



NGỮ NGHĨA HỌC TÍNH TOÁN

CHƯƠNG 1 – GIỚI THIỆU

NGUYỄN TRỌNG CHỈNH



TRÌNH BÀY

1. MỘT SỐ KHÁI NIỆM VỀ NGHĨA
2. NGỮ NGHĨA HỌC TÍNH TOÁN
3. MỘT SỐ ỨNG DỤNG
4. LOGIC VỊ TÙ BẬC MỘT
5. NGÔN NGỮ PROLOG



5. NGÔN NGỮ PROLOG

5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN



5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN

Nguyên tắc:

- Mỗi ký hiệu phi kết thúc biến đổi thành một vị từ.
- Mỗi luật sản sinh biến đổi thành một luật.
- Mỗi luật sản sinh có về phải là một từ biến đổi thành sự kiện.
- Trong mỗi luật, danh sách từ bên về trái bằng tất cả danh sách từ bên về phải nối lại theo đúng thứ tự.



5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN

Ví dụ: Cho văn phạm G có tập luật sản sinh P như bên dưới.

- | | | |
|---------------------------|------------------------------|-------------------------|
| 1. $S \rightarrow NP\ VP$ | 2. $NP \rightarrow NN\ ADJP$ | 3. $NP \rightarrow NN$ |
| 4. $ADJP \rightarrow JJ$ | 5. $VP \rightarrow VB\ NP$ | 6. $NN \rightarrow cây$ |
| 7. $NN \rightarrow hoa$ | 8. $JJ \rightarrow to$ | 9. $JJ \rightarrow nhỏ$ |
| 10. $VB \rightarrow có$ | | |

Dùng Prolog, hãy kiểm tra các câu sau đúng văn phạm G không.

- | | |
|---------------------------------|------------------------------------|
| 1) “ <i>cây có hoa</i> ” | 2) “ <i>cây to cây nhỏ</i> ” |
| 3) “ <i>cây nhỏ có hoa to</i> ” | 4) “ <i>có cây to có cây nhỏ</i> ” |



5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN

```
1 nn ( [cây] ) .  
2 nn ( [hoa] ) .  
3 jj ( [nhỏ] ) .  
4 jj ( [tô] ) .  
5 vb ( [có] ) .  
6 s (W) :- np (Wn) , vp (Wv) , append (Wn, Wv, W) .  
7 np (W) :- nn (Wn) , adjp (Wa) , append (Wn, Wa, W) .  
8 np (W) :- nn (W) .  
9 adjp (W) :- jj (W) .  
10 vp (W) :- vb (Wv) , np (Wn) , append (Wv, Wn, W) .
```



5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN

Kiểm tra cú pháp trên Prolog:

1) “cây có hoa”

Query: s([cây, có, hoa]).

Kết quả: **true.**

2) “cây to cây nhỏ”

Query: s([cây, to, cây, nhỏ]).

Kết quả: **false.**

3) “cây nhỏ có hoa to”

Query: s([cây, nhỏ, có, hoa, to]).

Kết quả: **true.**

4) “cây to có cây nhỏ”

Query: s([có, cây, to, có, cây, nhỏ]).

Kết quả: **false.**



5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN

Nhược điểm của cách cài đặt này:

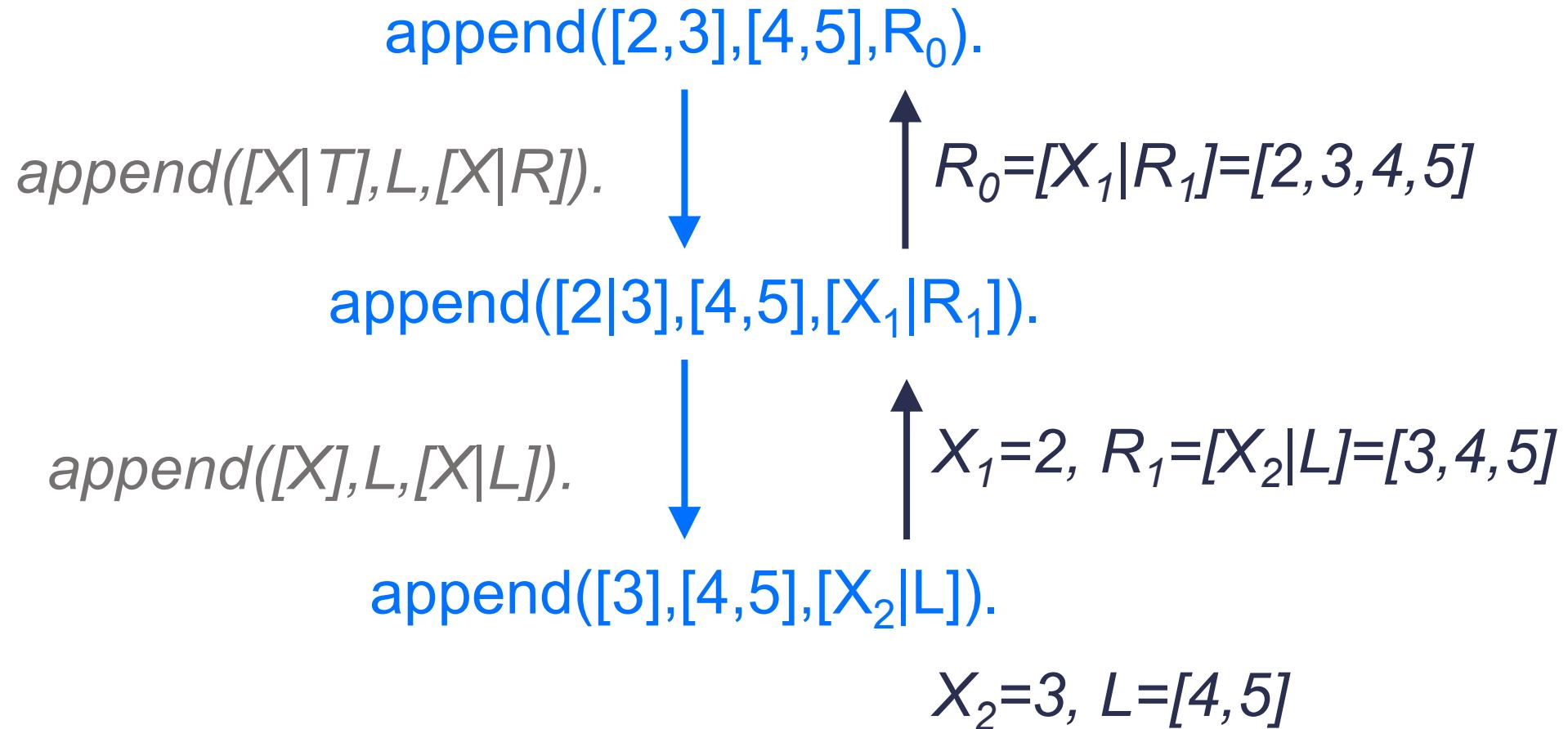
- Kiểm tra ràng buộc chuỗi ở về trái luật phải chính là các chuỗi ở về phải nối tiếp nhau theo thứ tự bằng vị từ append.
- Vị từ append nối hai chuỗi với độ phức tạp thuật toán $O(n)$

```
1  append( [ ] , L , L ) .  
2  append( [ X ] , L , [ X | L ] ) .  
3  append( [ X | T ] , L , [ X | R ] ) :-append( T , L , R ) .
```



