



CHƯƠNG TRÌNH DỊCH

BÀI 8: PHÂN TÍCH VĂN PHẠM BẰNG TOP-DOWN



Nội dung

1. Ý tưởng & thuật toán
2. Ví dụ minh họa
3. Cài đặt top-down đơn giản
 - Cấu trúc một luật văn phạm
 - Cấu trúc một suy diễn trực tiếp
 - Máy phân tích: các hàm hỗ trợ
 - Máy phân tích: các hàm chính
4. Đánh giá về top-down
5. Bài tập



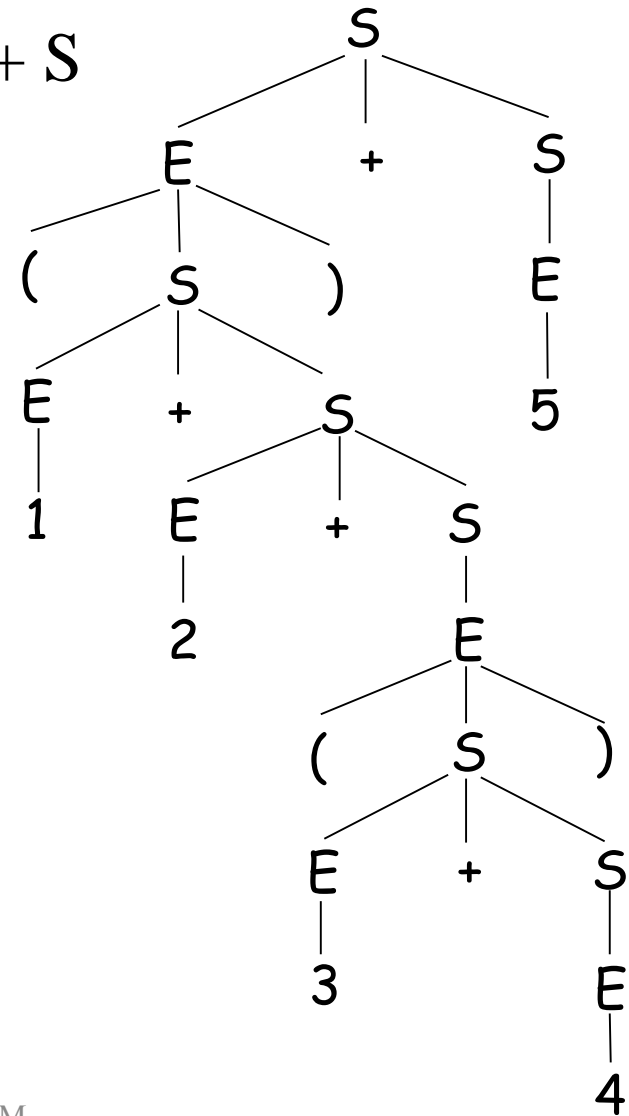
Phần 1

Ý tưởng & thuật toán



Top-down: ý tưởng

$S \Rightarrow E + S \Rightarrow (S) + S \Rightarrow (E + S) + S$
 $\Rightarrow (1 + S) + S$
 $\Rightarrow (1 + E + S) + S$
 $\Rightarrow (1 + 2 + S) + S$
 $\Rightarrow (1 + 2 + E) + S$
 $\Rightarrow (1 + 2 + (S)) + S$
 $\Rightarrow (1 + 2 + (E + S)) + S$
 $\Rightarrow (1 + 2 + (3 + S)) + S$
 $\Rightarrow (1 + 2 + (3 + E)) + S$
 $\Rightarrow (1 + 2 + (3 + 4)) + S$
 $\Rightarrow (1 + 2 + (3 + 4)) + E$
 $\Rightarrow (1 + 2 + (3 + 4)) + 5$





