

19 Марта 2010

Инсерционное моделирование 1

Лекция 5

Введение в APS

В Херсоне



Система APS

Основные особенности:

Алгебраические структуры данных
(графовые термы)

Переписывание с последующей нормализацией

Использование стратегий переписывания

Интеграция основных парадигм:

императивное, функциональное, алгебраическое и
логическое программирование.

(2004 весна 3 курс)

Язык APLAN

Алгебраические программы

Алгебраические выражения

Переписывающие правила

Процедуры

Базовые операторы

Вычисление значений

Приведение к нормальной форме

**описания имен и отметок
начальные присваивания
инклюды**

**присваивание
установка
while, if
Вызовы
compile**

Имена и отметки

<name description> ::=

NAMES <sequence of names separated by "," >;

<name> ::= <identifier> | <identifier>[<integer>]

<mark description> ::=

MARK <sequence of mark descriptions elements separated by "," >;

<mark description element> ::= <mark symbol>(<arity>)|

<mark symbol>(2, <priority>, "<infix notation>")

<mark symbol> ::= <identifier>

<arity> ::= <positive integer> | UNDEF

<priority> ::= <positive integer>

<infix notation> ::= <sequence of signs>

Алгебраические выражения

Отметки

интерпретированные

неинтерпретированные

Синтаксис алгебраических выражений

Два вида вычислений

- приведение к базовой нормальной форме с подстановкой значений имен;
- приведение без подстановки.

Синтаксис алгебраических выражений

<algebraic expression>::=<primary expression>|<prefix expression>|
 <application>|<infix expression>
<primary expression>::=<integer or rational number> | <string>|
 <empty object> | <name> | <atom> | VAL <name>|
 (<algebraic expression>)
<empty object>::= ()
<application>::=<algebraic expression> <algebraic expression>
<prefix expression>::=
 <mark symbol>(<sequence of algebraic expressions separated by ",")
<infix expression>::=
 <algebraic expression><infix notation> <algebraic expression>

Императивный APLAN

<statement> ::= <basic statement> | <conditional statement> |
 <while statement> | <do statement> | <internal call> |
 <external call> | return | return(<algebraic expression>) |
 (<program>)

<basic statement> ::= <set statement> | <assignment statement>
<set statement> ::= <selector> --> <algebraic expression>
<assignment statement> ::= <name> := <algebraic expression>
<selector> ::= <name> |
 arg(<selector>, <sequence of expressions separated by ", ">)

<conditional statement> ::= <condition> -> <statement> |
 <condition> -> <statement> else <statement>

<while statement> ::= while(<condition>, <statement>)
<do statement> ::= do(<name>)
<internal call> ::= <internal procedure name>(<actual parameter list>)
<internal name> ::= <atom>
<external call> ::=
 <name of procedure definition>(<actual parameter list>)
<actual parameter> ::= <algebraic expression>

<procedure definition> ::= proc(<formal parameters list>)
<local names> <statement>
<local names> ::= loc(<local names list>) | <empty>
<formal parameter> ::= <identifier>

Демонстрация

Алгебраическая программа

2-applications/1-Hallo-world

Стандартные отметки

1-system/1-aplan/std.ap

Системы переписывания

Арифметические операции и предикаты
Логические операции
Императивный АПЛАН
 два присваивания
Неинтерпретированные операции
Встроенные функции

2-applications/2-algebraic programming/Initial