Розширення мови SIPL. Deque

1. Синтаксис

БНФ

|  |  |  |
| --- | --- | --- |
| **Ліва частина правила –**  **метазмінна** | **Права частина правила** | **Ім'я**  **правила** |
| *<*програма*>* ::= | **begin** *<*оператор*>* **end** | *NP*1 |
| *<*оператор*>* ::= | *<*змінна*>* := *<*вираз*>* |  *<*оператор*>* ; *<*оператор*>*|  **if** *<*умова*>* **then** *<*оператор*>*  **else** *<*оператор*>* |  **while** *<*умова*>* **do** *<*оператор*>* |  **begin** *<*оператор*>* **end** |  **skip |**  **append\_left(***<*вираз*>***) |**  **appent\_right(***<*вираз*>***)** | *NS*1  *NS*2  *NS*3  *NS*4  *NS*5  *NS*6  ***NS*7**  ***NS*8** |
| *<*вираз*>* ::= | *<*число*>* | *<*змінна*>* | *<*вираз*>* + *<*вираз*>* | *<*вираз*>* – *<*вираз*>* | *<*вираз*>* \* *<*вираз*>* | (*<*вираз*>*) |  **pop\_left()** |  **pop\_right()** | *NA*1  *…*  *NA*6  ***NA*7**  ***NA*8** |
| *<*умова*>* ::= | *<*вираз*>* = *<*вираз*>* | *<*вираз*>* *>* *<*вираз*>* | *<*умова*>* ∨ *<*умова*>* | ¬ *<*умова*>* |  (*<*умова*>*) |  **is\_empty()** | *NB*1  *…*  *NB*5  ***NB6*** |
| *<*змінна*>* ::= | . . . *M* | *N* | . . . | *NV*… |
| *<*число*>* ::= | . . . –1 | 0 | 1 | 2 | 3 | . . . | *NN*… |

Метазмінні

|  |  |  |
| --- | --- | --- |
| Метазмінна | Синтаксична категорія | Нова метазмінна |
| *<*програма*>* | *Prog* | *P* |
| *<*оператор*>* | *Stm* | *S* |
| *<*вираз*>* | *Aexp* | *a* |
| *<*умова*>* | *Bexp* | *b* |
| *<*змінна*>* | *Var* | *x* |
| *<*число*>* | *Num* | *n* |

Синтаксис у нових метазмінних

|  |  |  |
| --- | --- | --- |
| **Ліва частина правила –**  **метазмінна** | **Права частина правила** | **Ім'я**  **правила** |
| *P*::= | **begin** *S* **end** | *P*1 |
| *S*::= | *x* := *a* | *S*1 ; *S*2| **if** *b* **then** *S*1 **else** *S*2 |  **while** *b* **do** *S* | **begin** *S* **end** | *skip |*  **append\_left(а) |**  **appent\_right(*а*)** | *S*1–*S*6  S7  S8 |
| *a*::= | *n* | *x* | *a*1 + *a*2 | *a*1 – *a*2 | *a*1 \* *a*2 | (*a*) |  **pop\_left()** |  **pop\_right()** | *A*1– *A*6  A7  A8 |
| *b*::= | *a*1 = *a*2 |*a*1 *> a*2 | *b*1 ∨ *b*2 |¬*b* | (*b*) |  **is\_empty()** | *B*1 – *B*5  B6 |
| *x*::= | . . . *M* | *N* | . . . | *NV*… |
| *n*::= | . . . –1 | 0 | 1 | 2 | 3 | . . . | *NN*… |

2. Композиційна семантика розширення

Дані

*Int*= { . . . , –1, 0, 1, 2, . . . }

*Bool*= {*true*, *false*}

*Var*= {…, *M*, *N*, … }

**Deque = {…, di, …}, i {-}**

*State*= *Var* → *Int*

*A*\_*Int*\_*Bool*\_*State\_Deque* = *<Int*, *Bool*, *State*; *add*, *sub*, *mult*, *or*, *neg*, *eq*, *gr*, **emp,** ⇒*x*, *x*⇒, , *id*, ∇, **al, ar, pl, pr**>

