Evaluation Oviders of SDD

SDT schemes

There address codes types - quadroples

troples, enderect troples

Swortch of atements

Intermediate code for procedures

Evaluation condens of 3DD:
-> Syntax Disucted Defination.

Dependency graphs- determining an evaluation and a for the all ribute

Dependency grapho:

\*-for each pance-tree node x the graph has a node for each attribute assocrated with a node X.

node attribute 10011 have some depend

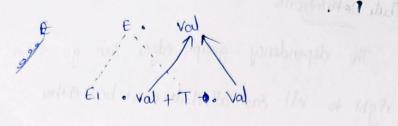
ed Redundacy elemination (PRE)

\* A Semantic stude associated & a with a production defenes the value synthesized all rip bulk A.b in Terms of X.c. the dependency graph in an edge from X.c. to A.b.

\*A semantic orde associated with a poroduction defines the value of Inherited attribute B.C Pro Terms of X.a., then the dependency graph has an edge from X.a to B.C

Postpal dependency graph. Production € → EI+T

Semantec Value €·Val -> €1·Val + T·Val synthesed ribute €·val



complete depency graph

today to the state of the state

Cordering the evaluation of Attribute:

If the dependency goiaph how an edge from node M to node N, then the attofbule at M'must be evaluated before the attribute of w'.

no entering edges

It sollows the topological sort of the graph, it there are no cycle.

3-attributed Defrittons:

An SDD is s-allebuted of every attribute os synthesoxed All robutes are evaluated on any bottom-up bades of the nodes of

passe tree. It so sample to evaluate the attributes by post oorder pure.

L-All Floute Defanthons

the dependency-goraph edges can go from left to right, but not right to left. Each attribute must be either

1. synthesized (or)

Eg: A -> XIXa ... Xn

Then the griberated attrobute XI:a can be computed

- a) from enheated altributed, with the head A! (x4.a)
- b from enheated or syntherized attrobutes located to the left of Xi (Xi X2 X3)

dong words degrees

c) from unhearted on synthesized attributes ausocrated on

dudants to continuous of procha

ast shot belief stop

example

production production sematric Rule

is to dulitalizate special buloulous at Tipohi & foral walled notice in with

dopp on one wealth to the to the test pologot with carolled to

boxPandlerya ar studishing your to believeline as a gos na

Allebotes are evaluated an any hottom op used of the nodes of

Syntax Droucted Translation Schema:

3DT= & Grammer + Semante rules

\* The syntax Droucted Translation Scheme 43 a context town grammer.

\* It evaluates the corder of semantic ordes.

\* In translation scheme, the Semantic sides are embedded within the right side of the poindudton lavor mun

\* The post from at which an action is to be executed is shown by enclosed between braces. It is corttlen within the right stee of the production

Example production

\$ → €\$

F +-> 3

3-3 2. d pgp+

J -> dPg9+

Semantec Rules.

Sprint evalue } Eprint E. value}

E-val = E-val + E-value}

¿ t. val : = E. val }

{E-val := I-val}

{ J. val = Tox J. val + lextral }

{ Zval:=lexval].

Implementation of syntax described translation describes as as to

\* SDT 95 perplemented by constructing a passe tree and performing the actions in a left to right depth first ander. a + b \* c

\*SDT PS amplementang by posses the apput and a posses tree as a result.

, Example:

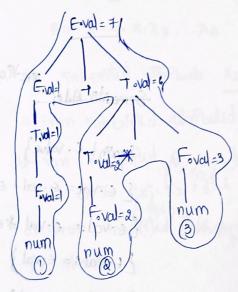
$$E \longrightarrow E+T \{E \cdot val := E \cdot val + T \cdot val\}$$

$$E \longrightarrow T \{E \cdot val := T \cdot val\}$$

$$T \longrightarrow T \{E \cdot val := T \cdot val * F \cdot val\}$$

$$T \longrightarrow F \{T \cdot val := F \cdot val\}$$

$$F \longrightarrow num \{F \cdot val := num \cdot lexval\}$$



Top down left to right

That address todes types - quadraples

\* It is an intermediate code. It is used by optimizing compiles.