Top-down: ý tưởng

- **Mục tiêu:** trong số nhiều suy dẫn dạng $S \Rightarrow^* w$, thuật toán sẽ tìm suy dẫn trái (hoặc suy dẫn phải)
- Dò theo quá trình suy dẫn từ $S \Rightarrow w_1 \Rightarrow \dots \Rightarrow w$ bằng cách thử áp dụng các luật văn phạm phù hợp cho đến khi đạt được w
- Sử dụng năng lực tính toán của máy tính trong việc thử sai và quay lui
- Nếu một w_i có chứa nhiều non-terminal thì chỉ cần thử với non-terminal trái nhất



Top-down: thuật toán

1. $A = S$
2. Với một chuỗi A đạt được trong quá trình suy dẫn:
 - Nếu $A = w$:
 - Kết luận: quá trình tìm kiếm thành công
 - Lưu lại kết quả (chuỗi biến đổi từ đầu để được A)
 - Kết thúc ngay lập tức quá trình tìm kiếm
 - Nếu $A \neq w$: tìm kí hiệu trung gian trái nhất x
 - Không tìm được x thì dừng bước 2, trở lại hàm gọi
 - Duyệt tất cả các luật sinh dạng $x \rightarrow \alpha$
 - Áp dụng luật đó trên A (ở vị trí x), ta được A'
 - Thử bước 2 với chuỗi $A = A'$



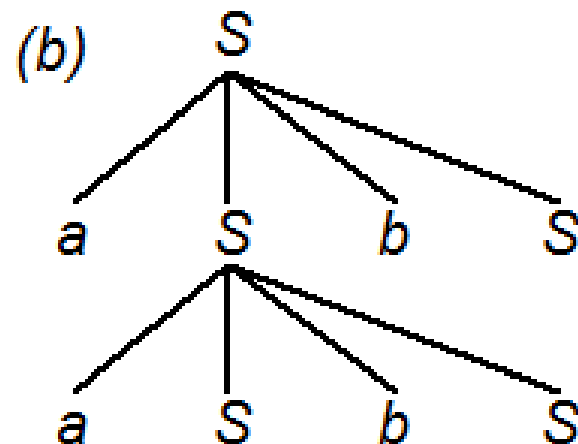
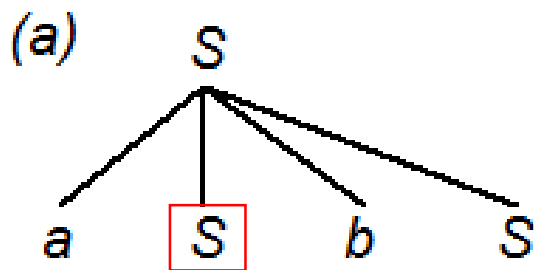
Phần 2

Ví dụ minh họa



Top-down: ví dụ

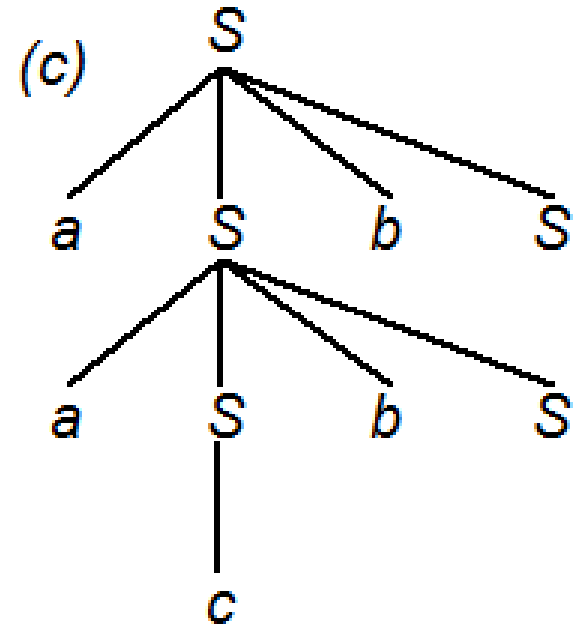
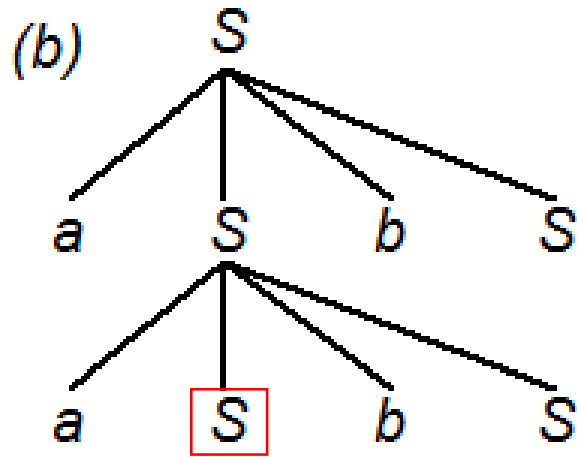
Phân tích $w = aacbc$ với tập luật $S \rightarrow aSbS \mid aS \mid c$