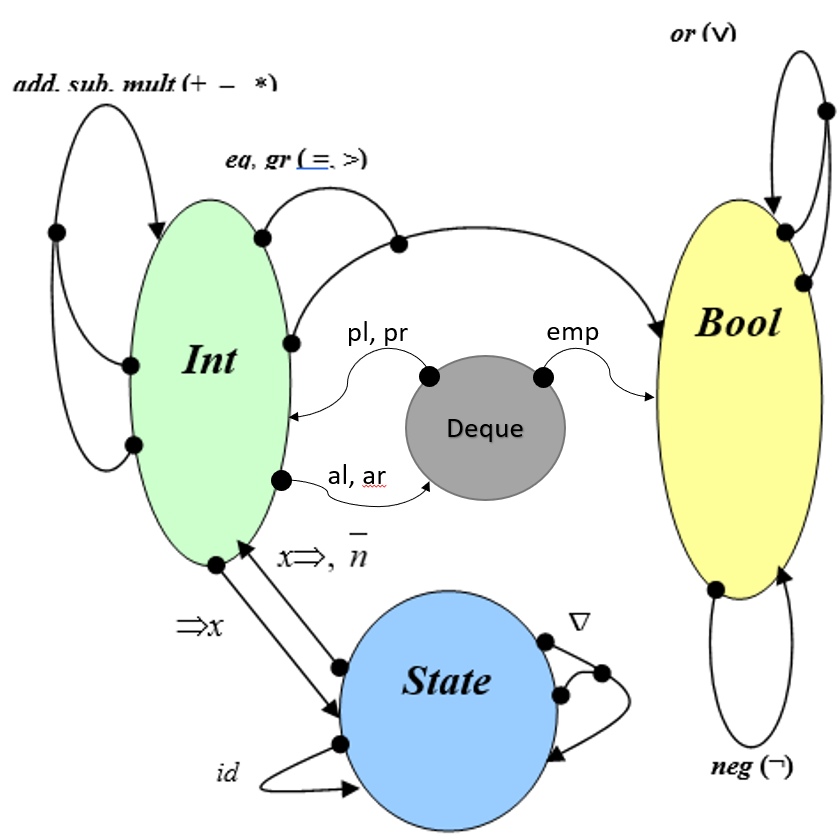
Функції

1. *n*-арні операції над базовими типами:

* *FNA*= *Intn* → *Int* – *n*-арні арифметичні функції (операції);
* *FNB*= *Booln* → *Bool* – *n*-арні булеві функції (операції);
* *FNAB*= *Intn* → *Bool* – *n*-арні функції (операції) порівняння.

1. Функції над станами змінних:

* *FA = State → Int – номінативні арифметичні функції*;
* *FB = State → Bool – номінативні предикати*;
* *FS = State → State – біномінативні функції-перетворювачі*
* ***FD = Deq → Int – номінативні функції над deque***

**

*Формули для обчислення композицій і функцій алгебри A\_Prog\_Deq:*

|  |  |  |
| --- | --- | --- |
| **Композиція** | **Формула обчислення** | **Ім’я формули** |
| Суперпозиція | (*S n*(*f*, *g*1,…, *gn* ))(*st*) = *f*(*g*1(*st*),…, *gn*(*st*)) | AF\_S |
| Присвоювання | *AS x* (*fa*)(*st*) = *st* ∇[*x* *fa*(*st*)] | AF\_AS |
| Послідовне виконання | (*fs*1• *fs*2)(*st*) = *fs*2(*fs*1(*st*)) | AF\_SEQ |
| Умовний оператор |  | AF\_IF |
| Цикл | *WH*(*fb*, *fs*)(*st*) = *stn*, де *st*0 = *st*, *st*1 = *fs*(*st*0), *st*2 = *fs*(*st*1),…, *stn* = *fs*(*stn*-1), причому *fb*(*st*0) = *true*, *fb*(*st*1) = *true*,…, *fb*(*stn*-1) = *true*, *fb*(*stn*) = *false* | AF\_WH |
| Функція розіменування | *x* ⇒ (*st*) = *st*(*x*) | AF\_DNM |
| Тотожна функція | *id*(*st*) = *st* | AF\_ID |
| **Append\_left** | ***AL* (*fa*)(*st*) = deq∇[di(min)-1 *fa*(*st*)]** | **AF\_AL** |
| **Append\_right** | ***AR* (*fa*)(*st*) = deq∇[di(max)+1 *fa*(*st*)]** | **AF\_AR** |
| **Pop\_left** | **PL (deq) = deq(di(min)) • deq∇[di(min) *null*]** | **AF\_PL** |
| **Pop\_right** | **PR (deq) = deq(di(max)) • deq∇[di(max) *null*]** | **AF\_PR** |
| **Is\_empty** | **EMP (deq) = true, if len(deq)>0, else false** | **AF\_EMP** |

