東南大學
SOUTHEAST UNIVERSITY

Compiler Principle
——Programming Language

Zhizheng Zhang
Southeast University

1. Programming Language Basics

- Static/Dynamic
- Environments/States
- Static Scope/Block Structure
- Explicit Access Control
- Dynamic Scope
- Parameters Passing Machenisms
- Aliasing

1. 1 Static/Dynamic

- **Static vs. Dynamic**
  - ***Static Issues can be addressed during the compile time.***
    - ✓**E.g., Scope of declarations in C or Java. Location of objects of a static class.**
  - ***Dynamic Issues are addressed at running time***
    - ✓**E.g., Location of local variables.**

1. 2 Environments/States

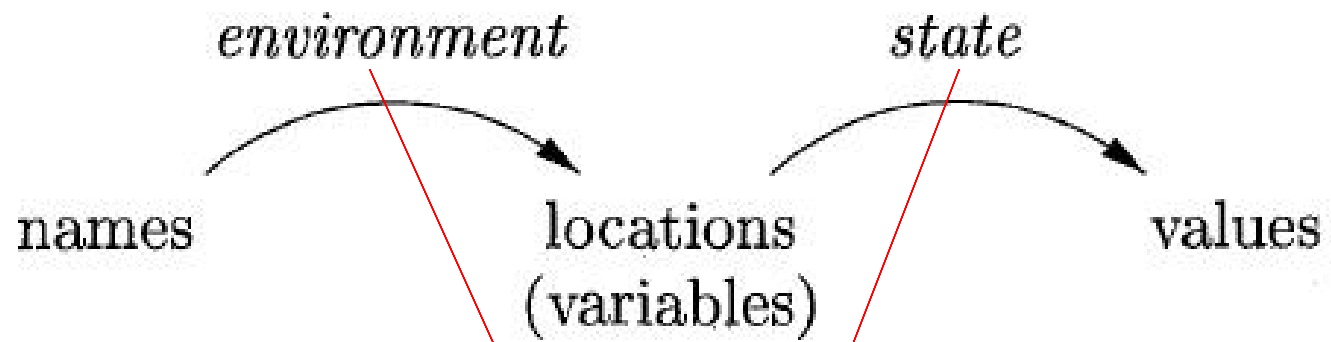


Figure 1.8: Two-stage mapping from names to values

Static /Dynamic

东南大学 SOUTHEAST UNIVERSITY

1. 3 Static Scopes/Block Structures

```
main() {
    int a = 1;                                   B1
    int b = 1;
    {
        int b = 2;                               B2
        {
            int a = 3;                           B3
            cout << a << b;
        }
        {
            int b = 4;                           B4
            cout << a << b;
        }
        cout << a << b;
    }
    cout << a << b;
}
```

Figure 1.10: Blocks in a C++ program

DECLARATION	SCOPE
`int a = 1;`	$B_1 - B_3$
`int b = 1;`	$B_1 - B_2$
`int b = 2;`	$B_2 - B_4$
`int a = 3;`	B_3
`int b = 4;`	B_4

东南大学 SOUTHEAST UNIVERSITY

1. 4 Explicit Access Control

Public
Private
Protected

1. 5 Dynamic Scope

A use of a name x refers to the declaration of x in the most recently called procedure with such a declaration

E.g., macro expansion, method resolution in O-O Programming

```
#define a (x+1)

int x = 2;

void b() { int x = 1; printf("%d\n", a); }

void c() { printf("%d\n", a); }

void main() { b(); c(); }
```

Figure 1.12: A macro whose names must be scoped dynamically

1. 6 Parameter Passing

- ***Call by Value***

  e.g. **f(x) {x=x+1; print (x)};**

  main {y=1; f(y+1);}

- ***Call by Reference***

  e.g. **array, pointer, object of all classes**

- ***Call by Name***

  e.g. **Macro definition**

1. 7 Aliasing

An interesting consequence of call-by-reference parameter passing. It is possible that two formal parameters can refer to the same location.

E.g., Suppose a *is an array belonging to a procedure* p*, and* p *calls* another procedure $q(x, y)$ with a call $q(a, a)$.

The important point is that if within q there is an assignment $x[10] = 2$**, then the value of** $y[10]$ **also becomes 2.**

2. Programming Language Design

- Artificial/formal Language
- Grammar
- Language Recognization
- Automata
- Grammar Construction

2. 1 Formal Language

□Formal(Artificial) Language

- **Vocabularies Table.** ***E.g., {a, b}***
- **Sentence:** ***a sequence of lexemes, E.g. ab, baa.***
- **Language:** ***a set of all sentences. E.g., {ab, baa}***

Note: not all sequences of lexemes are sentences, A sentence is a sequence grouped by constructions laws from lexemes.

