

۱. ارائه دستورات به زبان اسمبلي
۲. تحليل نوع گرامر مربوطه (براساس جدول)
اگر گرامر ابهام داشت آن را رفع ابهام کنيد

انواع دستورات CPU

- ۱ انتقال داده: **MOV, PUSH, POP**
- ۲ محاسباتي : **ADD, SUB, MUL, DIV**
- ۳ منطقی: **AND, OR, XOR, NOT**
- ۴ کنترل: **JMP, JZ, CALL, RET**
- ۵ مقایسه: **CMP, TEST**

JMP Start_Loop : دستوري که می خواهیم استفاده کنیم

یک مثال برای دستور **JMP Start_Loop**

main:

```

MOV AX, 10      ; Load AX with 10
start_loop:      ; Loop label
    ADD AX, 1    ; Increment AX by 1
    CMP AX, 20   ; Compare AX with 20
    JNE start_loop ; Jump to start_loop if Not Equal
RET             ; Return from procedure

```

قانون گرامر عبارت است از :

```

<Instruction> ::= <Mnemonic> <Operand>
<Mnemonic>  ::= JMP | JZ | JNZ | JE | CALL
<Operand>   ::= <Label>
<Label>     ::= <Identifier>
<Identifier> ::= [a-zA-Z_][a-zA-Z0-9_]*

```

```
Instruction → Mnemonic Operand
Mnemonic → JMP
Operand → Label
Label → Identifier
Identifier → id
```

Bottom-Up Parse (Shift-Reduce)

Input: JMP start_loop

Stack: JMP

Status: Mnemonic recognized

Input: -

Stack: JMP start_loop

Status: Identifier recognized

Rule: Identifier → Label

Stack: JMP Label

Status: Label reduced

Rule: Label → Operand

Stack: JMP Operand

Status: Single operand

Rule: Mnemonic Operand → Instruction

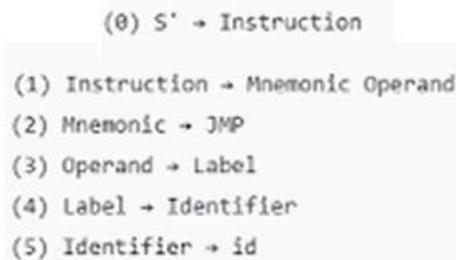
Stack: Instruction

Status: Valid JMP instruction

Final Stack: Instruction

Parse Complete: JMP start_loop is VALID

قاعده کلی عبارت است از:



جدول انتقالات: LR(0)

State	Input	Action	Next
0	JMP	Shift	S5
5	label	Reduce	R3
3	\$	Accept	

JMP Start_Loop گرامر کلی

```
I0:
S' → • Instruction
Instruction → • Mnemonic Operand
Mnemonic → • JMP
```

```
I5:
Mnemonic → JMP •
```

```
Instruction → Mnemonic • Operand
Operand → • Label
Label → • id
```

Label → id •

Operand → Label •

دستور **JMP Start_Loop** بدون ابهام است و از نظر نحوی معتبر است