Arrays

sample declaration:

var a:array[1..10] of char;

b:array['a'..'z'] of integer;

we will use a special array data type.

Keyword 'array' must be followed with [ ]. Inside of [ ] we have two constant of type int, char, boolean. Must be ordinal and same type. This is followed by the type declaration, which we will restrict to int, char, boolean.

TK\_AN\_ARRAY

Address of array

Size of array == (hi – low+1)\*elsize;

index type; ==> Integer, Boolean, Character

element type; ==> Integer, Real, Boolean, Character

lo

hi

Create this record (above) and allocated memory for it.

**Use of Array:**

LHS RHS

a[i] := 10; x := a[i];

RHS Use:

This is added to expressions:

We need an extra production in F:

F → ida [E];

F → idv

Save info about id:

match(TK\_LB);

type t= E();

match(TK\_RB);

if (t != indextype) error();

//generate: //if (low != 0)

pushi lo

sub

//part two

push size(element)

mul

//part three

push @array

add

//new instruction “get” will select the top of stack as address and get the element at that location

//in memory.

Get

return element\_type;

LHS:

<assignments> → variables

**arrays**

array must have index (arr[E])

MATCH(TK\_LB)

type t =e();

match(TK\_RB)

match(TK\_ASSIGN);

t = E();

match t against element\_type;

convert if needed.

Put

–---------========-----------=-------------=-=-=-=-=-=-=-=-=-=-=

{$R+}

{$R-}

if (t != indextype) error();

//generate: //if (low != 0)

→

**if (range\_check){**

**dup**

**pushi lo**

**lss**

**jtrue error**

**dup**

**pushi hi**

**gtr**

**jtrue error**

**}**

pushi lo

sub

–--------------------------------------------------------------------

bound lo, hi,line\_no

**STRING:**

strings are 256 chars maximum.

Metaware compiler.