OPERATION	DEFINITION AND NOTATION
Union of L and M	$L \cup M = \{s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$
Concatenation of L and M	$LM = \{st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$
Kleene closure of L	$L^* = \cup_{i=0}^{\infty} L^i$
Positive closure of L	$L^+ = \cup_{i=1}^{\infty} L^i$

L and M are sets of strings.

Example : Let L be the set of letters {A, B, . . . , Z, a, b, . . . , *z) and let* D be the set of digits {0,1,.. .9).

- $L \cup D$
- LD
- $L(L \cup D)^*$
- L^4
- D^+

2.2 Grammar

□Grammar I

- **A Grammar** ***includes a collection of construction laws which are usually represented in the form of productions, a set of terminal symbols, and a set of non-terminal symbols, one of those non-terminal symbols is start symbol.***

□Grammar II

- **Example 1.** $G=(T, N, P, S)$, where

$$
\begin{aligned}
&T=\{a, b\},\\
&N=\{A, B, S\},\\
&S\in N,\\
&P=\{S \to A|B,\\
&\qquad A \to ab,\\
&\qquad B \to baa\\
&\quad\}
\end{aligned}
$$

□Grammar III

From grammars to languages, there is a one-to-one mapping.

The language of a grammar *G* is written as *L*(*G*), which is the set of sequences of terminal symbols derived from the start symbol by replacing left sides of a production with its right side.

□Grammar VI

Derivation: ***Start with Start symbol, replacing an appear of a rule's left side with its right side.***

E.g., $S \to A,\ A \to ab,$

$S \to B,\ B \to baa.$

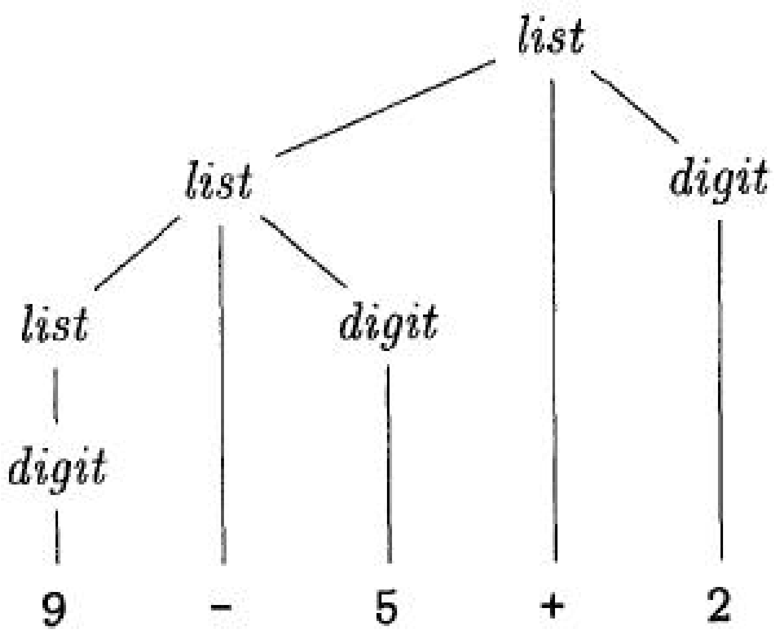
2.3 Language Recognization

☐TASK

Given a grammar G, and a sequence α of symbols, to check if α∈L(G), i.e., ***if α can be derived from the start symbol of G***.

□Example

$$
\begin{aligned}
list &\rightarrow list + digit \\
list &\rightarrow list - digit \\
list &\rightarrow digit \\
digit &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}
$$