5.7 CÀI ĐẶT THUẬT TOÁN TOP-DOWN

Ví dụ: $\text{append}([2,3],[4,5],R)$.





5. NGÔN NGỮ PROLOG

5.8 DANH SÁCH HIỆU



5.8 DANH SÁCH HIỆU

Danh sách hiệu (difference list):

- Biểu diễn theo cấu trúc $[a_1, a_2, \dots, a_k | T] - T$.
- Nối hai danh sách bằng phép hợp nhất \Rightarrow độ phức tạp $O(1)$.

Ví dụ: Nối hai danh sách $A=[1,2,3,4]$ và $B=[5,6,7]$

Biểu diễn A theo danh sách hiệu: $[1,2,3,4|X] - X$.

Biểu diễn B theo danh sách hiệu: $[5,6,7|Y] - Y$.

Nối B vào A như sau: $X=[5,6,7|Y], Y=[]$.



5.8 DANH SÁCH HIỆU

Ví dụ app sử dụng danh sách hiệu:

app(A-TA,B-TB,C-TC):-TA=B, C=A, TC=TB.

Viết lại:

app(A-B,B-T,A-T).

Để nối danh sách B vào danh sách A, kết quả là R:

- 1 A=[1, 2, 3, 4 | X]-X,
- 2 B=[5, 6, 7 | Y]-Y,
- 3 app (A, B, R- []).



5.8 DANH SÁCH HIỆU

Ví dụ: viết chương trình dịch chuyển các phần tử của một danh sách sang trái một vị trí sử dụng danh sách hiệu.

1 app (A-B, B-C, A-C) .

2 leftmove ([X | T] -T, [X | T] -T) .

3 leftmove ([X | T] -S, R-SR) :-app (T-S, [X | L] -L, R-SR) .

Query:

leftmove ([1,2,3,4,5|T]-T,R-[]).



Kết quả:

R=[2,3,4,5,1]



5.8 DANH SÁCH HIỆU

Phân tích cú pháp với danh sách hiệu:

- Biểu diễn luật sản sinh có về phải toàn ký hiệu non-terminal:
Head(W-T):- Body₁(W-B₂), Body₂(B₂-B₃), ..., Body_n(B_{n-1}-T).
- Biểu diễn luật sản sinh có về phải là ký hiệu terminal
Head(W-T):-Body([tù|T]-T).



5.8 DANH SÁCH HIỆU

Ví dụ: Cho văn phạm G có tập luật sản sinh P như bên dưới.

- | | | |
|---------------------------|------------------------------|-------------------------|
| 1. $S \rightarrow NP\ VP$ | 2. $NP \rightarrow NN\ ADJP$ | 3. $NP \rightarrow NN$ |
| 4. $ADJP \rightarrow JJ$ | 5. $VP \rightarrow VB\ NP$ | 6. $NN \rightarrow cây$ |
| 7. $NN \rightarrow hoa$ | 8. $JJ \rightarrow to$ | 9. $JJ \rightarrow nhỏ$ |
| 10. $VB \rightarrow có$ | | |

Dùng danh sách hiệu để biểu diễn các luật sản sinh P trong Prolog.



5.8 DANH SÁCH HIỆU

- 1 nn ([cây | T] -T) .
- 2 nn ([hoa | T] -T) .
- 3 jj ([nhỏ | T] -T) .
- 4 jj ([to | T] -T) .
- 5 vb ([có | T] -T) .
- 6 s (W-T) :- np (W-Wv) , vp (Wv-T) .
- 7 np (W-T) :- nn (W-Wa) , adjp (Wa-T) .
- 8 np (W-T) :- nn (W-T) .
- 9 adjp (W-T) :- jj (W-T) .
- 10 vp (W-T) :- vb (W-Wn) , np (Wn-T) .