1. Xét $A = aSbS$
2. Tìm được kí hiệu S thứ 2 trong A là non-terminal
3. Thử áp dụng luật $S \rightarrow aSbS$ được $A' = aaSbSbS$



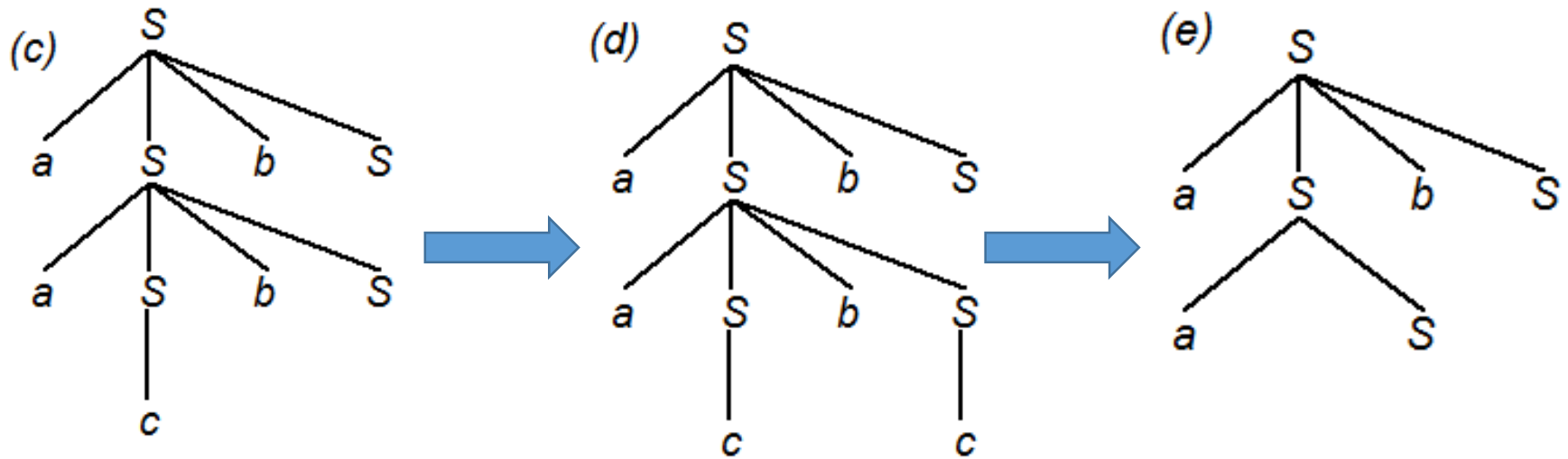
Top-down: ví dụ



1. Xét $A = aaSbSbS$
2. Tìm được kí hiệu S thứ 3 trong A là non-terminal
 1. Thử áp dụng luật $S \rightarrow aSbS$ được $A' = aaaSbSbSbS$
 2. Thử áp dụng luật $S \rightarrow aS$ được $A' = aaaSbSbS$
 3. Thử áp dụng luật $S \rightarrow c$ được $A' = aacbSbS$



