

Б. К. КАРАПЕТЯН

ОБ ОДНОЙ УНИФИКАЦИИ ПРЕДСТАВЛЕНИЯ ШАХМАТНЫХ ПОНЯТИЙ

1. ВВЕДЕНИЕ

Эффективное решение комбинаторных проблем управления связано с использованием различных типов знаний, позволяющих ограничить область поиска решений, определить множество действий, уместных в конкретной ситуации, учесть опыт решения аналогичных задач и т. д. Все указанные аспекты использования знаний характеризуются применением специальных конструкций для задания классов ситуаций с общими свойствами. В [1] рассмотрена возможность описания классов ситуаций посредством формул языка первого порядка. Выделен класс специальных алгоритмов-дескрипторов, вычисляющих формулы языка и предназначенных для распознавания множеств объектов универсума одинаковой полезности, их взаимных отношений, в частности, возможности преобразования одних множеств в другие единообразными средствами. Исследуемые свойства дескрипторов иллюстрируются на примере шахматной задачи, для которой построен универсум, позволяющий получить интерпретацию формул языка первого порядка, адекватную исследуемым вопросам.

В рамках указанного подхода в [2] описана реализация фонда дескрипторов для ЭВМ «Минск-32». Дескрипторы, составляющие фонд, охватывают широкий круг шахматных понятий, для каждого из которых построено соответствующее формульное описание и составлена фортран-программа, вычисляющая истинность формулы на заданной позиции. Анализ этой реализации выявил ряд недостатков, обусловленных как непригодностью языка программирования, так и слабой структурной организацией фонда дескрипторов. Эти недостатки (подробно перечисленные в [3]), ограничивают возможность расширения фонда, усложняют модификацию программ, их сопровождение и перевод на ЭВМ серии ЕС.

Разработка языка описания стратегий (*SDL*) [4], в свою очередь, потребовала создания средств вычисления дескрипторов. В языке *SDL* дескрипторы используются для целей описания спецификаций рассматриваемых стратегий, задания исходных и целевых классов позиций в закономерностях, описания условных выражений управляющих операторов языка *SDL* и др. В настоящей работе описывается программная

компонента пакета АКА-У2, предназначенная для вычисления в текущей позиции дескрипторов формул определенного вида.

При разработке описываемой компоненты (названной DESC—VALUE) основное внимание уделено обеспечению простоты модификации и пополнения фонда дескрипторов без существенных программных изменений. С этой целью:

1) Определен конкретный класс формул, допустимых для описания дескрипторов, а именно формулы языка первого порядка с ограниченными кванторами в предваренной форме.

2) Обеспечена возможность обработки формул без дополнительного кодирования.

3) Разработаны структуры данных для основных элементов универсума, обеспечивающие систематизацию используемых объектов (клетки, позиции и т. д.).

4) Выделено множество первичных шахматных понятий, реализованных функциями, с помощью которых описываются более сложные дескрипторы.

5) Определены объекты специального типа — «доски», позволяющие унифицировать процедуры вычисления атомных формул и основных функций.

6) Разработаны средства диагностики синтаксической корректности формул.

2. ФОРМУЛЬНОЕ ПРЕДСТАВЛЕНИЕ ШАХМАТНЫХ ПОНЯТИЙ

1. Шахматные понятия формализуются нами в ограниченном языке первого порядка [5]. Дескриптор формулы некоторого понятия определяет последовательность действий, необходимых для вычисления значения этой формулы в произвольной позиции, т. е., по существу, каждый дескриптор задает конкретизацию формулы в некоторой структуре.

Ниже приводится тип допустимых формул, посредством которых задаются рассматриваемые нами шахматные понятия, и специальная структура для их интерпретации. Описываемая далее программная компонента DESC—VALUE выполняет функцию компилятора для допустимых формул, а именно, обеспечивает конкретизацию формулы в рассматриваемой структуре для заданной позиции и переводит формульное описание понятия на язык, доступный вычислительной машине.

2. Пусть L язык первого порядка [5].

Термом языка L называется выражение, удовлетворяющее следующим условиям:

(i) всякая переменная или константа есть терм;

(ii) если f — n — местный функциональный символ языка L , а $t_1, t_2, \dots, t_n$ — термы, то $f(t_1, t_2, \dots, t_n)$ — также терм.