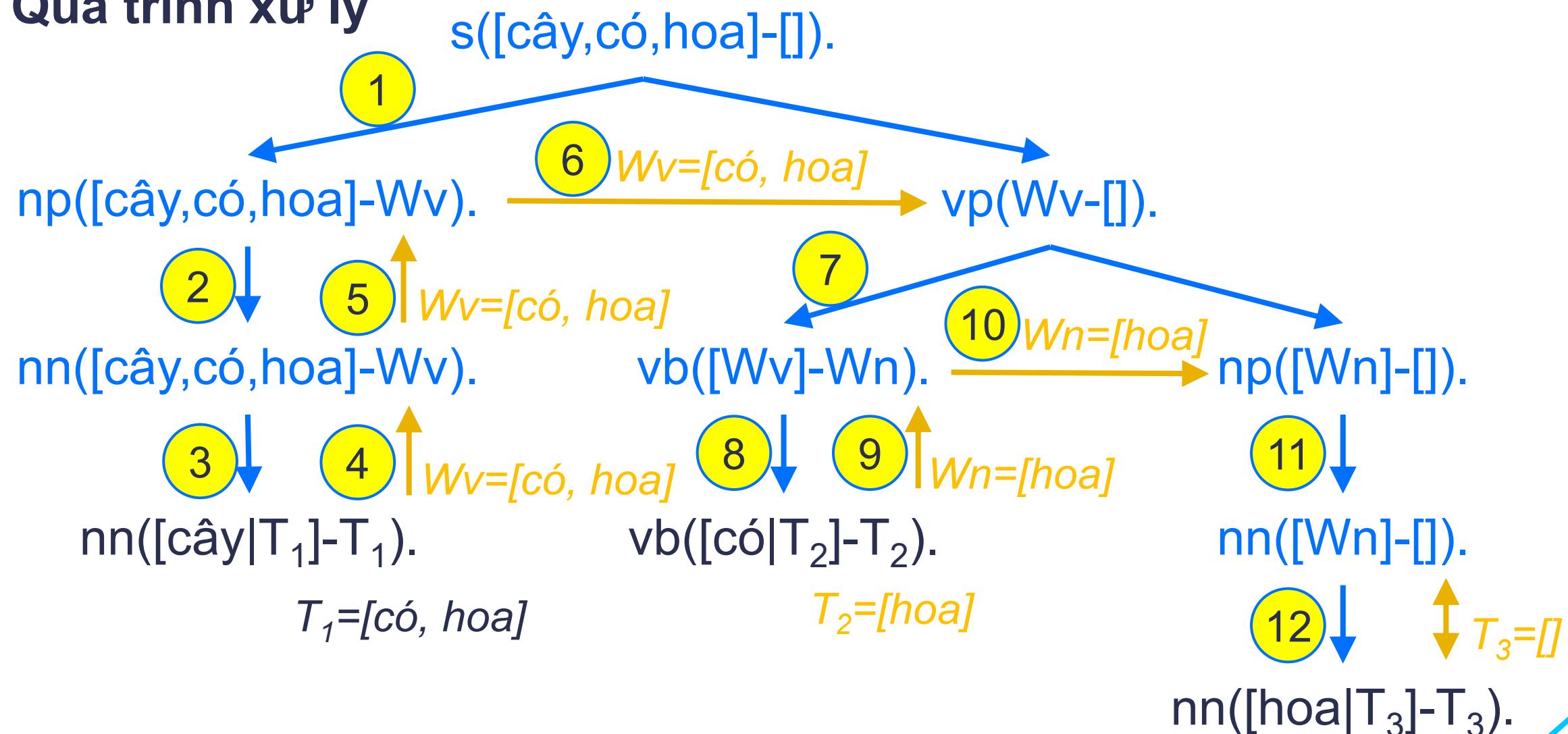
Query:

s ([cây , có , hoa] - []) .



5.8 DANH SÁCH HIỆU

Quá trình xử lý





5. NGÔN NGỮ PROLOG

5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Văn phạm mệnh đề xác định (Definite Clause Grammar – DCG):

- Là một tập tiên đề (axiom) cho phép xác định tính đúng đắn của một phát biểu và quá trình xác định đó.
- Được áp dụng cho các bài toán có dạng chứng minh định lý.

Ví dụ:

- ✓ Kiểm tra cú pháp của các câu lệnh trong ngôn ngữ lập trình.
- ✓ Phân tích cú pháp theo văn phạm phi ngữ cảnh.



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Cú pháp rút gọn biểu diễn các mệnh đề theo DCG:

$$L \rightarrow R_1, R_2, \dots, R_n.$$

- L là vế trái của luật dẫn. L là một ký hiệu phi kết thúc.
- R_1, R_2, \dots, R_n là phần thân (body) của luật dẫn. Mỗi R_i , $i=1..n$, có thể là một ký hiệu phi kết thúc (non-terminal) hoặc một danh sách ký hiệu kết thúc (terminal).
- \rightarrow là ký hiệu phân tách vế trái và phần thân của luật dẫn.
- “.” là dấu kết thúc một mệnh đề trong Prolog.



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Ví dụ: biểu diễn văn phạm CFG sau theo DCG

- | | | |
|---------------------------|------------------------------|-------------------------|
| 1. $S \rightarrow NP\ VP$ | 2. $NP \rightarrow NN\ ADJP$ | 3. $NP \rightarrow NN$ |
| 4. $ADJP \rightarrow JJ$ | 5. $VP \rightarrow VB\ NP$ | 6. $NN \rightarrow cây$ |
| 7. $NN \rightarrow hoa$ | 8. $JJ \rightarrow to$ | 9. $JJ \rightarrow nhỏ$ |
| 10. $VB \rightarrow có$ | | |



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Kết quả biểu diễn theo DCG

s --> np, vp.

np --> nn, adjp.

np --> nn.

adjp --> jj.

vp --> vb, np.

nn --> [cây] | [hoa].

jj --> [nhỏ] | [to].

vb --> [có].

nn --> [cây].
nn --> [hoa].



nn --> [cây] | [hoa].



