

۱. ارائه دستورات به زبان اسمبلی
 ۲. تحلیل نوع گرامر مربوطه (براساس جدول)
- اگر گرامر ابهام داشت آن را رفع ابهام کنید

انواع دستورات CPU

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

دستوری که می خواهیم استفاده کنیم: **JMP Start_Loop**

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 \rightarrow Label

Stack: JMP Label

Status: Label reduced

Rule: Label \rightarrow Operand

Stack: JMP Operand

Status: Single operand

Rule: Mnemonic Operand \rightarrow Instruction

Stack: Instruction

Status: Valid JMP instruction

Final Stack: Instruction

Parse Complete: JMP start_loop is VALID

LR قاعده کلی عبارت است از:

- (0) $S' \rightarrow \text{Instruction}$
- (1) $\text{Instruction} \rightarrow \text{Mnemonic Operand}$
- (2) $\text{Mnemonic} \rightarrow \text{JMP}$
- (3) $\text{Operand} \rightarrow \text{Label}$
- (4) $\text{Label} \rightarrow \text{Identifier}$
- (5) $\text{Identifier} \rightarrow \text{id}$

جدول انتقالات: LR(0)

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

گرامر کلی JMP Start_Loop

```
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