list→list + digit

→list-digit+digit

→digit-digit+digit

→9-digit+digit

→9-5+digit

→9-5+2

□Ambiguous Grammar

Two or more parse trees for some sentences

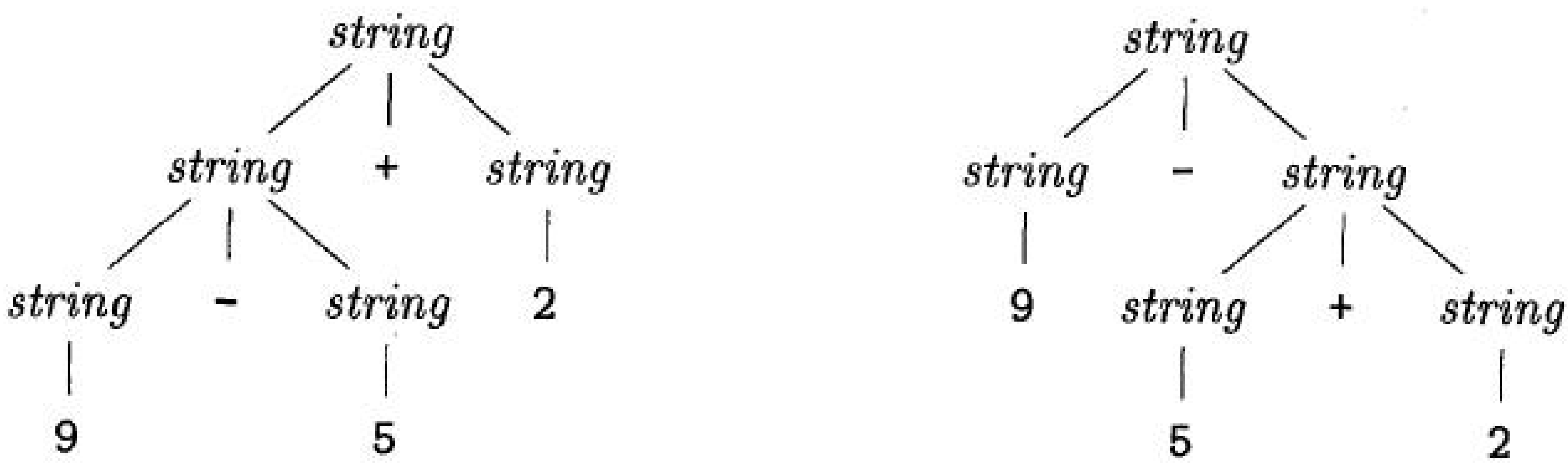


Figure 2.6: Two parse trees for 9-5+2

2.4 Automata

□Grammar vs Automata

An automata of a grammar can be regarded as an algorithm of recognizing its language.

See the relationship between grammars and automata in the next page.

Hierarchy	Alias	Production form	Automation name
0-type	Grammar without limitation	$\alpha \to \beta$. where $\alpha \in (V_N \cup V_T)^* V_N^+ (V_N \cup V_T)^*$, $\beta \in (V_N \cup V_T)^*$	Turing Machine
1-type	Context-sensitive grammar	$\alpha A \beta \to \alpha \gamma \beta$. Where $A \in V_N$, $\alpha, \beta \in (V_N \cup V_T)^*$	Linear Bound Automation
2-type	Context-free grammar	$A \to \beta$. Where $A \in V_N$, $\beta \in (V_N \cup V_T)^*$.	Pushdown automation
3-type	Regular grammar	$A \to \alpha B$ or $A \to \alpha$. where $A, B \in V_N$, $\alpha \in V_T^*$	Finite automation

2.5 Construct a grammar for a language

——Some Examples & Skills

Method 1. Stepwise Dividing (逐步分解法)

- E.g, $L=\{a^i \mid i\geq 0\}$

1) If $i=0$ $L\rightarrow\varepsilon$
2) if $i\geq 1$ $a^i = aa^{i-1}=aa^j$ $j\geq 0$
   $L\rightarrow aL$
3) The grammar is as following: $L\rightarrow aL|\varepsilon$

- E.g, $L=\{a^ib^j \mid i\geq 1, j\geq 0\}$

1) If i>1 $a^ib^j = aa^{i-1}b^j = aa^kb^j$ $k\geq 1, j\geq 0$ $L\rightarrow aL$
2) If i=1 $a^ib^j = ab^j$ $L\rightarrow aB$
3) $B\rightarrow bB|\varepsilon$
4) $L\rightarrow aB|\ aL$
   $B\rightarrow bB|\varepsilon$

- E.g, $L=\{i \mid i \bmod 5=0\}$

Firstbit	**middlebits**	**Lastbit**

●**For first bit**

$L \rightarrow$1A|2A|3A|……|9A

●**For middlebit**

A→0A|1A|2A|……|9A|B

●**For lastbit**

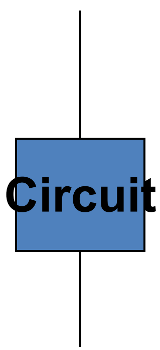
B →0|5

●**$L \rightarrow$ 0|5**

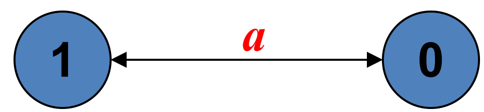
Method 2. Circuit Diagram (电路图法): odd and even problems

- E.g 1, $L=\{\omega \mid \omega\in a^*$, and the number of a in ω is odd$\}$

1) Imagine two states 0 and 1 in a circuit with one bit input



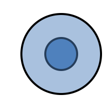
2) Two states can be transformed from each other with only one bit a input

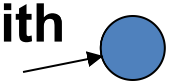


3) Decide the *start* state and *end* state.

We always assume that the state with full 0 bits is the end state. And we decide the start state according to the requirement in the problem. If we need a specific symbol with odd numbers of occurrences, we make the correspond bit 1, otherwise 0.

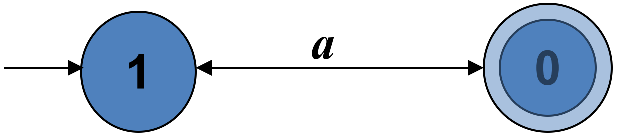
So in the above problem, we have the end state with





we have the start state with

4) Circuit diagram.