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Cú pháp đầy đủ để biểu diễn các tiên đề theo DCG:

$L(argL) \rightarrow R_1(argR_1), R_2(argR_2), \dots, R_n(argR_n), \{clause\}.$

- $argL, argR_1, argR_2, \dots, argR_n$: các tham số phụ của từng thành phần.
- $clause$: một mệnh đề Prolog để xử lý các tham số phụ khi cần.

Cú pháp đầy đủ để biểu diễn các luật có mối liên hệ ràng buộc, chẳng hạn văn phạm cảm ngữ cảnh (Context-sensitive Grammar – CSG).



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Ví dụ: Cho văn phạm G bên dưới sản sinh hai câu như sau.

- a) Leo núi.
- b) Lội suối.

- | | | | |
|--------------|--------------|-------------|-------------|
| 1. S → VP NP | 2. NP → NN | 3. VP → VB | |
| 4. NN → núi | 5. NN → suối | 6. VB → leo | 7. VB → lội |

Khi đó, hai câu sau cũng thỏa văn phạm G nhưng không đúng

- a) Lội núi.
- b) Leo suối.



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Văn phạm G

- | | | | |
|--------------|--------------|-------------|-------------|
| 1. S → VP NP | 2. NP → NN | 3. VP → VB | |
| 4. NN → núi | 5. NN → suối | 6. VB → leo | 7. VB → lội |

Chưa thể hiện ràng buộc ngữ nghĩa:

- Động từ “leo” đi với danh từ “núi” nhưng không đi với danh từ “suối”.
- Động từ “lội” đi với danh từ “suối” nhưng không đi với danh từ “núi”.



5.9 VĂN PHẠM MỆNH ĐỀ XÁC ĐỊNH

Thêm tham số phụ O làm ràng buộc giữa động từ và danh từ.

Văn phạm DCG:

s --> vp(O), np(O).

np(O) --> nn(O).

vp(O) --> vb(O).

nn(nui_non) --> [núi].

nn(song_suoi) --> [suối].

vb(nui_non) --> [leo].

vb(song_suoi) --> [lội].

```
?- s([lội, suối], []).  
true.
```

```
?- s([leo, núi], []).  
true.
```

```
?- s([leo, suối], []).  
false.
```

```
?- s([lội, núi], []).  
false.
```



5. NGÔN NGỮ PROLOG

5.10 XỬ LÝ ĐỆ QUY TRÁI



5.10 XỬ LÝ ĐỆ QUY TRÁI

Vấn đề đệ quy trái ở Prolog:

- Tìm kiếm lời giải theo chiến lược Top-down.
- Văn phạm DCG có thể chứa các quy tắc sản sinh có dạng:

$$X \rightarrow X \alpha \beta$$

⇒ **Khả năng lặp vô tận khi tìm lời giải**



5.10 XỬ LÝ ĐỆ QUY TRÁI

Ví dụ: cho văn phạm DCG như bên dưới. Kiểm tra cụm từ “vở sạch , chữ đẹp” có thỏa văn phạm đã cho hay không?

np --> np, cc, np.

np --> nn, jj.

nn --> [chữ]||[vở].

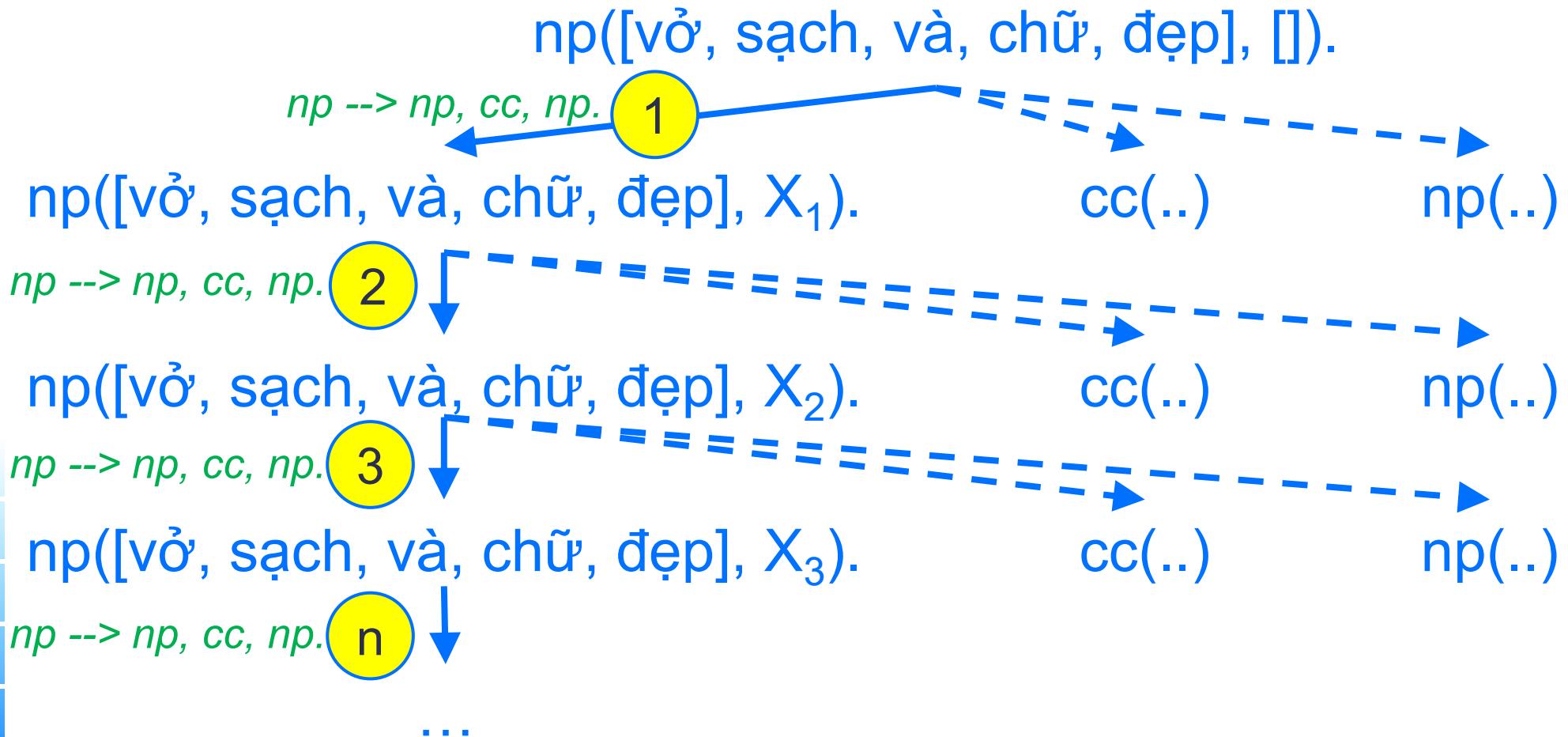
jj --> [đẹp]||[sạch].

