

TẠO BIỂU DIỄN LUẬN LÝ CHO NGÔN NGỮ TỰ NHIÊN TRONG QUÁ TRÌNH PHÂN TÍCH CÚ PHÁP

Phan Thị Tươi, Nguyễn Chí Hiếu

Trường Đại học Bách khoa – ĐHQG TP.HCM

(Bài nhận ngày 14 tháng 3 năm 2002, hoàn chỉnh sửa chữa ngày 29 tháng 3 năm 2002)

TÓM TẮT: Bài báo sẽ trình bày những ý tưởng cơ bản của lý thuyết ngữ nghĩa, giới thiệu mức ngữ nghĩa không phụ thuộc ngữ cảnh của câu, được gọi là dạng luận lý. Dạng này được sinh ra trực tiếp trong quá trình phân tích cú pháp của câu trong ngôn ngữ tự nhiên. Bài báo cũng trình bày những cải thiện của chúng tôi cho giải thuật phân tích cú pháp biểu đồ từ trên xuống cũng như việc xây dựng văn phạm lưu vết quá trình phân tích cú pháp để sử dụng cho giai đoạn tạo dạng luận lý.

1. ĐẶT VẤN ĐỀ

Trong xử lý ngôn ngữ tự nhiên, người ta luôn mong muốn “nắm bắt” được nghĩa của câu nhập. Mô hình biểu diễn ngữ nghĩa của câu ngôn ngữ tự nhiên là một công cụ quan trọng để biên dịch sang một ngôn ngữ khác.

Hiện nay nghiên cứu về dịch máy, người ta đã tập trung vào phân tích nghĩa của câu chứ không chỉ quan tâm đến phân tích cú pháp, mặc dù phân tích cú pháp vẫn luôn đóng vai trò quan trọng trong quá trình xử lý ngôn ngữ tự nhiên[6].

Để biểu diễn ngữ nghĩa của ngôn ngữ tự nhiên người ta dùng dạng luận lý (logical form). Dạng luận lý là mức biểu diễn ngữ nghĩa mà không phụ thuộc ngữ cảnh [4, 7, 8].

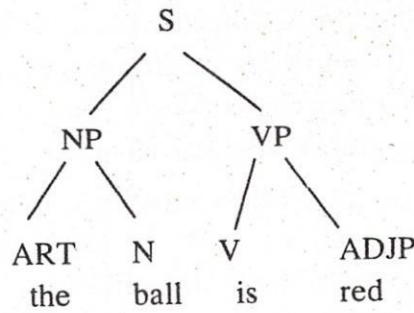
Trong bài báo này, chúng tôi sẽ trình bày ý tưởng của lý thuyết cơ bản về ngữ nghĩa cũng như sự liên hệ giữa cú pháp và ngữ nghĩa mà ứng dụng quan trọng là việc tạo trực tiếp dạng biểu diễn luận lý của ngữ nghĩa từ phân tích cấu trúc cú pháp của câu. Phần cuối bài báo sẽ trình bày một số cải thiện của chúng tôi cho giải thuật phân tích cú pháp biểu đồ từ trên xuống, cũng như việc xây dựng văn phạm lưu vết để sử dụng cho việc tạo dạng luận lý trong quá trình phân tích cú pháp.

2. NGÔN NGỮ DẠNG LUẬN LÝ CƠ BẢN

2.1 Ngữ nghĩa và dạng luận lý

Để lưu giữ nghĩa của câu người ta dùng dạng luận lý. Quá trình ánh xạ một câu vào dạng luận lý được gọi là biểu diễn ngữ nghĩa. Còn quá trình ánh xạ từ dạng luận lý sang ngôn ngữ biểu diễn tri thức cuối cùng được gọi là biểu diễn ngữ cảnh. Dạng luận lý chính là sự biểu diễn trung gian của ngữ nghĩa như ở hình H.2.1.

Phân tích cú pháp



biểu diễn ngữ nghĩa

Dạng luận lý

(RED1 <THE b1 BALL>)

biểu diễn ngữ cảnh

Biểu diễn cuối cùng

Red (BO73)

H.2.1. Dạng luận lý là dạng biểu diễn trung gian cho ngữ nghĩa

2.2 Ngôn ngữ dạng luận lý

Từ là đơn vị cơ bản của ngữ nghĩa, ngữ nghĩa của câu phụ thuộc vào nghĩa của các từ mà nó sử dụng. Trong phần này chúng ta sẽ nói đến ngôn ngữ, trong đó có sự kết hợp nghĩa của các thành phần để tạo nghĩa cho các mệnh đề phức tạp hơn. Các thành phần cấu thành câu của ngôn ngữ dạng luận lý gồm có: nguyên từ, vị từ, mệnh đề.

Những dạng phức tạp hơn sẽ được xây dựng từ các câu đơn giản, kết nối với nhau bằng các cấu trúc:

Toán tử luận lý: NOT, AND, OR...

Lượng từ: MOST, PLUR.

Toán tử thì: PAST, PRE, FUT

- **Mã hóa sự nhập nhằng trong dạng luận lý:**

Thông thường một từ có nhiều nghĩa nên sẽ gây ra sự nhập nhằng, do đó trong dạng luận lý người ta phải miêu tả tất cả các nghĩa của từ. Ví dụ từ BALL trong tiếng Anh vừa có nghĩa quả bóng lấy nghĩa BALL1, vừa có nghĩa buổi khiêu vũ lấy nghĩa BALL2. Ngoài ra tên riêng, đại từ cũng gây ra sự nhập nhằng nên ta phải thêm các cấu trúc như:

(NAME < variable > < name >) : cho tên riêng

(PRO < variable > < preposition >) : cho đại từ

- **Động từ và trạng thái:**

Động từ được ánh xạ vào các nghĩa tương ứng và đóng vai trò vị từ trong dạng luận lý. Tuy nhiên điều này làm mất đi tính tổng quát của động từ khi chúng cùng đề cập một hiện tượng nhưng trong các tiểu tiết khác nhau.

Ví dụ: John broke the window with the hammer

The hammer broke the window

The window broke

Trong ba câu trên, chúng ta mong muốn là động từ break sẽ được ánh xạ vào một nghĩa cho từng trường hợp khác nhau. Vì thế để giữ tính tổng quát của động từ, người ta đã dùng văn phạm

tình huống (case grammar) để miêu tả nghĩa của câu thông qua mối quan hệ giữa động từ với các đối tượng tham gia trong sự kiện, đó là: tác nhân (agent); chủ tố (theme); công cụ (instrument).

Ngoài các chức năng: tác nhân, chủ tố, công cụ, trong văn phạm tình huống còn có chức năng: nơi chốn, đồng chủ tố, đồng tác nhân...

Ví dụ: Câu "John broke the window with the hammer" sẽ được ánh xạ vào dạng luận lý:

($\exists e$: (& PAST BREAKS e) (AGENT e (NAME j1 "John")) (THEME e < THE w1 WINDOW >) (INSTR e < THE h1 HAMMER))))

Dạng tổng quát của dạng luận lý là:

(Event - p e [Relation l e obj l] ... [Relation n e Obj n])

3. TẠO TỰ ĐỘNG DẠNG LUẬN LÝ TRONG QUÁ TRÌNH PHÂN TÍCH CÚ PHÁP CHO CÂU CỦA NGÔN NGỮ TỰ NHIÊN

Trong phần này chúng tôi sẽ trình bày phương pháp liên kết ngữ nghĩa với cấu trúc cú pháp để tạo tự động dạng luận lý trong quá trình phân tích cú pháp.

Một trong các nguyên tắc tạo biểu diễn ngữ nghĩa là quá trình tổng hợp. Nghĩa của một thành phần trên cây cú pháp được dẫn ra từ nghĩa của các thành phần con. Cũng tương tự như lý thuyết tổng hợp của cú pháp, luật sinh áp dụng trên cơ sở phạm trù của các thành phần trong luật, mà không quan tâm đến cấu trúc bên trong của nó. Để phát triển lý thuyết tổng hợp cho biểu diễn ngữ nghĩa, người ta giả định đã tồn tại sự tương đương giữa cấu trúc cú pháp và luận lý. Tuy nhiên thực tế không đơn giản.

Ví dụ câu "Jill loves every dog"

có cấu trúc cú pháp:

((Jill) (loves (every dog)))

nhưng dạng luận lý lại là:

(EVERY d: (DOG1 d) (LOVES1 l1 (NAME j1 "Jill") d))

Rõ ràng hai cấu trúc này không giống nhau về thứ tự các thành phần.

Vì cấu trúc cú pháp là có tầm vực, còn cấu trúc của dạng luận lý thì không. Từ đó người ta phải điều chỉnh lý thuyết tổng hợp để tạo ra dạng luận lý như sau:

(LOVES1 l1 (NAME j1 "Jill") < EVERY d DOG1 >).

Dạng này gần với dạng của cấu trúc cú pháp. Để tạo ra dạng luận lý trong quá trình phân tích cú pháp người ta phải gán cấu trúc ngữ nghĩa vào các thành phần cú pháp và để giải quyết các vị từ đơn nguyên và các dạng vị từ phức tạp, trong quá trình tổng hợp người ta đã dùng phép toán λ (lambda calculus) để hình thức hóa vị từ. Để giải thích quá trình tạo ra dạng biểu diễn luận lý dựa vào phép toán lambda tôi sẽ trình bày phép toán lambda và thu giảm lambda như sau:

3.1. Phép toán lambda và thu giảm lambda [4]

Phép toán lambda là một ngôn ngữ mạnh dựa trên tập các thành phần nguyên thủy. Các công thức trong phép toán lambda gồm những xác nhận tương đương có dạng:

< expression > = < expression >

Tiên đề quyết định trong hệ thống này, phục vụ cho mục đích biểu diễn ngữ nghĩa là:

(($\lambda x P x$) a) = P (x/a)

với P là công thức bất kỳ có liên quan đến x. P (x/a) là công thức mà trong đó mỗi x được thay thế bằng a. Trong tiên đề này có hai phép toán cơ bản là: thu giảm lambda (lambda reduction) và trừu tượng lambda (lambda abstraction). Thu giảm lambda được thực

hiện từ trái qua phải của tiên đề. Trừu tượng lambda thực hiện từ phải qua trái tiên đề. Thu giảm lambda là quá trình thay thế một công thức bằng công thức đơn giản hơn mà tương đương với nó.

3.2 Phép toán lambda và thu giảm lambda với dạng biểu diễn luận lý

Ví dụ câu "Jack kissed Sue" có dạng luận lý:

(KISS1 k1 (NAME j1 "Jack") (NAME s1 "Sue"))

nếu gán cấu trúc ngữ nghĩa cho thành phần VP trong cấu trúc cú pháp thì VP sẽ bao gồm nhóm từ kissed Sue. Đây là vị từ đơn nguyên. Vị từ này đúng nếu có một đối tượng "kiss" Sue. Dựa vào phép toán *lambda* ta sẽ biểu diễn vị từ đơn nguyên như sau:

$(\lambda x (\text{KISS1 } k1 (\text{NAME } s1 \text{ "Sue"})))$

Sau đó ta có thể xây dựng một mệnh đề từ biểu thức *lambda* và một đối số

Trong ngôn ngữ dạng luận lý ta sẽ có mệnh đề của câu *Jack kissed Sue* là:

$((\lambda x (\text{KISS1 } k1 x (\text{NAME } s1 \text{ "Sue"}))) (\text{NAME } j1 \text{ "Jack"}))$

Vậy mệnh đề trên đúng nếu và chỉ nếu (NAME j1 "Jack") phù hợp với vị từ $(\lambda x (\text{KISS1 } x (\text{NAME } s1 \text{ "Sue"})))$

Theo định nghĩa nó chỉ đúng khi và chỉ khi:

$(\text{KISS1 } k1 (\text{NAME } j1 \text{ "Jack"}) (\text{NAME } s1 \text{ "Sue"}))$ là đúng

Quá trình xác định mệnh đề từ biểu thức *lambda* như trên chính là thu giảm *lambda*.

Chính biểu thức *lambda* đã thiết lập mối quan hệ mật thiết của cặp cú pháp – ngữ nghĩa. Từ đó chúng ta nghĩ tới phương pháp biên dịch hướng đến cú pháp. Phương pháp này sẽ tự động truyền thuộc tính; tức là gán các cấu trúc ngữ nghĩa cho các nút trên cây cú pháp trong quá trình phân tích cú pháp và thực hiện quá trình tổng hợp thuộc tính để tạo tự động dạng luận lý cho câu. Để thực hiện được điều này chúng ta cần phải xây dựng từ điển (lexicon) sao cho với mỗi từ phải chứa đầy đủ các nghĩa. Đồng thời phải nhúng ngữ nghĩa vào trong các thành phần của luật sinh trong văn phạm.

Hình H.3.2 là ví dụ về từ điển có nhúng nét (feature) ngữ nghĩa SEM và một số nét cú pháp.

a (art AGR 3s SEM INDEF1)

can (aux SUBCAT base SEM CAN1)

car (n SEM CAR1 AGR 3s)

cry (v SEM CRY1 VFORM base SUBCAT_none)

decide (v SEM DECIDES1 VFORM base SUBCAT_none)

decide (v SEM DECIDES-ON1 VFORM base SUBCAT_pp:on)

dog (n SEM DOG1 AGR 3s)

fish (n SEM FISH AGR 3s)

house (n SEM HOUSE1 AGR 3p)

has (aux VFORM pres AGR 3s SUBCAT pastprt SEM PERF)

he (pro SEM HE1 AGR 3s)

in (p PFORM {LOC MOT} SEM (N-LOC1))

Jill (name AGR 3s SEM "Jill")

man (n SEM MAN1 AGR 3s)

men (n SEM (PLUS MAN1) AGR 3p)

on (p PFORM LOC SEM ON-LOC1)

saw (v SEM SEES1 VFORM past SUBCAT_np AGR ?a)
 see (v SEM SEES1 VFORM base SUBCAT_np IRREG-PAST + EN-PASTPRT +)
 she (pro AGR 3s SEM SHE1)
 the (art SEM THE AGR {3s 3p})
 to (to AGR - VFORM inf)