Атомной формулой называется выражение вида $p(t_1, t_2, \dots, t_n)$, где p — n — местный предикатный символ языка L , а $t_1, \dots, t_n$ — термы. Правильно построенной формулой языка L (ППФ) называется выражение, удовлетворяющее следующим условиям:

(i) всякая атомная формула есть ППФ;
 (ii) если A и B — ППФ, то $\neg(A)$, $(A \vee B)$, $(A \& B)$ и $(A \rightarrow B)$ — также ППФ;

(iii) если A — ППФ, то $\forall x(A)$ и $\exists x(A)$ также ППФ.
 ППФ находится в пренексной форме, если она:

(i) либо не содержит кванторов,

(ii) либо имеет вид $Qx_1 \dots Qx_n A$, где каждое Qx_i есть либо $\exists x_i$, либо $\forall x_i$, $x_1, \dots, x_n$ — различны, а A — не содержит кванторов; в этом случае $Qx_1 \dots Qx_n$ называется приставкой, а A — матрицей. Можно показать, что для любой ППФ можно построить эквивалентную ей формулу в пренексной форме.

Для формул вида $\forall x(A \rightarrow B)$ либо $\exists x(A \& B)$, где A — формула с единственной свободной переменной x , введем специальные обозначения $\forall x A(x)(B)$ и $\exists x A(x)(B)$. Такие формулы будем называть формулами с ограниченными кванторами.

Для описания шахматных понятий мы используем формулы в пренексной форме с ограниченными кванторами.

3. Опишем конкретную структуру A для языка L , в которой интерпретируются допустимые формулы. В универсум структуры A , кроме индивидов, перечисленных в [1] (константы для обозначения клеток, фигур, позиций и т. д.), добавим индивиды специальных типов — «доски». Доска — это массив размерности 8 на 8, каждый элемент которого может принимать значение 1 или 0.

Если B обозначение переменной или константы типа доска, то компоненты B далее будем обозначать как $B(i, j)$, полагая, что $i, j \in \{1, \dots, 8\}$.

Переменные и константы типа доска описывают используемые первичные шахматные понятия, которые зависят от геометрических характеристик позиций и от взаиморасположения фигур. Например, понятие четвертая вертикаль описывается константой B типа доска, значения компонент которой следующие:

$$B(i, j) = \begin{cases} 1, & \text{если } j = 4, 1 \leq i \leq 8, \\ 0, & \text{в противном случае.} \end{cases}$$

К функциям структуры A отнесены базовые функции, перечисленные в табл. 1, а также двухместные функции объединения и пересечения досок, обозначенные символами $\cup$ и $\cap$. Термы $B^1 \cup B^2$ и $B^1 \cap B^2$ языка L интерпретируются следующим образом: если O — один из символов $\cup$ либо $\cap$, а B^1 и B^2 — доски, то значение терма $B^1 O B^2$ есть доска, компоненты которой определяются дизъюнкцией (если O — это символ $\cup$), либо конъюнкцией (если O — это символ $\cap$) соответствующих компонент.

К предикатам структуры A отнесены бинарные предикаты равенства и включения, определенные на индивидах типа доска, а также унарные предикаты, выделяющие области изменения переменных.

Базовые функции

Обозначение	Тип аргументов	Тип значения	Примечание
PLACE (f)	фигура	клетка	Координаты фигуры f в данной позиции.
COLOR (f)	фигура	цвет	Цвет фигуры f .
NEXT (SQ)	клетка	доска	Если $SQ = (i, j)$, то функция описывает вертикали $i-1$ и $i+1$.
ROUND (SQ)	клетка	доска	Если $SQ = (i, j)$, то функция описывает пояс клетки, т. е. множество $\{(i-1, j-1), (i, j-1), \dots, (i+1, j), (i+1, j+1)\}$.
AROUND (SQ)	клетка	доска	Если $SQ = (i, j)$, то функция описывает множество $\{(i-2, j-2), (i-2, j-1), \dots, (i+2, j+2)\}$.
E-COLOR (C)	цвет	цвет	Определение цвета фигур противника.
MANS (C)	цвет	доска	Описание фигур цвета C .
PAWNS (C)	цвет	доска	Описание пешек цвета C .
P-MOVES (C)	цвет	доска	Описание ходов пешек цвета C .
P-HITS (C)	цвет	доска	Описание ударов пешек цвета C .
M-MOVES (C)	цвет	доска	Описание ходов фигур цвета C .
M-HITS (C)	цвет	доска	Описание ударов фигур цвета C .
LINE (SQ, L)	клетка, линия	доска	Описание вертикали, горизонтали или диагонала с клеткой SQ .
C-LINE ($SQ1, SQ2, L$)	клетка, клетка, линия	доска	Описание отрезка линии между $SQ1$ и $SQ2$.
FORWRD (SQ, C)	клетка, цвет	доска	Описание клеток, расположенных выше (ниже) клетки SQ , если C —белые (черные).
QUINING (SQ, C)	клетка, цвет	доска	Описание поля превращения пешки на SQ цвета C .
Q-FIELDS (SQ, C)	клетка, цвет	доска	Описание критических полей проходной пешки,
DISP ($SQ1, SQ2$)	клетка, клетка	доска	Описание прямоугольника с вершинами в клетках $SQ1$ и $SQ2$.
OPP ($SQ1, SQ2, L$)	клетка, клетка, линия	доска	Описание оппозиции по линии L клеток $SQ1, SQ2$.
POPP ($SQ1, SQ2, L$)	клетка, клетка, линия	доска	Описания конфигурации клеток, в которой через ход возможна оппозиция по линии L .