cc --> [và].



5.10 XỬ LÝ ĐỆ QUY TRÁI

Quá trình tìm lời giải:





5.10 XỬ LÝ ĐỆ QUY TRÁI

Khử đệ quy trái ở Prolog:

- Xác định các luật đệ quy trái, có dạng $X \rightarrow X \alpha \beta$
- Xác định số cấp đệ quy tối đa cần xử lý.
- Thêm các luật theo từng cấp đệ quy.



5.10 XỬ LÝ ĐỆ QUY TRÁI

Ví dụ: khử đệ quy trái cho văn phạm DCG bên dưới

np --> np, cc, np.

np --> nn, jj.

nn --> [chữ] | [vở].

jj --> [đẹp] | [sạch].

cc --> [và].



5.10 XỬ LÝ ĐỆ QUY TRÁI

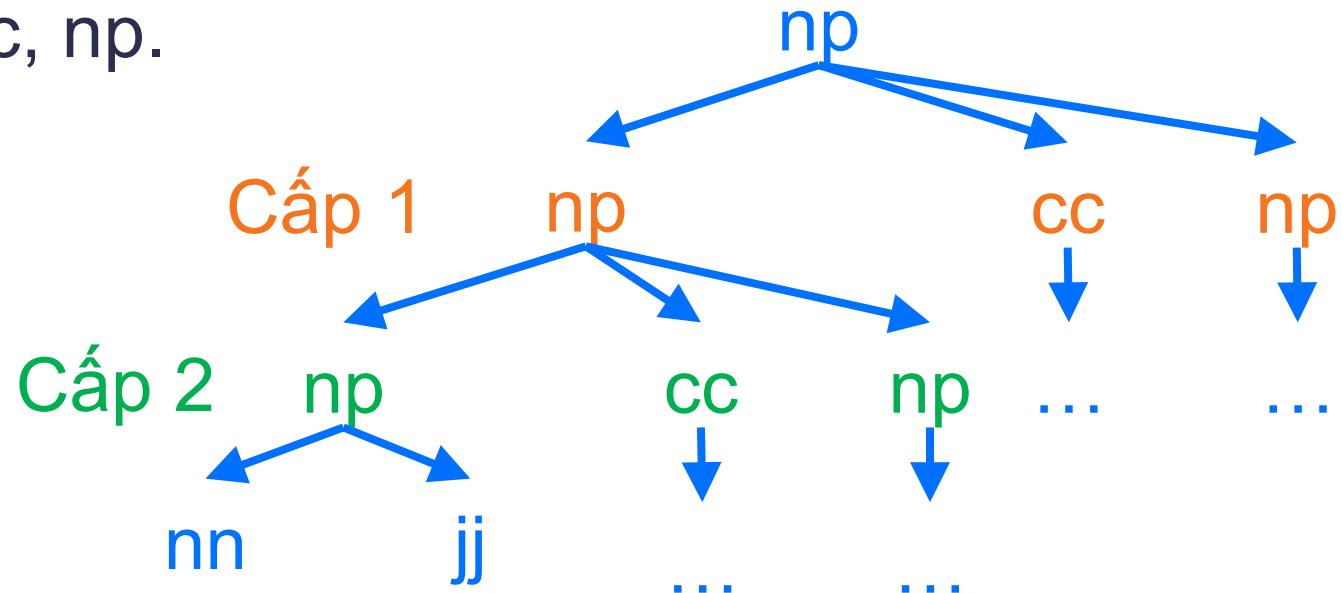
Luật đệ quy trái: $np \rightarrow np, cc, np$.

Giả sử số cấp tối đa là 2.

Các luật có vẽ trái np là

$np \rightarrow np, cc, np$.

$np \rightarrow nn, jj$.



⇒ **Thêm các luật có vẽ trái là np cho 2 cấp là $np1$ và $np2$.**



5.10 XỬ LÝ ĐỆ QUY TRÁI

Tập luật DCG đã khử đệ quy trái:

np --> np1, cc, np.

np --> nn, jj.

np1 --> np2, cc, np.

np1 --> nn, jj.

np2 --> nn, jj.

nn --> [chữ] | [vở].

jj --> [đẹp] | [sạch].

cc --> [và].



5.10 XỬ LÝ ĐỆ QUY TRÁI

Ví dụ: xác định văn phạm DCG để nhận dạng hai câu sau:

- Học bài và làm bài tập.
- Không kinh doanh pháo.