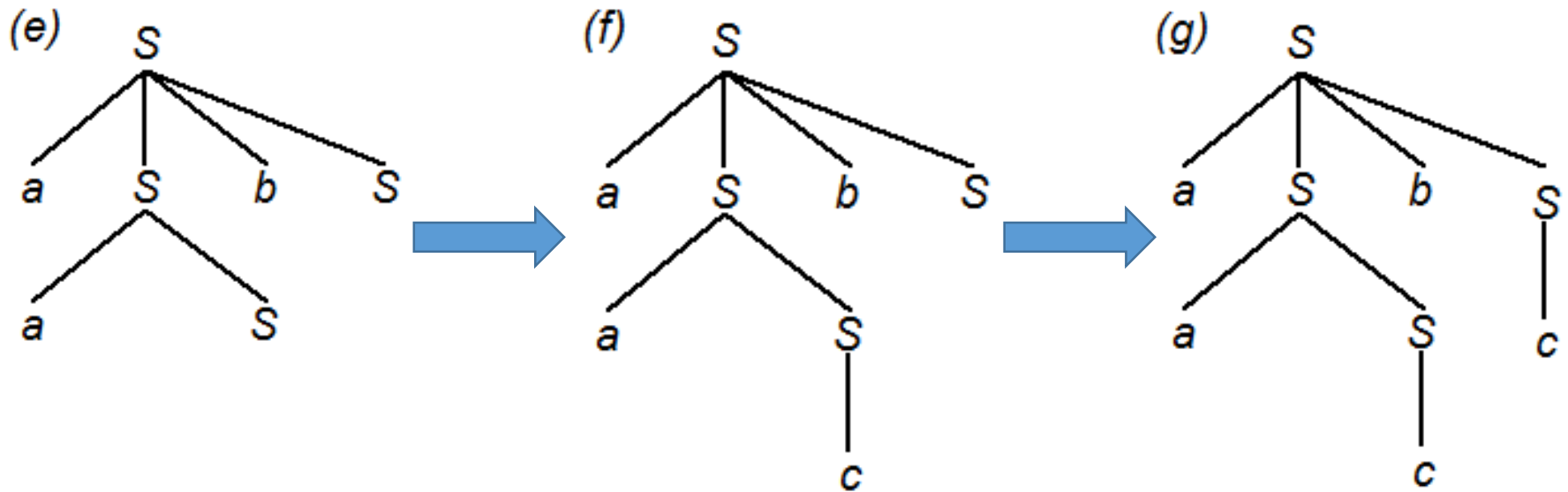
Top-down: ví dụ



- Quá trình thử sai kết luận rằng $A = aSbS$ không thể áp dụng luật $S \rightarrow aSbS$
- Quay lui về đến tình huống ban đầu ở hình (a)
- Thử phương án tiếp theo $S \rightarrow aS$, được $A' = aaSbS$



Top-down: ví dụ



- Quá trình thử sai tiếp tục và cuối cùng dừng ở phương án được thể hiện ở hình (g)
- Khi nhận được chuỗi $A = w = aacbc$, ngay lập tức thuật toán dừng và trả về quá trình áp dụng luật



Phần 3

Cài đặt top-down đơn giản



Cấu trúc một luật văn phạm

```
class Rule {  
    public string left, right;  
    public Rule(string l, string r) {  
        left = l;  
        right = r;  
    }  
    public string ToFineString() {  
        string s = left + " -->";  
        for (int i = 0; i < right.Length; i++)  
            s += " " + right[i];  
        return s;  
    }  
}
```

Cấu trúc một suy diễn trực tiếp



```
class Step {  
    public int ruleNumber, position;  
    public Step(int r, int p) {  
        ruleNumber = r;  
        position = p;  
    }  
}
```

Giải thích:

- Biến ruleNumber lưu số thứ tự của luật sẽ được dùng
- Biến position lưu vị trí sẽ áp dụng luật đó



Máy phân tích: các hàm hỗ trợ

```
class PTTD {
    public List<Rule> rules = new List<Rule>();
    public List<Step> steps;
    string w = null;
    // thêm luật left --> right vào tập luật
    public void AddRule(string left, string right) {
        rules.Add(new Rule(left, right));
    }
    public void PrintAllRules() {
        Console.WriteLine("<bo luat van pham>");
        foreach (Rule r in rules)
            Console.WriteLine("  " + r.ToFineString());
    }
}
```