5) Grammar. According to the circuit diagram,

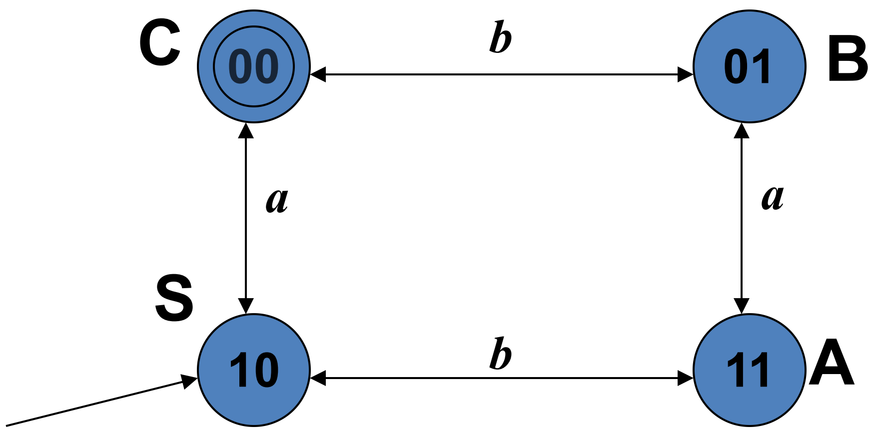
Write a non-terminal corresponding to a state. Let the start state has start terminal.

$$A \rightarrow aB$$

$$B \rightarrow aA \mid \varepsilon$$

- E.g.2. $L=\{\omega \mid \omega\in(a, b)^*$, and the number of a in ω is odd, and that of b is even$\}$

1) Construct the circuit diagram: Tow input bits and four states. Num of a is odd, num of b is even, so start state is 10



2) Grammar

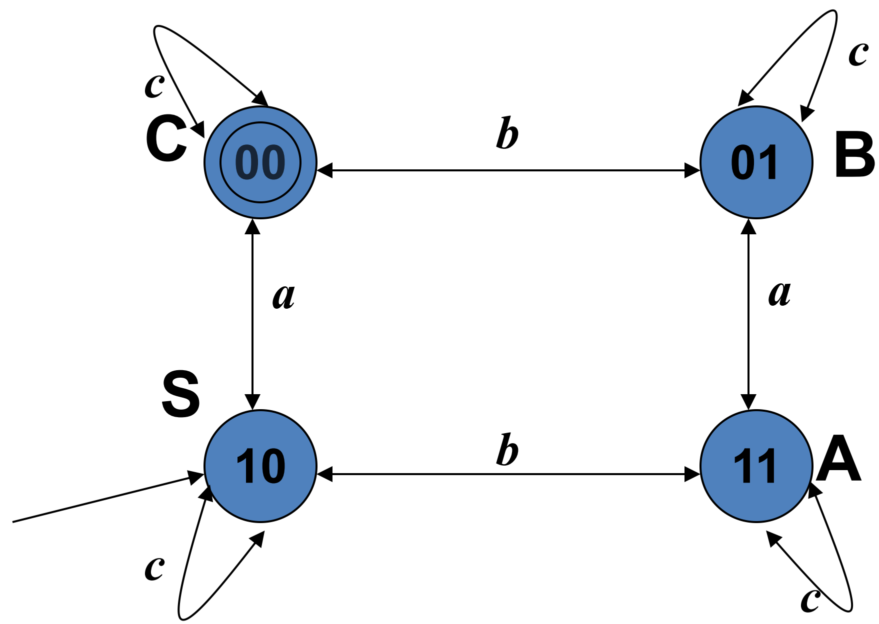
$S \to aC \mid bA$

$A \to aB \mid bS$

$B \to aA \mid bC$

$C \to aS \mid bB \mid \varepsilon$

- E.g.3. $L=\{\omega \mid \omega \in (a, b, c)^*$, and the number of a in ω is odd, and that of b is even$\}$