5.10 XỬ LÝ ĐỆ QUY TRÁI

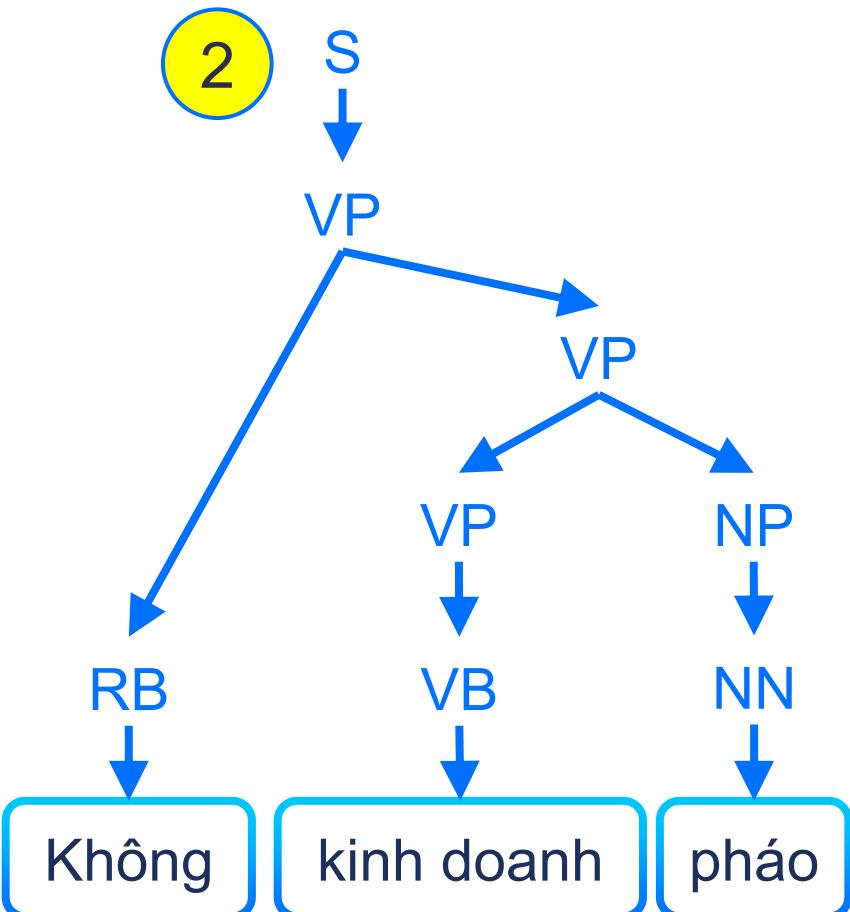
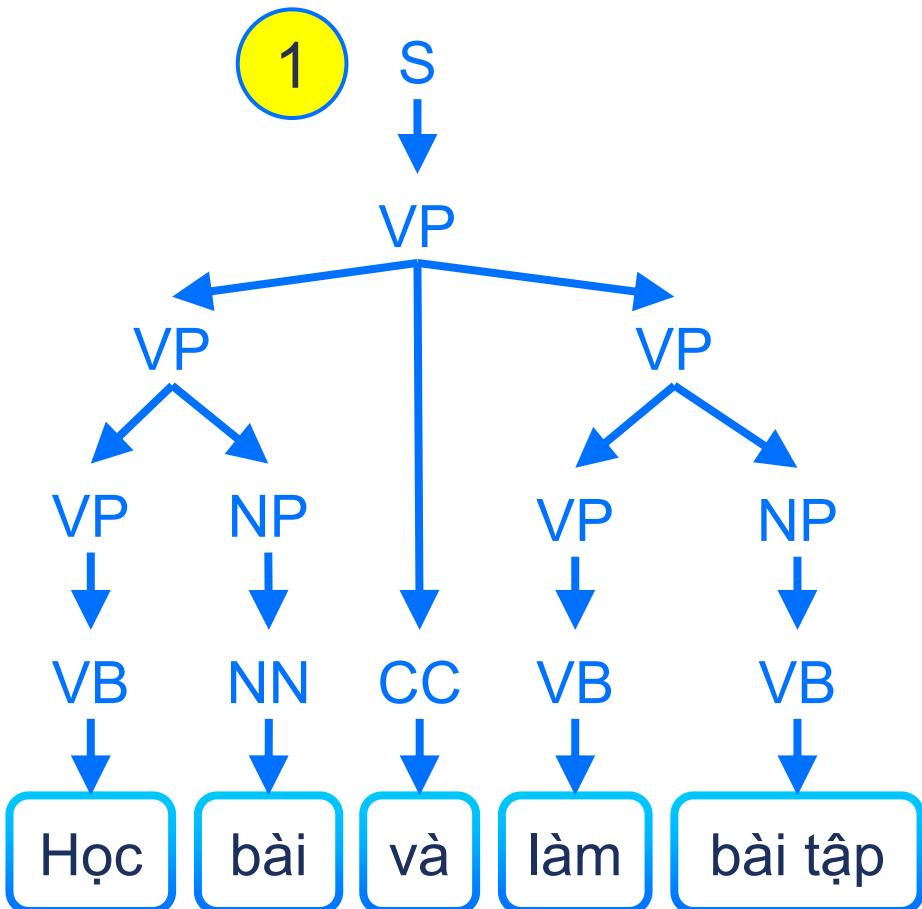
Các từ xuất hiện trong 2 câu đã cho:

- Danh từ chung (nn): bài, bài tập, pháo.
- Động từ (vb): học, kinh doanh, làm.
- Phụ từ (rb): không.
- Liên từ (cc): và.