Предикатные символы равенства ($=$) и включения ($\subset$) языка L интерпретируются следующим образом: если B_1 и B_2 термы, имеющие значение типа доска, то:

(i) значение предиката $B_1 = B_2$ истинно, если значения термов B_1 и B_2 покомпонентно совпадают, и ложно, если хотя бы для одной компоненты $B_1(i, j) \neq B_2(i, j)$;

(ii) значение предиката $B_1 \subset B_2$ истинно, если для значений термов B_1 и B_2 выполняется покомпонентная импликация, т. е. $\forall ij (B_1(i, j) \rightarrow B_2(i, j))$, и ложно, если хотя бы для одной компоненты $B_1(i, j) = 1 \& B_2(i, j) = \emptyset$.

Унарные предикаты и их интерпретация приведены в табл. 2. Формулу языка L , интерпретированную в структуре A , назовем допустимой, если она либо не содержит кванторов и унарных предикатов, либо находится в предваренной форме с ограниченными кванторами, а сами ограниченные кванторы имеют вид $\exists x p(x)$ либо $\forall x p(x)$, где p — один из одноместных предикатов, перечисленных в табл. 2.

Таблица 2

Унарные предикаты

Предикат	Область истинности
T -MANS	переменные типа фигура
T -COLOR	переменные типа цвет
T -SQUARE	переменные типа клетка
T -LINE	переменные типа линия
T -BOARD	переменные типа доска

4. Базовые функции, включенные в описанную выше структуру A , получены в результате анализа дескрипторов [2]. При этом структура A позволяет получить допустимые формулы почти для всех из более 100 приведенных в [2] шахматных понятий (исключение составляют те понятия, которые сильно зависят от динамических характеристик позиции, в частности, количество ударов на клетку). Ниже в качестве примера приведены формульные представления десяти из них. Первые шесть формул определяют пешечные характеристики позиций, последние четыре использованы в [4] для описания стратегии в эндшпиле «король, ладья против короля». Описание каждого из этих понятий включает:

- 1) Описание на естественном языке;
- 2) Список констант, используемых в формуле;
- 3) Формульное уточнение понятия.

1. Проходная пешка: пешка называется проходной, если впереди этой пешки по вертикали нет пешек противника, и ни одна из этих клеток не бьется пешками противника.

Константы: SQ — клетка пешки, C — цвет пешки, F — вертикаль. Формула:

$$((NEXT(SQ) \cup LINE(SQ, F)) \cap FORWRD(SQ, C) \cap PAWNS(E - COLOR(C))) = \emptyset.$$

2. Блокированная пешка: пешка называется блокированной, если перед ней стоит пешка противника.

Константы: SQ — клетка пешки, C — цвет пешки, F — вертикаль.
Формула:

$$\neg (\text{LINE}(SQ, F) \cap \text{ROUND}(SQ) \cap \text{FORWRD}(SQ, C) \cap \\ \text{PAWNS}(E - \text{COLOR}(C) = \emptyset).$$

3. Изолированная пешка: пешка называется изолированной, если на соседних с ней вертикалях нет пешек того же цвета.

Константы: SQ — клетка пешки, C — цвет пешки.

Формула:

$$\neg (\text{NEXT}(SQ) \cap \text{PAWNS}(C) = \text{PLACE}(SQ)).$$

4. Отсталая пешка: пешка называется отсталой, если на горизонталях с данной клеткой и ниже нет пешек того же цвета.