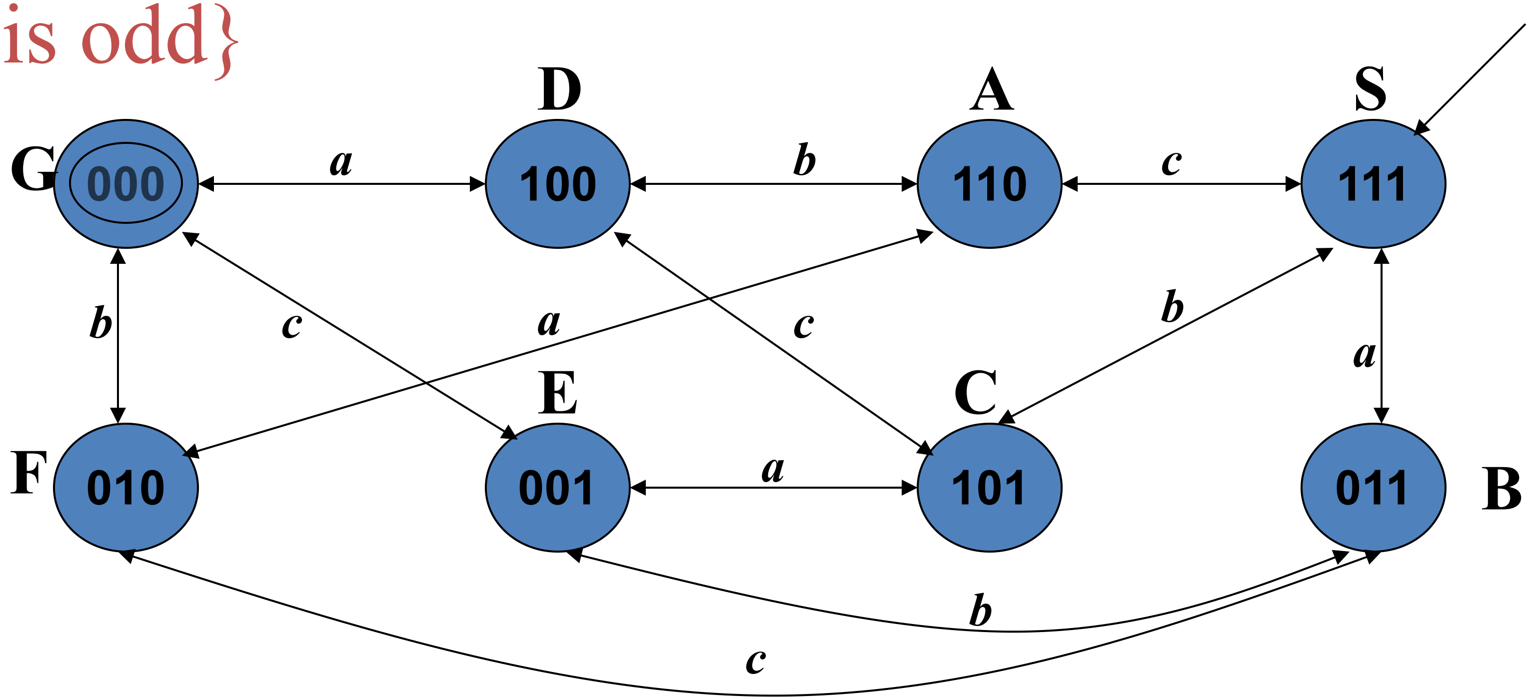
$S \to aC \mid bA \mid cS$

$A \to aB \mid bS \mid cA$

$B \to aA \mid bC \mid cB$

$C \to aS \mid bB \mid cC \mid \varepsilon$

- E.g.4. $L=\{\omega \mid \omega \in (a, b, c)^*$, and the number of a in ω is odd, and that of b is odd, and that of c is odd$\}$



Method 3. Equivalent to 2 sides (两边等价法)

- E.g. $L=\{\omega \mid \omega\in(0, 1)^*$, and the number of 0 in ω is equal to that of 1$\}$

1) S can start with ‘0’ or ‘1’, so $S\rightarrow 0A|1B$
2) A has ‘0’ less 1 than ‘1’, so $A\rightarrow 1S|0AA$
3) B has ‘0’ more 1 than ‘1’, so $B\rightarrow 0S|1BB$

4) Special cases: $S\rightarrow\varepsilon$

From above all, we have:

$S\rightarrow 0A|1B|\varepsilon$

$A\rightarrow 1S|0AA$

$B\rightarrow 0S|1BB$

Method 4. Symmetrical method (对称法)

Embedded Grammar

- E.g.1. $L=\{a^i c b^i \mid i \geq 0\}$

1) Special case1: $S \rightarrow c$
2) Special Case2: $S \rightarrow acb$
3) …
4) …
5) $S \rightarrow aSb|c$

- E.g.2. $L=\{a^{mi} c b^{ni} \mid i \geq 0\}$

1) Special case1: $S \rightarrow c$
2) Special Case2: $S \rightarrow a^m c b^n$
3) …
4) …
5) $S \rightarrow a^m S b^n \mid c$

- E.g.3. $L=\{\omega c \omega^R \mid \omega \in (a, b)^* \}$

1) Special case1: $S \to c$
2) Special Case2: $S \to abcba$
3) …
4) …
5) $S \to aSa \mid bSb \mid c$

Method 5. Hybrid (状态转移图方法)

- E.g.1. L={能够被2整除的正整数的二进制数}
- Exercise. L={能够被5整除的正整数的二进制数}

提示：

1. 考虑余数变化规律，构造状态转移图；
2. 将状态转移图转化为左（右）线性文法

Method 6. Hybrid (混合法)