\* In theree adores code, the grown experieuren is broken down rate several Seperate metructions. There metruction can easily triculate into Assembly s language severy o how lugar all severy pd partaundran er rask

consider translation schools

all ameles astleteast at a

of (messord assulation

destroy at to une type

\* Each thous addoses code protruction has at most thous operands. It is a Combination of assignment and a binary operation in TAC, these is at most one operator on the right side of an Pristruction Switch(E) [ 2+4 \* 2 4, & to are comple generated temperary names +1= 9#X ta=x+ti egr a + a\* (b-c) +d\* (b-c) (b-c) de-lautiso +1=b-c ta + 0/\* +1 ta= a\* +1 13 F/d\* +1 Entry = a+tout to matel mort ty=/+2+t3 244 #2 ty = d\*+1 t5= @'t3 + ty 15=a+t2+t3 3 frelds O quatriples ⇒4 fields \_ Sounce 1 Iroples \* source 2 destination eg: - a:=-b\*c+d Eg: a:=-b\*c+d equiplonation la production b TAC: (bold who) every to about down) i t3 = +1 \* ta tresibor, off rot dolo dechar= 13 traples ta = ti+d (or) ta = 444 operator Source 1 Source 2. T3=+ 17\*+2 a:= t2 (o) mรกพ a:= 13 Quatruples (1) plus Source a destrution C operator Source +1 (1) (a) Poto 4(0) b eunim (0) d 12 equalto (3) **Q**(g) plus (1) to 12 11 Ponto Paval to (2)

Switch-statement:

acresind before ter

type of selectron control statement that allows the value of a Vaurable to change the control flow of program execution.

eg:-Switch(E) [

Case VI: SI Case Va: Sa

(a) Vn-1: Sn-1de-fault: Sn

Translater of surter statements:

\* Evaluate the Express on E

\* FAND the value Vj an the last of cases

\* Execute the statement.

Steps) - Proplemented as

\* sequence of condestronal jumps btoxd on 15 (suate table of pairs (value, label) \* Hash table for the values

008/08/09/

eblost-pe-colorius

(0)

goto test

Li: code for 31 begins to see solo poto next seein second ile second to solo complete as changed to

la: code for sa

goto next

Informedists code can be suppressed in the transfer Ln-1 : code for Sn-1

gotonext

In: code for sn en destamon land 4917

goto next

De con supercritid as source cate test: of t = Vi goto Li odlowing 91 l=Va goto la

> Pf += Vn-1 goto kn-1 motor who want + mostlylo goto Ln 1) Abortock syntax free

Another translation of a suntich statements to for deslogic

Internediate code

It 93 used to translate the Source code Porto machine code. It lies between high level language and marhino language

e) These Address code.

according to the machine.

Intermediate Intermodicate Code
Code Generator Generator Paul seu Statec checkey

\* If the compres descetly translates source code ento machine code without generating entermedicate code then a full native compiler po ougusted to

each new machene \* Intermedante code generation recerves imput from its poudurs and phase on semantec analyzes phase. It takes enput on the form of annoted zing, tree \* using retermodrate code, the second phase of the compiler as changed 181 (och dor 38 accosiding to the machine. Job rext Internediate code can be suppresented in two ways Lines code for Se High level Intermediate code low level Intermedial code 21 Ps close to the target main TI can experiented as Source code It stop No Dr Ps used for machine admin 31 alog sv=1 1 Sptpnfxattoo. the defferent forms of Intermediate code , it is 1) Abstract syntax tree of stop 2) polos h notation made date de 3) Thous Address code.

is the compress descript travalete some code ento machine code wellout

- 1) stack allocation
- 2) access to non local data
- 3) Heap manage mont
- 4) Books and flow graphs
  - 5) optimization of basic blocks
  - 6) per phole optimization
  - 7) Regreter allocation and current.

State storage Allocation:

- \* In state allocation names are bound to storage locations.
- \* If memosy 93 covaled at comprle time then the memosy will be executed an state are a and only once

Stack allocations

Storage P3 auganteed as stack = Flast In first out

Activation succords are pushed and popped

Activation ou