3. Правила перетворення програми на семантичний терм:

|  |  |
| --- | --- |
| **Правило заміни** | **Номер**  **правила** |
| *sem*\_*P*: *Prog* → *TFS* задається правилами: | |
| *sem*\_*P*(**begin** *S* **end**) =  *sem*\_*S*(*S*) | *NS*\_*Prog* |
| *sem*\_*S*: *Stm* → *TFS* задається правилами: | |
| *sem*\_*S*(*x*:= *a*) = *AS x*(*sem*\_*A*(*a*))  *sem*\_*S*(*S*1 ; *S*2) = *sem*\_*S*(*S*1) • *sem*\_*S*(*S*2)  *sem*\_*S*(**if** *b* **then** *S*1 **else** *S*2) =  *= IF*(*sem*\_*B*(*b*), *sem*\_*S*(*S*1), *sem*\_*S*(*S*2))  *sem*\_*S*(**while** *b* **do** *S*) = *WH*(*sem*\_*B*(*b*), *sem*\_*S*(*S*))  *sem*\_*S*(**begin** *S* **end**) = (*sem*\_*S*(*S*))  *sem*\_*S*(*skip*) = *id*  ***sem*\_*S*(append\_left(*a)*) = *AL*(*sem*\_*A*(*a*))**  ***sem*\_*S*(append\_right(*a)*) = *AR*(*sem*\_*A*(*a*))** | *NS*\_*Stm*\_*As*  *NS*\_*Stm*\_*Seq*  *NS*\_*Stm*\_*If*  *NS*\_*Stm*\_*Wh*  *NS*\_*Stm*\_*BE*  *NS*\_*Stm*\_*skip*  ***NS\_Stm\_Al***  ***NS\_Stm\_Ar*** |
| *sem*\_*A*: *Aexp* → *TFA* задається правилами: | |
| *sem*\_*A*(*n*)) =  *sem*\_*A*(*x*)) = *x*⇒  *sem*\_*A*(*a*1+ *a*2) = *S*2(*add*, *sem*\_*A*(*a*1), *sem*\_*A*(*a*2))  *sem*\_*A*(*a*1– *a*2) = *S*2 (*sub*, *sem*\_*A*(*a*1), *sem*\_*A*(*a*2))  *sem*\_*A*(*a*1\* *a*2) = *S*2(*mult*, *sem*\_*A*(*a*1), *sem*\_*A*(*a*2))  *sem*\_*A*((*a*)) = *sem*\_*A*(*a*)  **sem\_A(pop\_left()) = PL(deq)**  **sem\_A(pop\_right()) = PR(deq)** | *NS*\_*A*\_*Num*  *NS*\_*A*\_*Var* *NS*\_*A*\_*Add*  *NS*\_*A*\_*Sub*  *NS*\_*A*\_*Mult*  *NS*\_*A*\_*Par*  **NS\_A\_Pl**  **NS\_A\_Pr** |
| *sem*\_*B*: *Bexp* → *TFB* задається правилами: | |
| *sem*\_*B*(*a*1**=***a*2) = *S*2(*eq*, *sem*\_*A*(*a*1), *sem*\_*A*(*a*2))  *sem*\_*B*(*a*1***>****a*2) = *S*2(*gr*, *sem*\_*A*(*a*1), *sem*\_*A*(*a*2))  *sem*\_*B*(*b*1**∨***b*2) = *S*2(*or*, *sem*\_*B*(*b*1), *sem*\_*B*(*b*2))  *sem*\_*B*(¬*b*) = *S*1(*neg*, *sem*\_*B*(*b*))  *sem*\_*B*((*b*)) = *sem*\_*B*(*b*)  **sem\_B(is\_empty()) = EMP(deq)** | *NS*\_*B*\_*eq*  *NS*\_*B*\_*gr*  *NS*\_*B*\_*or*  *NS*\_*B*\_*neg*  *NS*\_*B*\_*Par*  **NS\_B\_Emp** |

4. Операційна семантика

|  |  |
| --- | --- |
| **Назва правила** | **Правило операційної семантики** |
| Правила для програми та операторів: | |
| PR |  |
| AS |  |
| **AL** |  |
| **AR** |  |
| SEQ |  |
| IF*true* |  |
| IF*false* |  |
| WH*false* |  |

|  |  |
| --- | --- |
| WH*true* |  |
| BEG |  |
| skip |  |
| Правила для виразів: | |
| Num |  |
| Var |  |
| A+ |  |
| A– |  |
| A\* |  |
| A() |  |
|  |  |
|  |  |
| Правила для умов: | |
| B= |  |
| B> |  |
| B∨ |  |
| B¬ |  |
| B() |  |