- E.g.1. $L=\{a^i b^j \mid i \geq j \geq 1\}$

- We can get $a^i b^j = a^{i-j} a^j b^j =$
  $\boxed{a^k}\ \boxed{a^j b^j}$

- Then $S \to AB$

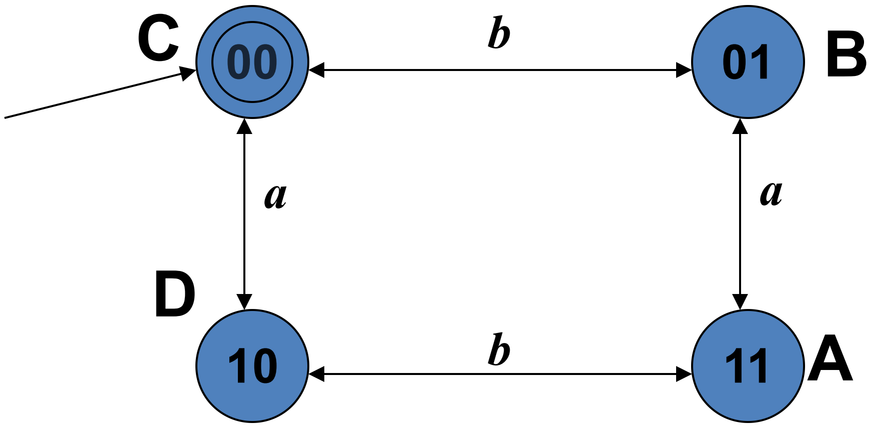
  And $A \to aA|a$

  And $B \to aBb|ab$

- Exercise. $L=\{\omega \mid \omega \in (a, b)^*$, ω starts with a, ends with b, and the number of a is odd, the number of b is odd$\}$

- Way 1
- S $\rightarrow a\text{H}b$

H is a string of a and b, in which the number of a is even, and the number of b is even too.



$H \to C$

$C \to b\mathrm{B} \mid a\mathrm{D} \mid \varepsilon$

$A \to a\mathrm{B} \mid b\mathrm{D}$

$B \to a\mathrm{A} \mid b\mathrm{C}$

$D \to a\mathrm{C} \mid b\mathrm{A}$

- Then

$\mathrm{S} \to a\mathrm{C}b$

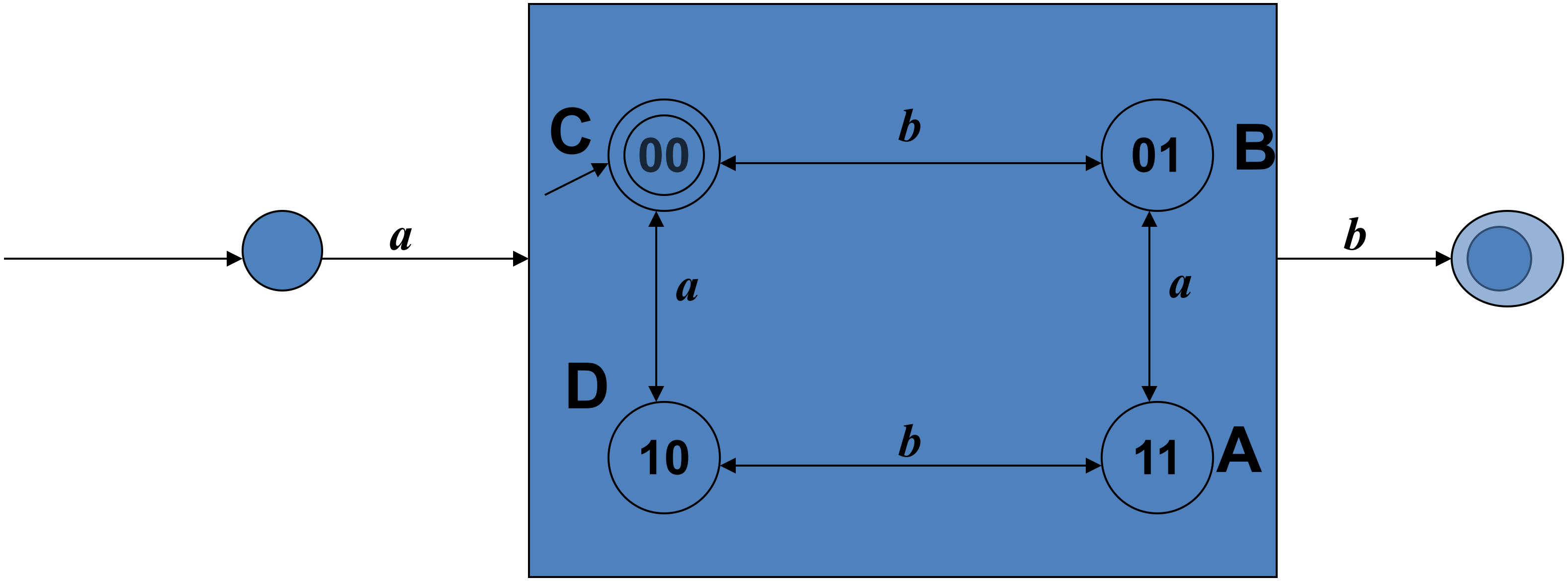
$\mathrm{C} \to b\mathrm{B} \mid a\mathrm{D} \mid \varepsilon$