Máy phân tích: các hàm hỗ trợ

```
public void PrintSteps() {
    Console.WriteLine("Doan nhan thanh cong sau...");
    string w = "S";
    foreach (Step s in steps) {
        string w0 = DoStep(w, s);
        Console.WriteLine("  {0} => {1} (vi tri...");
        w = w0;
    }
}

string DoStep(string w, Step s) {
    string w1 = w.Substring(0, s.position);
    string w2 = w.Substring(s.position + 1);
    return w1 + rules[s.ruleNumber].right + w2;
}
```




Máy phân tích: các hàm chính

```
public bool Process(string x) {
    steps = new List<Step>();
    w = x;
    return Try("S");
}
// tìm kí hiệu không kết thúc đầu tiên trong s
public int FindNonterminal(string s) {
    for (int i = 0; i < s.Length; i++) {
        if (i >= w.Length) return i;
        if (s[i] != w[i]) return i;
    }
    return -1;
}
```



Máy phân tích: các hàm chính

```
public bool Try(string s) {  
    if (s == w) return true;  
    int n = FindNonterminal(s);  
    if (n != -1)  
        for (int i = 0; i < rules.Count; i++)  
            if (rules[i].left[0] == s[n]) {  
                Step st = new Step(i, n);  
                steps.Add(st);  
                if (Try(DoStep(s, st))) return true;  
                steps.RemoveAt(steps.Count - 1);  
            }  
    return false;  
}
```



Thử nghiệm

```
class Program {  
    public static void Main() {  
        PTTD parser = new PTTD();  
        parser.AddRule("S", "B");  
        parser.AddRule("B", "R");  
        parser.AddRule("B", "(B)");  
        parser.AddRule("R", "E=E");  
        ...  
        parser.PrintAllRules();  
        if (parser.Process("(a=(b+a))"))  
            parser.PrintSteps();  
    }  
}
```



Phần 4

Đánh giá về top-down



Đánh giá về top-down

- Thuật toán duyệt gần như hoàn toàn, không có các bước cắt nhánh, chậm nhưng dễ hiểu
- Thuật toán không phải vạn năng, không làm việc được với các văn phạm có đệ quy trái
 - Lý do là vì không có cắt nhánh phù hợp, dẫn đến việc đi mãi theo chiều sâu mà không quay lui
- Việc cài đặt cắt nhánh là rất khó khăn mặc dù có nhiều gợi ý thú vị:
 - Cắt nhánh khi trong A có terminal không có trong w
 - Cắt nhánh khi số terminal trong A nhiều hơn trong w
 - Sử dụng lại những kết quả đã duyệt cũ



Phần 5

Bài tập



Bài tập

1. Chỉ ra quá trình thực hiện phân tích top-down của chuỗi **(a=(b+a))** thuộc văn phạm G có tập luật:
 - $S \rightarrow B$
 - $B \rightarrow R \mid (B)$
 - $R \rightarrow E = E$
 - $E \rightarrow a \mid b \mid (E + E)$
2. Chỉ ra quá trình thực hiện phân tích top-down của chuỗi **true and not false** với tập luật văn phạm:
 - $E \rightarrow E \text{ and } T \mid T$
 - $T \rightarrow T \text{ or } F \mid F$
 - $F \rightarrow \text{not } F \mid (E) \mid \text{true} \mid \text{false}$



Bài tập

3. Chỉ ra quá trình thực hiện phân tích top-down của chuỗi **abcbcd** thuộc văn phạm G có tập luật:
- $S \rightarrow a A \mid b A$
 - $A \rightarrow c A \mid b A \mid d$
4. Có thể áp dụng thuật toán phân tích top-down cho chuỗi **(5+7)*3** thuộc văn phạm G dưới đây hay không? Chỉ ra quá trình thực hiện nếu có thể
- $E \rightarrow E + T \mid T$
 - $T \rightarrow T * F \mid F$
 - $F \rightarrow (E) \mid \text{số}$