Константы: SQ — клетка пешки, C — цвет пешки, R — горизонталь.

Формула:

$$((\text{LINE}(SQ, R) \cap \text{PAWNS}(C)) \cup (\text{PLACE}(SQ) \cap \\ \text{FORWRD}(SQ, E - \text{COLOR}(C)))) = \text{PLACE}(SQ).$$

5. Сдвоенная пешка: пешка называется сдвоенной, если на той же вертикали находится еще одна пешка своего цвета.

Константы: SQ — клетка пешки, C — цвет пешки, F — вертикаль.

Формула:

$$\neg (\text{LINE}(SQ, F) \cap \text{PAWNS}(C) = \text{PLACE}(SQ)).$$

6. Фаланга пешек: пешка образует фалангу с другой пешкой своего цвета, если последняя стоит на соседней с ней по горизонтали клетке.

Константы: SQ — клетка пешки, C — цвет пешки, R — горизонталь.

Формула:

$$\neg (\text{LINE}(SQ, R) \cap \text{ROUND}(SQ) \cap \text{PAWNS}(C) = \emptyset).$$

7. Отсечение ладьей: короли соперников находятся по разные стороны от вертикали или горизонтали, на которой находится ладья.

Константы: WK — белый король, BK — черный король, WR — белая ладья.

Формула:

$$\exists L T - \text{LINE}(L) (\neg (\text{LINE}(\text{PLACE}(WR), L) \cap \text{DISP}(\text{PLACE}(WK), \\ \text{PLACE}(BK)) = \emptyset) \& \neg (\text{LINE}(\text{PLACE}(WR), L) = \\ = \text{LINE}(\text{PLACE}(BK), L))).$$

8. Мат в эндшпиле «король, ладья против короля»: король черных находится под шахом и у него нет ходов.

Константы: WR — белая ладья, BK — черный король, B — черные.

Формула:

$$\exists L T - \text{LINE}(L) (M - \text{MOVES}(B) = \emptyset \& \text{LINE}(\text{PLACE}(WR), L) = \\ = \text{LINE}(\text{PLACE}(BK), L)).$$

9. Оппозиция королей: короли находятся в оппозиции, если они стоят на одной линии и расстояние между ними минимально.

Константы: WK , BK — белый и черный короли.

Формула:

$$\exists L \text{ T — LINE}(L) (\text{OPP}(\text{PLACE}(WK), \text{PLACE}(BK), L) = \text{PLACE}(WK) \cup \text{PLACE}(BK)).$$

10. Ладья атакована королем противника.

Константы: WR — ладья, B — черные.

Формула:

$$\neg (\text{PLACE}(WR) \cap M - \text{HITS}(B) = \emptyset).$$

3. ОПИСАНИЕ АЛГОРИТМА ФУНКЦИОНИРОВАНИЯ

1. Разработка компоненты DESC—VALUE проводилась в соответствии с принципами нисходящего структурного программирования, что позволило получить программу с иерархической модульной структурой. Ниже дается описание модулей первого и второго уровней. Модули нижних уровней выполняют технические функции вида сканирования текста, операций со стеком, вычислений функций и, поэтому детально не описываются. Приводимые описания работы модулей включают в себя описания входных и выходных параметров, словесное описание алгоритмов, тексты модулей на PL-подобном псевдоязыке. Используемый псевдоязык состоит из:

а) управляющих операторов DO — WHILE, END — DO и IF — THEN, ELSE — IF, ELSE, END — IF;

б) функциональных операторов, которые описываются на естественном языке и задают обращение к модулям нижних уровней.

2. Модуль DESC—VALUE реализован на языке PL / 1 в виде процедуры — функции, входными параметрами которой являются: символьная строка DESC, задающая текст формулы дескриптора допустимого типа (см. 2.2) в виде последовательности соответствующих логических символов; описание позиции POS, задающее список фигур текущей позиции.

При обращении к модулю DESC — VALUE в любом месте PL-программы возвращается битовое значение '1'B, если формула DESC истинна в позиции POS и '0'B, в противном случае.

Так как каждая допустимая формула может содержать кванторную приставку, то матрица формулы должна проверяться для всех распределений связанных переменных формулы, определяемых кванторами. Для того, чтобы обеспечить простоту проверки матрицы для конкретного распределения, она переводится в постфиксную форму записи. Алгоритм работы модуля заключается в выделении приставки и матрицы формулы, построении постфиксной формы и цикла, на каждом шаге которого вычисляется значение матрицы для текущего распределения переменных и порождается очередное распределение.