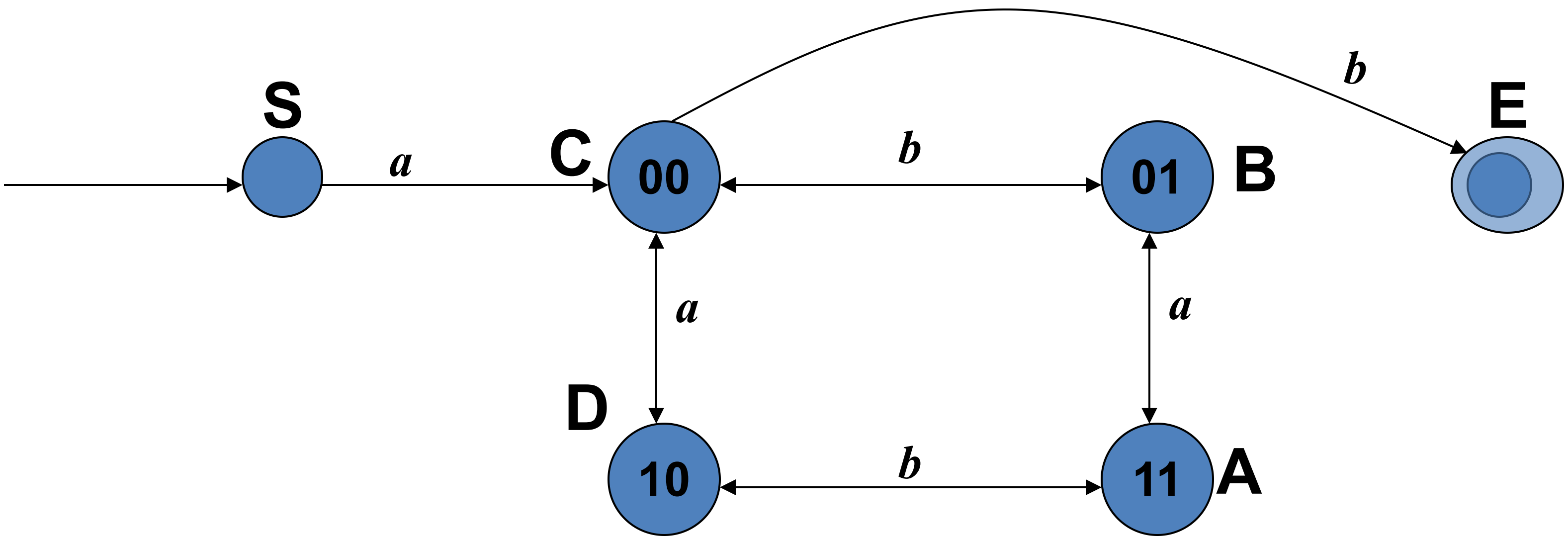
$\mathrm{A} \to a\mathrm{B} \mid b\mathrm{D}$

$\mathrm{B} \to a\mathrm{A} \mid b\mathrm{C}$

$\mathrm{D} \to a\mathrm{C} \mid b\mathrm{A}$

- Way 2.




