I42：

WIL -> while (· ALS ) : { BLK },BLK

ALS -> ·ETT alop ETT,)

ALS -> ·ALS alop ETT,)

ALS -> ·ETT alop ETT,alop

ALS -> ·ALS alop ETT,alop

ETT -> ·id,alop

ETT -> ·DIGIT,alop

DIGIT -> ·int,alop

DIGIT ->·float,alop

ETT -> ·id,)

ETT -> ·DIGIT,)

DIGIT -> ·int,)

DIGIT ->·float,)

I41：

WIL -> while · ( ALS ) : { BLK },BLK

While

I35 ：

FUC ->id ( ETT· ), :

BLK

I32：

BLK -> VGL BLK·,}

VGL

I29：

DEF ->def FUC : { BLK · },BLK

}

sBLK

I30：

DEF ->def FUC : { BLK }·,BLK

Id DIGIT int float

I34 ：

FUC ->id (· ETT ), :

ETT ->·id, )

ETT ->·DIGIT,)

DIGIT ->·int,)

DIGIT ->·float,)

I37 ：

ETT ->·d·,）

I38:

ETT ->DIGIT·,)

I39:

DIGIT ->int·,)

I40:

DIGIT ->float·,)

)

ETT

I36：

FUC ->id ( ETT )·, :

(

Id

I33 ：

FUC ->id · ( ETT ), :

I28：

DEF ->def FUC : {· BLK },BLK

BLK -> ·VGL BLK,}

BLK -> ·IPT BLK,}

BLK -> ·DEF BLK,}

BLK -> ·WIL BLK,}

BLK -> ·IF BLK,}

BLK -> ·RET BLK,}

BLK -> ·NULL,}

VGL -> ·id vop ETT ,BLK

VGL -> ·id vop ALS ,BLK

IPT -> ·import id,BLK

DEF ->·def FUC : { BLK },BLK

WIL -> ·while ( ALS ) : { BLK },BLK

IF -> ·if ( ALS ) : { BLK } ELF else : { BLK },BLK

RET -> ·return ETT,BLK

{

I27：

DEF ->def FUC :· { BLK },BLK

：

I26：

DEF ->def FUC · : { BLK },BLK

FUC

I25：

DEF ->def ·FUC : { BLK },BLK

FUC -> ·id ( ETT ), :

I2：

BLK -> VGL · BLK ,$

BLK -> ·VGL BLK,$

BLK -> ·IPT BLK,$

BLK -> ·DEF BLK,$

BLK -> ·WIL BLK,$

BLK -> ·IF BLK,$

BLK -> ·RET BLK,$

BLK -> ·NULL,$

VGL -> ·id vop ETT ,BLK

VGL -> ·id vop ALS ,BLK

IPT -> ·import id,BLK

DEF ->·def FUC : { BLK },BLK

WIL -> ·while ( ALS ) : { BLK },BLK

IF -> ·if ( ALS ) : { BLK } ELF else : { BLK },BLK

RET -> ·return ETT,BLK

VGL IPT DEF WIL IF RET

def

Id

I24：

IPT -> import id· ,BLK

I23：

IPT -> import ·id ,BLK

import

ETT

I22：

ALS -> ALS alop ETT·,BLK

ALS -> ALS ·alop ETT,alop

Id DIGIT int float

alop

I21:

ALS -> ALS alop ·ETT,$

ALS -> ALS alop ·ETT,alop

ETT -> ·id,$

ETT ->·DIGIT,$

ETT -> ·id,alop

ETT ->·DIGIT,alop

DIGIT -> ·int,$

DIGT -> ·float,$

DIGIT -> ·int,alop

DIGT -> ·float,alop

I20：

VGL -> id vop ALS·,BLK

ALS -> ETT ·alop ETT,BLK

ALS -> ETT ·alop ETT.alop

ALS

Id DIGIT int float

Id DIGIT int float

I16：

ETT -> id·,BLK

ETT -> id·,alop

I17:

ETT -> DIGIT·,BLK

ETT -> DIGIT·,alop

I18:

DIGIT -> int·,BLK

DIGIT -> int·,alop

I19:

DIGIT -> float·,BLK

DIGIT -> float·,alop

I11 I12 I13

ALS -> ETT alop ETT,alop ETT -> id,alop ETT->DIGIT,alop

I14 I15

DIGIT ->int,alop DIGIT ->float,alop

I10：

ALS -> ETT alop ·ETT,alop

ETT -> ·id,alop

ETT -> ·DIGIT,alop

DIGIT -> ·int,alop

DIGIT -> ·float,alop

alop

I9：

ALS -> ETT alop ETT·,BLK

ALS -> ETT ·alop ETT,alop

ETT

I8：

ALS -> ETT alop ·ETT,BLK

ALS -> ·ETT alop ETT,alop

ETT -> ·id,BLK

ETT -> ·DIGIT,BLK

ETT -> ·id,alop

ETT -> ·DIGIT,alop

DIGIT -> ·int,BLK

DIGIT -> ·float,BLK

DIGIT -> ·int,alop

DIGIT -> ·float,alop

alop

ETT

I7：

VGL -> id vop ETT·,BLK

ALS -> ETT ·alop ETT,BLK

ALS -> ETT ·alop ETT.alop

vop

I6：

VGL -> id vop ·ETT,BLK

VGL -> id vop ·ALS,BLK

ETT -> ·id,BLK

ETT -> ·DIGIT,BLK

ETT -> ·id,alop

ETT -> ·DIGIT,alop

ALS -> ·ETT alop ETT,BLK

ALS -> ·ALS alop ETT,BLK

ALS -> ·ETT alop ETT,alop

ALS -> ·ALS alop ETT,alop

DIGIT -> ·int,BLK

DIGIT -> ·float,BLK

DIGIT -> ·int,alop

DIGIT -> ·float,alop

I5：

VGL -> id ·vop ETT ,BLK

VGL -> id ·vop ALS ,BLK

id

BLK

I0:

S -> ·BLK ,$

BLK -> ·VGL BLK,$

BLK -> ·IPT BLK,$

BLK -> ·DEF BLK,$

BLK -> ·WIL BLK,$

BLK -> ·IF BLK,$

BLK -> ·RET BLK,$

BLK -> ·NULL,$

VGL -> ·id vop ETT ,BLK

VGL -> ·id vop ALS ,BLK

IPT -> ·import id,BLK

DEF -> ·def FUC : { BLK },BLK

WIL -> ·while ( ALS ) : { BLK },BLK

IF -> ·if ( ALS ) : { BLK } ELF else : { BLK },BLK

RET -> ·return ETT,BLK

BLK

I3：

BLK -> VGL BLK· ,$

I1：

S -> BLK· ,$

Principles:

0 S -> BLK

1 BLK -> VGL BLK

2 BLK -> IPT BLK

3 BLK -> DEF BLK

4 BLK -> WIL BLK

5 BLK -> IF BLK

6 BLK -> RET BLK

7 BLK -> NULL

8 VGL -> id vop ETT

9 VGL -> id vop ALS

10 ALS -> ETT alop ETT

11 ALS -> ALS alop ETT

12 ETT -> id

13 ETT -> DIGIT

14 DIGIT -> int

15 DIGIT -> float

16 IPT -> import id

17 DEF -> def FUC : { BLK }

18 WIL -> while ( ALS ) : { BLK }

19 IF -> if ( ALS ) : { BLK } ELF else : { BLK }

20 ELF -> elif ( ALS ) : { BLK } ELF | NULL

21 RET -> return ETT

22 FUC -> id ( ETT )