Ниже следует текст модуля на псевдокоде:

DESC — VALUE: PROCEDURE

(DESC, /* ПРОВЕРЯЕМАЯ ФОРМУЛА */

POS /* ТЕКУЩАЯ ПОЗИЦИЯ */)

RETURNS (BIT (1)) /* ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ */;

< ВЫДЕЛИТЬ ПРИСТАВКУ И МАТРИЦУ ФОРМУЛЫ DESC >;

< ПЕРЕВЕСТИ МАТРИЦУ В ПОСТФИКСНУЮ ФОРМУ >;

< ОПРЕДЕЛИТЬ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ СВЯЗАННЫХ ПЕРЕМЕННЫХ >;

DO WHILE (< ФОРМУЛА НЕ ВЫЧИСЛЕНА >);

< ВЫЧИСЛИТЬ ЗНАЧЕНИЕ МАТРИЦЫ ДЛЯ ДАННОГО РАСПРЕДЕЛЕНИЯ ПЕРЕМЕННЫХ >;

< ПОЛУЧИТЬ ОЧЕРЕДНОЕ РАСПРЕДЕЛЕНИЕ СВЯЗАННЫХ ПЕРЕМЕННЫХ >;

END — DO;

RETURN (< ЗНАЧЕНИЕ ФОРМУЛЫ >);

END DESC — VALUE;

3. Модуль IN—TO—PST. Входными данными для модуля является текст матрицы формулы в инфиксной форме. Преобразование матрицы в постфиксную форму выполняется на основе следующей приоритетной функции P , определенной для символов операций и скобок.

Символ	Приоритет P
(,)	0
U n	1
= c	2
∨ &	3
⌈	4

Исходный текст сканируется слева направо. Если очередная лексема является константой типа доска либо функцией, то она становится очередной лексемой постфиксной формы; остальные символы поступают в стек и далее либо удаляются (если это скобки), либо перемещаются в результирующую постфиксную форму в соответствии с их приоритетами.

Ниже приведено описание модуля IN—TO—PST на псевдокоде:

IN — TO — PST; PROCEDURE(

INMAT /* ИНФИКСНАЯ ФОРМА */

PSTMAT /* ПОСТФИКСНАЯ ФОРМА */)

I = 1;

DO WHILE (< INMAT НЕ ИСЧЕРПАНО >);

LEX = < ОЧЕРЕДНАЯ ЛЕКСЕМА ИНФИКСНОЙ ФОРМЫ >;

IF LEX = '(' THEN

< ЗАСЛАТЬ В СТЕК СИМВОЛ '(' >;

ELSE — IF LEX = < БАЗОВАЯ ФУНКЦИЯ > ∨

LEX = < ПЕРЕМЕННАЯ ТИПА ДОСКА > ∨

LEX = < КОНСТАНТА ТИПА ДОСКА > THEN


```

    < ПЕРЕСЛАТЬ LEX В ФОРМУ PSTMAT >;
ELSE — IF LEX = ' THEN
DO WHILE ( < ОЧЕРЕДНОЙ ЭЛЕМЕНТ СТЕКА >  $\neq$  ' ');
    < ПЕРЕСЛАТЬ ОЧЕРЕДНОЙ ЭЛЕМЕНТ СТЕКА В ФОРМУ
        PSTMAT >;
END — DO;
    < ИСКЛЮЧИТЬ ' ИЗ СТЕКА >;
ELSE IF LEX = < СИМВОЛ ОПЕРАЦИИ >  $\vee$ 
    LEX = < ЭЛЕМЕНТАРНЫЙ ПРЕДИКАТ > THEN
DO WHILE ( < СТЕК НЕ ПУСТ > &
    P(LEX) < = P( < ЭЛЕМЕНТА СТЕКА > ));
    < ПЕРЕСЛАТЬ ЭЛЕМЕНТ СТЕКА В ФОРМУ PSTMAT >;
END — DO;
    < ЗАСЛАТЬ LEX В СТЕК >;
ELSE < ПЕЧАТЬ СООБЩЕНИЯ ОБ ОШИБКЕ >;
END — DO;
DO WHILE ( < СТЕК НЕ ПУСТ > );
    < ПЕРЕСЛАТЬ ЭЛЕМЕНТ СТЕКА В ФОРМУ PSTMAT >;
END — DO;
END IN — TO — PST;