S
a
C
00
b
01
B
b
E
a
a
D
10
b
11
A

$S \to aC$

$C \to bB \mid aD \mid bE$

$A \to aB \mid bD$

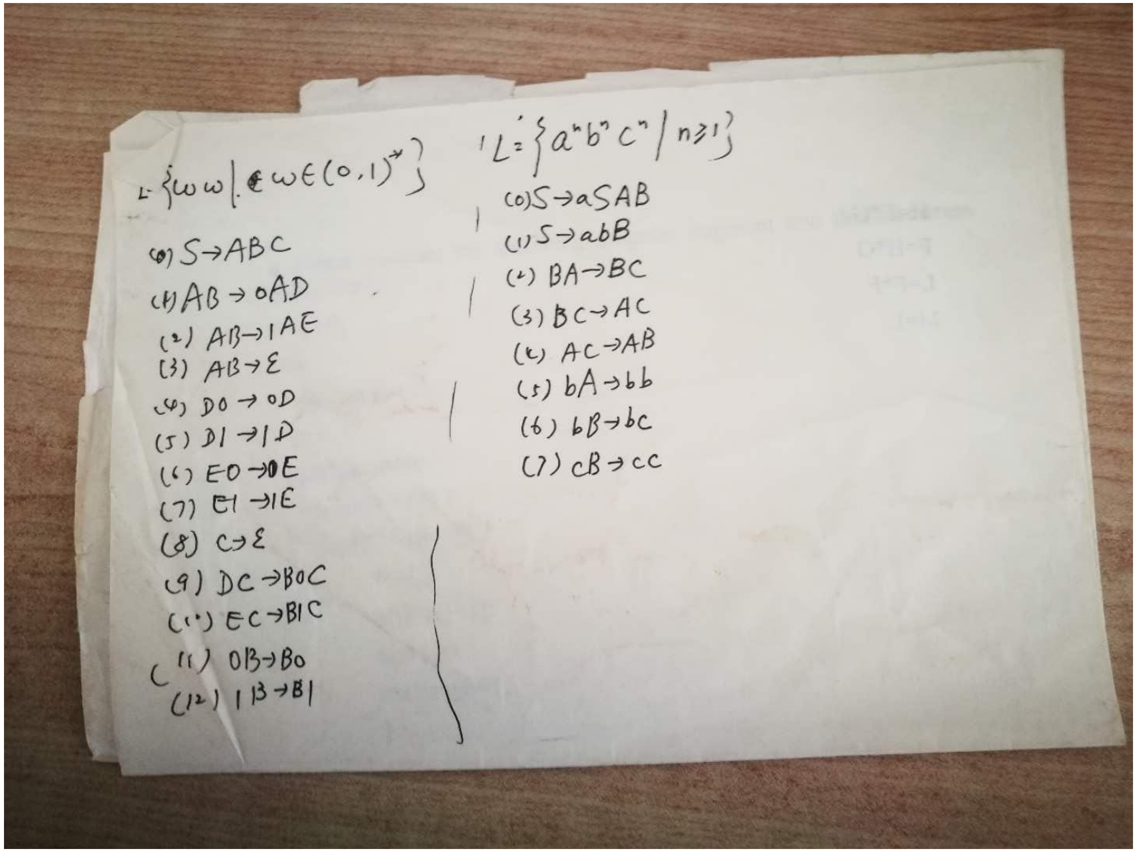
$B \to aA \mid bC$

$D \to aC \mid bA$

$E \to \varepsilon$

Method 7. (非上下文无关文法)

- $L=\{\omega\omega \mid \omega\in(a, b)^*\}$
- Exercise. $L=\{a^m c^m b^m\}$



$L=\{\omega\omega \mid \omega\in(0,1)^*\}$

(0) $S\to ABC$
(1) $AB\to 0AD$
(2) $AB\to 1AE$
(3) $AB\to \varepsilon$
(4) $D0\to 0D$
(5) $D1\to 1D$
(6) $E0\to 0E$
(7) $E1\to 1E$
(8) $C\to \varepsilon$
(9) $DC\to B0C$
(10) $EC\to B1C$
(11) $0B\to B0$
(12) $1B\to B1$

$L=\{a^n b^n c^n \mid n\geq 1\}$

(0) $S\to aSAB$
(1) $S\to abB$
(2) $BA\to BC$
(3) $BC\to AC$
(4) $AC\to AB$
(5) $bA\to bb$
(6) $bB\to bc$
(7) $cB\to cc$

Exercises

- a) $\{\omega \mid \omega \in (a,b,c)^*$ and the numbers of a' s and b' s and c' s occurred in ω are **even**$\}$

- b) $\{a^i b^j \mid \boldsymbol{i \geq (2j+1)}$ and $j \geq 0\}$

- c) $\{\omega \mid \omega \in (a,b,c)^*$ and the numbers of a' s and b' s occurred in ω are **odd**$\}$

- d) $\{a^i b^j \mid \boldsymbol{i \geq (j+1)}$ and $j \geq 0\}$

- e) $\{\omega \mid \omega \in (a,b,c)^*$, ω is **lead by *a*** and the numbers of a' s and b' s occurred in ω are **even** $\}$

- f) $\{a^{2i}b^{2j} | \boldsymbol{j \geq i \geq 1}\}$

- g) $\{\omega |\ \omega \in (a,b,c)^*$ and ω **starts with *a* and ends with *b*, the numbers of *a*' s and *c*' s occurred in** ω are **even**$\}$

- h) $\{a^ib^jc^k | \boldsymbol{j \geq (i+k+1)}$ and $\boldsymbol{i \geq 0,\ k \geq 1}\}$

- i) {ω| ω∈(*a*,*b*,*c*)* and **the numbers of *a*' s , *b*' s ,*c*' s occurred in** ω are all **even**}

- j) {ω| ω∈(*a*,*b*,*c*)* and ω starts with *a* or *b*, ends with *c*, and the numbers of *a*' s and *b*' s and *c*' s occurred in ω are **even**}

- k) $a^{2i\text{-}1}b^{2j\text{-}1}c^{2k\text{-}1}$ $(i\geq 1, j\geq i+k, k\geq 1)$

END