Procesadores de Lenguajes

Memoria de proyecto — Hito 4:Finalización del procesador

GRUPO 14
RODRIGO SOUTO SANTOS
LEONARDO PRADO DE SOUZA
JUAN ANDRÉS HIBJAN CARDONA
IZAN RODRIGO SANZ

Grado en Ingeniería informática
Facultad de Informática
Universidad Complutense de Madrid

2024–2025

Índice general

1 | Especificación del Procesamiento de Vinculación

```
var ts, idens

def vincula(programa(Bloq)):
      ts = creaTS()
      vincula(SecDecs)
def vincula(bloque(SecDecs, SecIs)):
      abreAmbito(ts)
      recolectaDecs(SecDecs)
      vincula(SecDecs)
      vincula(SecIs)
      cierraAmbito(ts)
def vincula(si_decs(LDecs)):
      vincula(LDecs)
def vincula(no_decs()):x
def vincula(muchas_decs(LDecs, Dec)):
      vincula(LDecs)
      vincula(Dec)
def vincula(una_dec(Dec)):
      vincula(Dec)
def vincula(dec_base(TipoNom)):
      vincula(TipoNom)
def vincula(dec_type(TipoNom)):
      vincula(TipoNom)
def vincula(dec_proc(iden, ParamFs, Bloq)):
      abreAmbito(ts)
      if contiene(ts, iden):
            error()
      else  inserta(ts, iden, $)
      vincula(ParamFs)
      vincula(Bloq)
      cierraAmbito(ts)
def vincula(si_params_f(LParamFs)):
      vincula(LParamFs)
def vincula(no_params_f()):x
def vincula(muchos_params_f(LParamFs, ParamF)):
      vincula(LParamFs)
      vincula(ParamF)
def vincula(un_param_f(ParamF)):
      vincula(ParamF)
def vincula(si_refparam_f(Tipo, iden)):
      vincula(Tipo)
      if contiene(ts, iden):
            error()
      else  inserta(ts, iden, $)
def vincula(no_refparam_f(Tipo, iden)):
      vincula(Tipo)
      if contiene(ts, iden):
            error()
      else  inserta(ts, iden, $)
def vincula(tipo_nombre(Tipo, iden)):
      vincula(Tipo)
def vincula(tipo_array(Tipo, litEntero)):
      vincula(Tipo)
```

```
        if litEntero < 0:
                error()
def vincula(tipo_indir(Tipo)):
        vincula(Tipo)
def vincula(tipo_int()):x
def vincula(tipo_real()):x
def vincula(tipo_bool()):x
def vincula(tipo_string()):x
def vincula(tipo_type(iden)):
        $.vinculo = vinculoDe(ts, iden)
        if (not $.vinculo) or (not $.vinculo == dec_type(TipoNom)):
                error()
def vincula(tipo_struct(LCampos)):
        idens = []
        vincula(LCampos)
def vincula(muchos_campos(LCampos, TipoNom)):
        vincula(LCampos)
        if contiene(idens, TipoNom.iden):
                error()
        else    inserta(idens, TipoNom.iden)
        vincula(TipoNom)
def vincula(un_campo(TipoNom)):
        vincula(TipoNom)
        if contiene(idens, TipoNom.iden):
                error()
        else    inserta(idens, TipoNom.iden)
def vincula(si_ins(LIs)):
        vincula(LIs)
def vincula(no_ins()):x
def vincula(muchas_ins(LIs, I)):
        vincula(LIs)
        vincula(I)
def vincula(una_ins(I)):
        vincula(i)
def vincula(ins_eval(Exp)):
        vincula(Exp)
def vincula(ins_if(Exp, Bloq)):
        vincula(Exp)
        vincula(Bloq)
def vincula(ins_if_else(I, Bloq)):
        vincula(I)
        vincula(Bloq)
def vincula(ins_while(Exp, Bloq)):
        vincula(Exp)
        vincula(Bloq)
def vincula(ins_read(Exp)):
        vincula(Exp)
def vincula(ins_write(Exp)):
        vincula(Exp)
def vincula(ins_nl()):
        vincula()
def vincula(ins_new(Exp)):
        vincula(Exp)
def vincula(ins_delete(Exp)):
        vincula(Exp)
def vincula(ins_call(iden, ParamRs)):
        $.vinculo = vinculoDe(ts, iden)
        if not $.vinculo:
                error()
        vincula(ParamRs)
```

```
def vincula(ins_bloque(Bloq)):
        vincula(Bloq)
def vincula(si_params_r(LParamRs)):
        vincula(LParamRs)
def vincula(no_params_r()):x
def vincula(muchos_params_r(LParamRs, Exp)):
        vincula(LParamRs)
        vincula(Exp)
def vincula(un_param_r(Exp)):
        vincula(Exp)
def vincula(exp_asig(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_menor(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_menor_ig(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_mayor(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_mayor_ig(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_ig(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_dist(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_suma(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_resta(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_and(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_or(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_mul(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_div(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_mod(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_menos(op)):
        vincula(op)
def vincula(exp_not(op)):
        vincula(op)
def vincula(exp_index(op0, op1)):
        vincula(op0)
        vincula(op1)
def vincula(exp_reg(op, iden)):
        vincula(op)
def vincula(exp_indir(op)):
        vincula(op)
def vincula(exp_entero(litEntero)):x
def vincula(exp_real(litReal)):x
def vincula(exp_true()):x
def vincula(exp_false()):x
def vincula(exp_cadena(litCadena)):x
def vincula(exp_iden(iden)):
        $.vinculo = vinculoDe(ts, iden)
        if (not $.vinculo) or (not $.vinculo == dec_base(TipoNom)):
                error()
def vincula(exp_null()):x

def recolectaDecs(si_decs(LDecs)):
        recolectaDecs(LDecs)
def recolectaDecs(no_decs()):x
def recolectaDecs(muchas_decs(LDecs, Dec)):
        recolectaDecs(LDecs)
        recolectaDec(Dec)
def recolectaDecs(una_dec(Dec)):
        recolectaDec(Dec)
def recolectaDec(dec_base(TipoNom)):
        if contiene(ts, TipoNom.iden):
                error()
        else    inserta(ts, TipoNom.iden, $)
def recolectaDec(dec_type(TipoNom)):
        if contiene(ts, TipoNom.iden):
                error()
        else    inserta(ts, TipoNom.iden, $)
def recolectaDec(dec_proc(iden, ParamFs, Bloq)):
        if contiene(ts, iden):
                error()
        else    inserta(ts, iden, $)
```

2 Especificación del Procesamiento de Pre-tipado

3 Especificación del Procesamiento de Comprobación de Tipos

4 Especificación del procesamiento de asignación de espacio

5 Descripción de las Instrucciones de la Máquina-p

6 Especificación del procesamiento de etiquetado

7 | Especificación del Procesamiento de Generación de Código