5.10 XỬ LÝ ĐỆ QUY TRÁI

Kết quả phân tích cú pháp của hai câu





5.10 XỬ LÝ ĐỆ QUY TRÁI

Hợp các luật sản sinh theo hai cây cú pháp 1 và 2:

- | | | |
|----------------|--------------------------------|-----------------------------|
| 1. S → VP | 2. VP → VP CC VP | 3. VP → VP NP |
| 4. VP → VB | 5. VP → RB VP | 6. NP → NN |
| 7. VB → học | 8. VB → làm | 9. VB → kinh doanh |
| 10. NN → bài | 11. NN → bài tập | 12. NN → pháo |
| 13. RB → không | 14. CC → và | |



5.10 XỬ LÝ ĐỆ QUY TRÁI

Chuyển về văn phạm DCG và khử đệ quy trái với hai cấp đệ quy:

s --> vp.

vp2 --> vb.

vp --> vp1, cc, vp.

vp2 --> rb, vp.

vp --> vp1, np.

np --> nn.

vp --> vb.

cc --> [và].

vp --> rb, vp.

vb --> [học] | [làm] | [kinh, doanh].

vp1 --> vp2, cc, vp.

nn --> [bài] | [bài, tập] | [pháo].

vp1 --> vp2, np.

rb --> [không].

vp1 --> vb.

vp1 --> rb, vp.



5. NGÔN NGỮ PROLOG

5.11 TÍNH TOÁN CÂY CÚ PHÁP



5.11 TÍNH TOÁN CÂY CÚ PHÁP

Cây cú pháp (parse) hay cây dẫn xuất (derivation tree): thể hiện quá trình áp dụng các quy tắc sản sinh của văn phạm để sản sinh ra một câu cho trước.

Tính toán cây cú pháp:

- Chọn các quy tắc sản sinh phù hợp.
- Lưu vết các quy tắc đã được sử dụng theo cấu trúc cây.



5.11 TÍNH TOÁN CÂY CÚ PHÁP

Lưu vết các quy tắc đã được sử dụng theo cấu trúc cây.

- Sử dụng tham số phụ trong văn phạm DCG.
- Cập nhật cấu trúc của nút cha từ cấu trúc của các nút con.

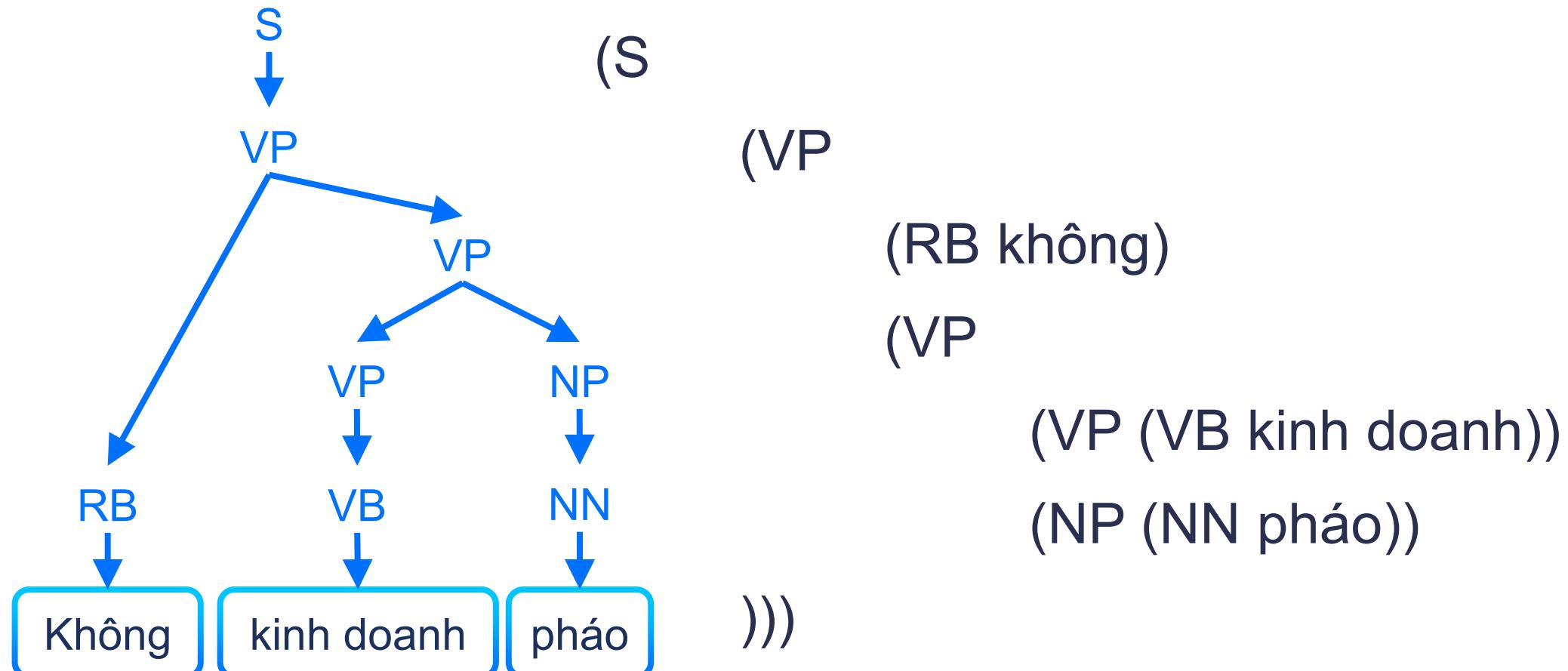
Cấu trúc của các nút:

- Xác định theo dạng dấu ngoặc theo Penn Treebank.
- Dùng dấu “,” để phân tách hai thành tố.
- Thay ký tự in hoa bằng ký tự thường.