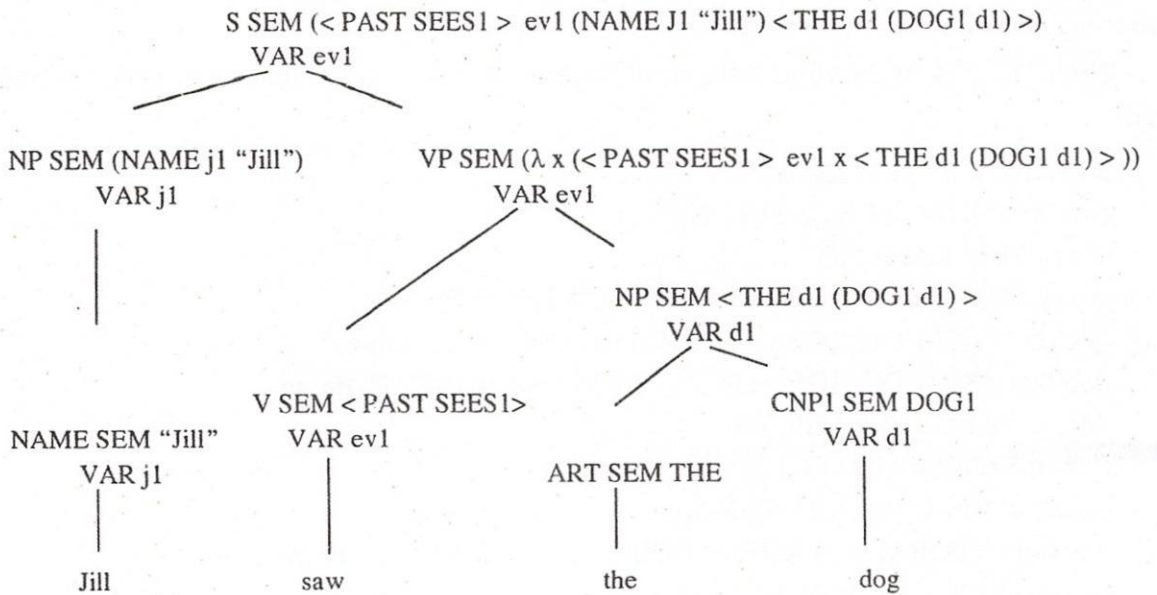
H.3.2. Từ điển đơn giản có những nét ngữ nghĩa SEM

Để thực hiện biên dịch hướng đến cú pháp, chúng ta cần văn phạm có những nét ngữ nghĩa SEM. Hình H.3.3. là văn phạm có nét ngữ nghĩa SEM.

1. (S SEM (?semvp ?semnp) → (NP SEM ?semnp) (VP SEM ?semvp)
2. (VP VAR ?v SEM (λ a2 (?semv ?v a2))) → (V[_none] SEM ?semv)
3. (VP VAR ?v SEM (λ a3 (?semv ?v a3 ?semnp))) → (V[_np] SEM ?semv) (NP SEM ?semnp)
4. (NP WH - VAR ?v SEM (PRO ?v ?sempro)) → (PRO SEM ?sempro)
5. (NP VAR ?v SEM (NAME ?v ?semname)) → (NAME SEM ?semname)
6. (NP VAR ?v SEM (<?semart ?v (?semcnp ?v) >)) → (ART SEM ?semart) (CNP SEM ?semcnp)
7. (CNP SEM ?semn) → (N SEM ?semn)
8. Head features for S, VP, NP, CNP: VAR

H.3.3. Văn phạm có nét ngữ nghĩa SEM

Quá trình phân tích cú pháp và tạo tự động dạng biểu diễn ngữ nghĩa cho câu tiếng Anh “Jill saw the dog” được trình bày ở H.3.4.



H.3.4. Tạo tự động dạng luận lý cho câu “Jill saw the dog” trong quá trình phân tích cú pháp

4. THỬ NGHIỆM GIẢI THUẬT TẠO TỰ ĐỘNG DẠNG LUẬN LÝ CHO CÂU CỦA NGÔN NGỮ TỰ NHIÊN

Như đã phân tích ở phần 3, để thực hiện quá trình ánh xạ từ câu sang dạng luận lý, ta phải xây dựng được cây phân tích cú pháp của câu. Trong quá trình phân tích cú pháp chúng ta phải “gắn” ngữ nghĩa vào các nút trên cây cú pháp, giống như quá trình tổng hợp thuộc tính trong biên dịch hướng đến cú pháp [1, 3].

Vậy quá trình “gắn” ngữ nghĩa vào cây cú pháp sẽ được thực hiện như thế nào?

Trong thử nghiệm này chúng tôi sửa đổi giải thuật phân tích cú pháp biểu đồ đi xuống (Top-down chart parser) [4] để thực hiện việc lưu giữ các bước của quá trình phân tích cú pháp, sẽ được sử dụng cho việc tạo sinh dạng luận lý của câu.