```

4. Модуль MAT—VALUE. Входными данными для модуля MAT—VALUE является текст матрицы формулы в постфиксной форме и описание текущей позиции. Модуль организован как подпрограмма-функция, выдающая значение вычисляемой формулы. Для вычисления значения матрицы используется стек промежуточных результатов, в который заносятся либо указатели на вычисленные ранее доски (операнды для операций $\cup$, $\cap$, $=$ и $\subset$), либо значения атомных формул (операнды для операций $\neg$, $\vee$, $\&$). Исходный текст сканируется слева направо и если очередная лексема является операндом операций, то в стек заносится указатель на соответствующее значение, если же очередная лексема — операция, то эта операция применяется для верхних элементов стека, которые далее заменяются указателем на полученное значение. Результат вычисления матрицы формулы получается в вершине стека. Операции со стеком выполняются модулями нижних уровней.

Ниже приводится описание алгоритма работы модуля MAT—VALUE на псевдокоде.

```

MAT — VALUE: PROCEDURE (
    PSTMAT          /* ПОСТФИКСНАЯ ФОРМА */
    POS             /* ОПИСАНИЕ ПОЗИЦИИ */
    RETURNS (BIT (1) /* ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ */;
    I = 1;
    DO I = 1 BY 1 WHILE ( < PSTMAT НЕ ИСЧЕРПАНА > );
    LEX = < I-ТАЯ ЛЕКСЕМА PSTMAT >;
    IF — LEX = < БАЗОВАЯ ФУНКЦИЯ > THEN

```



```

        < ВЫЧИСЛИТЬ ФУНКЦИЮ И ЗАСЛАТЬ ЗНАЧЕНИЕ В
        СТЕК >;
ELSE — IF LEX = < КОНСТАНТА ТИПА ДОСКА > THEN
    < ПЕРЕСЛАТЬ LEX В СТЕК >;
ELSE — IF LEX = '7' THEN
    < ВЫПОЛНИТЬ ОПЕРАЦИЮ '7' В СТЕКЕ >;
ELSE — IF LEX ∈ {U, V, &, n, =, <} THEN
    < ВЫПОЛНИТЬ ДВУМЕСТНУЮ ОПЕРАЦИЮ LEX >;
ELSE < ВЫДАТЬ СООБЩЕНИЕ ОБ ОШИБКЕ >;
END — IF;
END — DO;
END MAT — VALUE;

```

Ր. Կ. ԿԱՐԱՊԵՏՅԱՆ

ՇԱԽՄԱՏԱՅԻՆ ԳԱՂԱՓԱՐՆԵՐԻ ՆԵՐԿԱՅԱՑՄԱՆ ՄԻԱՎԱՐՈՒՄ

Դիտարկված է վերջավոր քանորակներով առաջին կարգի ստորոգյալների հաշվի բանաձևերով տրված շախմատային գաղափարների նկարագրիչների իրացման համար նախատեսված ԱԿԱ-յ2 կապոցի ծրագրային բաղադրամասը։ Առաջարկված են նկարագրիչների հիմնական տարրերի համար տվյալների կազմություններ, որոնք թույլ են տալիս համակարգել և միավարել ինչպես նկարագրիչների ներկայացումները, այնպես էլ դրանց առանձին բաղադրամասերի մշակման ընթացակարգերը։ Տրվում է հետազոտված շախմատային գաղափարների բանաձևերի ցուցակ։

Л И Т Е Р А Т У Р А

1. Погосян Э. М. Адаптация комбинаторных алгоритмов, Ереван, Изд-во АН АрмССР, 1983.
2. Погосян Э. М., Амбарцумян М. А., Аджабян Н. А., Арутюнян Ю. Г., Джндоян Л. О., Карапетян Б. К. Пакет прикладных программ для исследований методов адаптации комбинаторных алгоритмов управления (АКА 1,2). Отчеты ВЦ АН АрмССР за 1974—1980 гг.
3. Погосян Э. М., Калантарян А. В., Карапетян Б. К. Требования к программному обеспечению исследований методов адаптации комбинаторных алгоритмов управления, см. настоящий сборник.
4. Карапетян Б. К. Язык высокого уровня для описания стратегий, см. настоящий сборник.
5. Шенфилд Дж. Математическая логика, М., Мир, 1975.