5.11 TÍNH TOÁN CÂY CÚ PHÁP

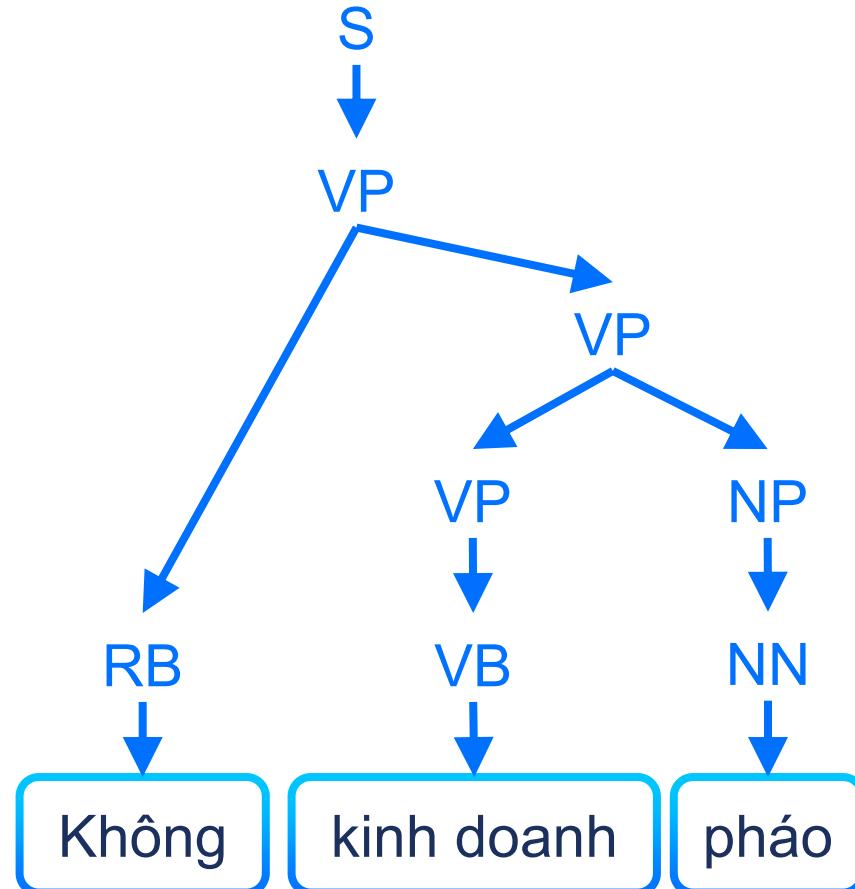
Ví dụ: Cây cú pháp bên dưới theo dạng dấu ngoặc:





5.11 TÍNH TOÁN CÂY CÚ PHÁP

Văn phạm DCG tương ứng (chưa xử lý đệ quy trái):



$S \rightarrow vp.$

$vp \rightarrow rb, vp.$

$vp \rightarrow vp, np.$

$vp \rightarrow vb.$

$np \rightarrow nn.$

$rb \rightarrow [không].$

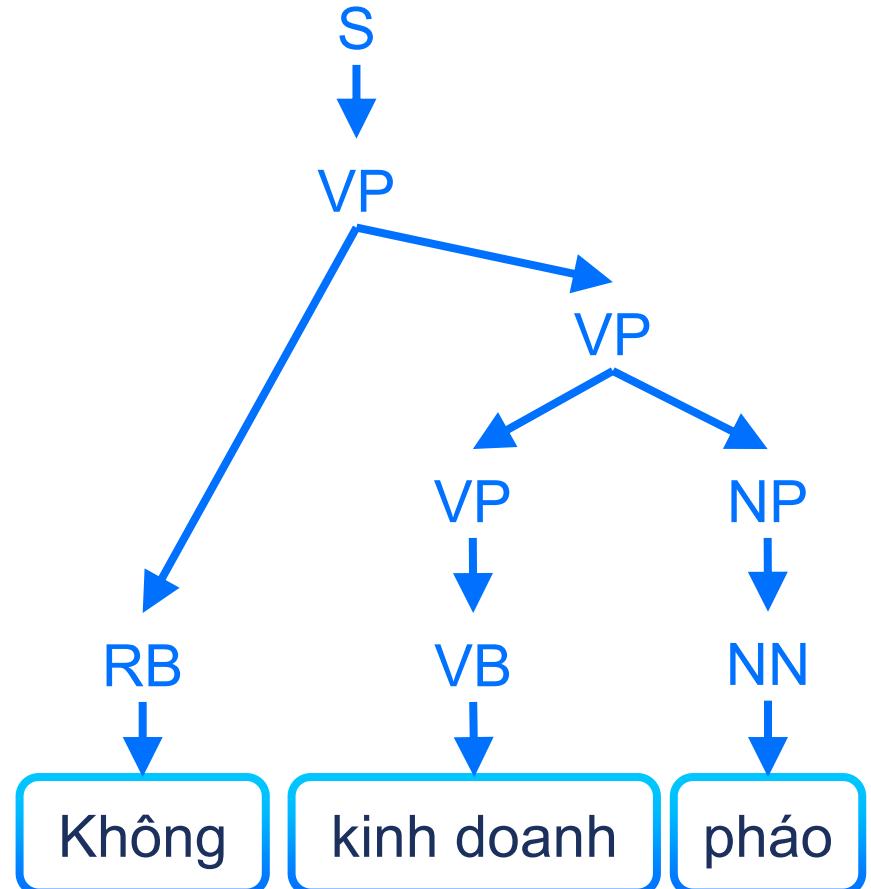
$vb \rightarrow [kinh, doanh].$

$nn \rightarrow [pháo].$



5.11 TÍNH TOÁN CÂY CÚ PHÁP

Thêm tham số phụ lưu vết các quy tắc



$s(s(VP)) \rightarrow vp(VP).$

$vp(vp(RB, VP)) \rightarrow rb(RB), vp(VP).$

$vp(vp(VP, NP)) \rightarrow vp(VP), np(NP).$

$vp(vp(VB)) \rightarrow vb(VB).$

$np(np(NN)) \rightarrow nn(NN).$

$rb(rb(không)) \rightarrow [không].$

$vb(vb(kinh_doanh)) \rightarrow [kinh, doanh].$

$nn(nn(pháo)) \rightarrow [pháo].$



5.11 TÍNH TOÁN CÂY CÚ PHÁP

Kết quả sau khi khử đệ quy trái:

