բաղադրիչները**՝ կամայական *թերմեր*: Բաղադրիչների քանակը կոչվում է կառուցվածքի տեղայնություն (*arity*): Ատումը դիտարկվում է որպես **0** տեղանի կառուցվածք:

Նշենք, որ փաստը շարահյուսորեն կառուցվածք է, իսկ կանոնն ու հարցումը՝ հատուկ նշաններից և կառուցվածքներից բաղկացած հաջորդականություններ:

2.2. Թվաբանություն

2.2.1. Օպերատորներ

Prolog լեզվում ***թվաբանական արտահայտությունը*** սահմանվում է որպես *թիվ*, *փոփոխական* կամ այնպիսի *կառուցվածք*, որի ֆունկտորը գործողության նշան է, իսկ բաղադրիչները՝ թվաբանական արտահայտություններ: Միջաձանցային ձևով գրառված թվաբանական արտահայտությունը կոչվում է ***օպերատորային գրառում***: Օրինակ՝ **X + Y * Z** օպերատորային գրառմանը համապատասխանում է **+(X, *(Y, Z))** թվաբանական արտահայտությունը: Օպերատորային գրառման հիման վրա կառուցվածքը միարժեքորեն վերականգնելու համար անհրաժեշտ է օգտագործել գործողությունների հետևյալ բնութագրիչները.

- ներկայացում (նախաձանցային, վերջաձանցային կամ միջաձանցային),

- առաջնայնություն,
- ասոցիատիվություն (ձախ կամ աջ):

Այսպես, օրինակ, $a + b / c$ օպերատորային գրառումը վերծանվում է որպես $+(a, (b, c))$ թվաբանական արտահայտություն՝ գումարման համեմատ բաժանման գործողության ավելի բարձր առաջնայնությունից ելնելով: Իսկ $8/2/2$ օպերատորային գրառումը՝ որպես $/((8, 2), 2)$ թվաբանական արտահայտություն՝ բաժանման գործողության ձախ ասոցիատիվությունից ելնելով:

Թվաբանական արտահայտության (օպերատորային գրառման) արժեքը հաշվվում է միայն այն դեպքում, երբ այն հանդիպում է **is** նախասահմանված բինար պրեդիկատի երկրորդ արգումենտի տեղում: Այնպես որ միմյանցից տարբեր են համարվում օպերատորային հետևյալ գրառումները՝ **3+4**, **4+3**, **7**:

Թվաբանական գործողություններ կատարելու համար **Prolog**-ը տրամադրում է հետևյալ նախասահմանված օպերատորները.

+	գումարում,
-	հանում,
*	բազմապատկում
//	ամբողջաթիվ բաժանում,
mod	մնացորդի որոշում,
/	իրական թվերի բաժանում:

2.2.2. Պրեդիկատներ

Ամբողջ թվերի համեմատության համար Prolog-ում սահմանված են հետևյալ պրեդիկատները.

$X < Y$	X -ը փոքր է Y -ից,
$X > Y$	X -ը մեծ է Y -ից,
$X \leq Y$	X -ը փոքր է կամ հավասար Y -ին,
$X \geq Y$	X -ը մեծ է կամ հավասար Y -ին:

Չի թույլատրվում այս պրեդիկատները օգտագործել փաստեր կառուցելու համար: Այսպես, օրինակ, $2 > 3$ փաստը շարահյուսորեն ճիշտ է գրառված, սակայն Prolog համակարգը թույլ չի տա ավելացնել այն տվյալների բազային՝ տվյալների ամբողջականությունը չխախտելու համար:

Դիտարկենք հետևյալ օրինակը:

Դիցուք **governed(X, Y, Z)**-ը պնդում է, ըստ որի՝ **X** անձը ղեկավարել է տվյալ երկիրը՝ սկսած **Y** թվականից մինչև **Z** թվականը: Նշանակենք **president(X, Y)** պնդումը, ըստ որի՝ **X**-ը երկրի նախագահ է եղել **Y** թվականին: **president** պրեդիկատը կարող ենք արտահայտել **governed** պրեդիկատի միջոցով հետևյալ կերպ.

president(X, Y):- governed(X, A, B), Y >= A, Y <= B.

Prolog-ում սահմանված է միջաձանցային տեսքով գրավող = նույնականացման օպերատորը, որն ունի հետևյալ իմաստը: Դիցուք **X**-ը և **Y**-ը կամայական թերմեր են, որոնք կարող են պարունակել նաև չկոնկրետացված փոփոխականներ: **X = Y** պնդման մշակումը կատարվում է հետևյալ կանոնների համաձայն.

- **X = Y** պնդումը համարժեք է **Y = X** պնդմանը,
- եթե **X**-ը և **Y**-ը հաստատուններ են կամ հաստատուններով կոնկրետացված փոփոխականներ, ապա **X = Y** պնդումը բավարարվում է այն և միայն այն դեպքում, երբ **X**-ին և **Y**-ին համապատասխանող հաստատունները համընկնում են,
- եթե **X**-ը չկոնկրետացված փոփոխական է, ապա **X = Y** պնդումը նույնաբար բավարարվում է, և այն ապացուցելու արդյունքում **X**-ը նույնանում է **Y**-ի հետ: Օրինակ՝

?- X= book(clocks_in_melish, programming_in_prolog).

պնդումը բավարարվում է, ինչը հավաստելու արդյունքում **X**-ը կոնկրետանում է՝ ընդունելով **book(clocks_in_melish, programming_in_prolog)** արժեքը,

- եթե **X**-ը և **Y**-ը կառուցվածքներ կամ կառուցվածքներով կոնկրետացված փոփոխականներ են, ապա **X = Y** պնդումը բավարարվում է այն և միայն այն դեպքում, երբ.
 - համընկնում են **X**-ին և **Y**-ին համապատասխանող կառուցվածքների ֆունկտորները և արգումենտների քանակները,
 - բավարարվում են համապատասխան արգումենտների հավասարության մասին պնդումները:

Օրինակ՝ հետևյալ պնդումը

a(b, C, d(e, F, g(h, i, J))) = a(B, c, d(E, f, g(H, i, j)))

բավարարվում է, ինչը հավաստելու արդյունքում **B, C, E, F, H, J** փոփոխականները կոնկրետանում են՝ համապատասխանաբար ընդունելով **b, c, e, f, h, j** արժեքները:

Prolog լեզվում սահմանվում է նաև '**=**' պրեդիկատը, ընդ որում՝ **X \= Y** պնդումը բավարարվում է այն և միայն այն դեպքում, երբ չի բավարարվում **X = Y** պնդումը: Հարկ է նշել, որ, ի տարբերություն **X = Y** պնդմանը, **X \= Y** պնդումն ապացուցելիս փոփոխականների կոնկրետացում տեղի չի ունենում:

Հաշվումներ իրականացնելու համար Prolog լեզվում օգտագործվում է միջաձանցային ձևով գրառվող **is** նախասահմանված բինար պրեդիկատը՝

X is Y,

որտեղ **Y**-ը թվաբանական արտահայտություն է: Պահանջվում է, որ **X is Y** պնդումն ապացուցելու պահին **Y**-ի բոլոր փոփոխականները կոնկրետացված լինեն և ընդունեն թվային արժեքներ:

X is Y պնդումն ապացուցվում է հետևյալ եղանակով.

- հաշվվում է **Y** արտահայտության արժեքը, դիցուք՝ **A**,
- եթե **X**-ը չկոնկրետացված փոփոխական է, ապա **X is Y** պնդումը հավաստի է համարվում, ինչն ապացուցելու արդյունքում **X**-ը կոնկրետանում է՝ ընդունելով **A** արժեքը,
- հակառակ դեպքում **X is Y** պնդումը բավարարվում է այն և միայն այն դեպքում, երբ **X**-ը **A** հաստատուն կամ **A** հաստատունով կոնկրետացված փոփոխական է:

Դիտարկենք հետևյալ օրինակը:

Դիցուք **population(X, Y)**-ը պնդում է, ըստ որի՝ **X** երկրի բնակչությունը կազմում է **Y** մարդ, իսկ **area(X, Y)**-ը՝ պնդում, ըստ որի՝ **X** երկրի տարածքը հավասար է **Y** քառակուսի կիլոմետրի: Նշանակենք **density(X, Y)** պնդումը, ըստ որի՝ **X** երկրի բնակչության խտությունը կազմում է **Y** մարդ մեկ քառակուսի կիլոմետրի վրա: **density** պրեդիկատը կարող ենք արտահայտել **population** և **area** պրեդիկատների միջոցով հետևյալ կերպ.

density(X, Y):- population(X, P), area(X, A), Y is P/A.

Վարժություններ

1. Ստորև բերված օբյեկտներից նշել ճիշտ կազմված լեքսիկական միավորները ու դրանց տեսակը՝ *ատոմ*, *թիվ*, *փոփոխական* կամ *կառուցվածք*.

- a) **Diana**
- b) **diana**
- c) **'Diana'**
- d) **_diana**
- e) **'Diana goes south'**
- f) **goes(diana, south)**
- g) **45**
- h) **5(X, Y)**
- i) **-(north, west)**
- j) **three(Black(Cats)).**

2. Տրված է հետևյալ ծրագիրը.

f(1, one).

f(s(1), two).

f(s(s(1)), three).

f(s(s(s(X))), N) :- f(X, N).

Ցույց տալ ստորև բերված հարցումների մշակման ընթացքը.

a) ?- f(s(1), A).

b) ?- f(s(s(1)), two).

c) ?- f(s(s(s(s(s(s(1)))))), C).

d) ?- f(D, three).