Giải thuật tạo tự động dạng luận lý của câu trong quá trình phân tích cú pháp của chúng tôi được trình bày ở giải thuật 4.1.

Giải thuật 4.1: Tạo dạng luận lý của câu.

Input 1. Câu nhập cùng với các từ loại tương ứng của các từ trong câu.

2. Tập luật sinh có các đối số được gọi là tập luật sinh P, của văn phạm lưu vết G_v

Output: dạng luận lý của câu nhập

Phương pháp:

1. Thực hiện phân tích cú pháp biểu đồ đi xuống cho câu nhập với văn phạm lưu vết.
2. Nếu câu nhập được chấp nhận (đúng cú pháp) thì tiếp tục ở bước 3. Ngược lại sẽ báo lỗi (do thiếu luật, câu viết sai, hoặc từ chưa có từ loại).
3. Lưu tất cả các luật được thu giảm vào một stack theo thứ tự: luật nào được thu giảm trước sẽ cho vào stack trước. Stack này được gọi là stack lưu vết.
4. Thực hiện giải thuật xác định dạng luận lý bằng các hành vi thu giảm:
 - a) Khi có một luật từ vựng được chọn để thu giảm, lập tức một nét ngữ nghĩa mới được tạo ra ở một nút tương ứng trên cây cú pháp (luật từ vựng có dạng $A \rightarrow a\bullet$, với a là ký hiệu kết thúc).
 - b) Khi một thành phần tương ứng với một nút mới trên cây cú pháp đượ xây dựng, (đồng nghĩa với việc một luật được thu giảm $A \rightarrow \alpha\bullet$), bộ biên dịch sẽ kiểm tra trong luật tương ứng của văn phạm gia tố G_t có phép toán λ không? Nếu có thì nét ngữ nghĩa SEM được thu giảm bởi phép rút gọn λ .

Giải thuật 4.2: Xác định dạng luận lý.

Input: stack lưu vết và chuỗi từ loại của câu nhập đã được chấp nhận.

Output: dạng luận lý của câu nhập.

Phương pháp:

Khởi động các biến tạm:

Make: = 0; Reduce: = 0;

Repeat

If nếu ký hiệu mục tiêu S ở trên đỉnh stack thu giảm và stack nhập chỉ còn ký hiệu \$ then quá trình xác định dạng luận lý của câu nhập kết thúc thành công; gán make: = 1;

Else begin

Giả sử $A \rightarrow \alpha\bullet$ là luật thu giảm đang nằm trên stack lưu vết.

If phần tử trên đỉnh stack thu giảm trùng với phần tử cuối của vế phải luật $A \rightarrow \alpha\bullet$ (dấu \bullet không phải là ký hiệu của văn phạm)

If chuỗi α xuất hiện trên stack thu giảm then reduce: = 1;
 If reduce = 1 then /* thu giảm*/

Begin

- loại bỏ r phần tử ra khỏi stack thu giảm; $r = |\alpha|$ ($|\alpha|$ là kích thước của α)
- $sp := sp - r + 1$; với sp là con trỏ chỉ mức trên của stack thu giảm.
- stack thu giảm $[sp] := A$
- xác định nét ngữ nghĩa SEM và biến VAR cho A.
- tạo phần tử mới có tên là A với các thuộc tính SEM, VAR.

end

else /* đẩy */

begin

- đẩy phần tử trên stack nhập lên stack thu giảm.
- xác định SEM, VAR cho phần tử này.

End;

End;

Until make = 1;

Ví dụ 4.1: *Jill saw the dog.*

a) Cho văn phạm phi ngữ cảnh G

- 1.S \rightarrow NP VP
- 2.VP \rightarrow V
- 3.VP \rightarrow V NP
- 4.NP \rightarrow PRO
- 5.NP \rightarrow NAME
- 6.NP \rightarrow ART CNP
- 7.CNP \rightarrow N

b) Từ văn phạm a), chúng tôi tạo văn phạm lưu vết G_v

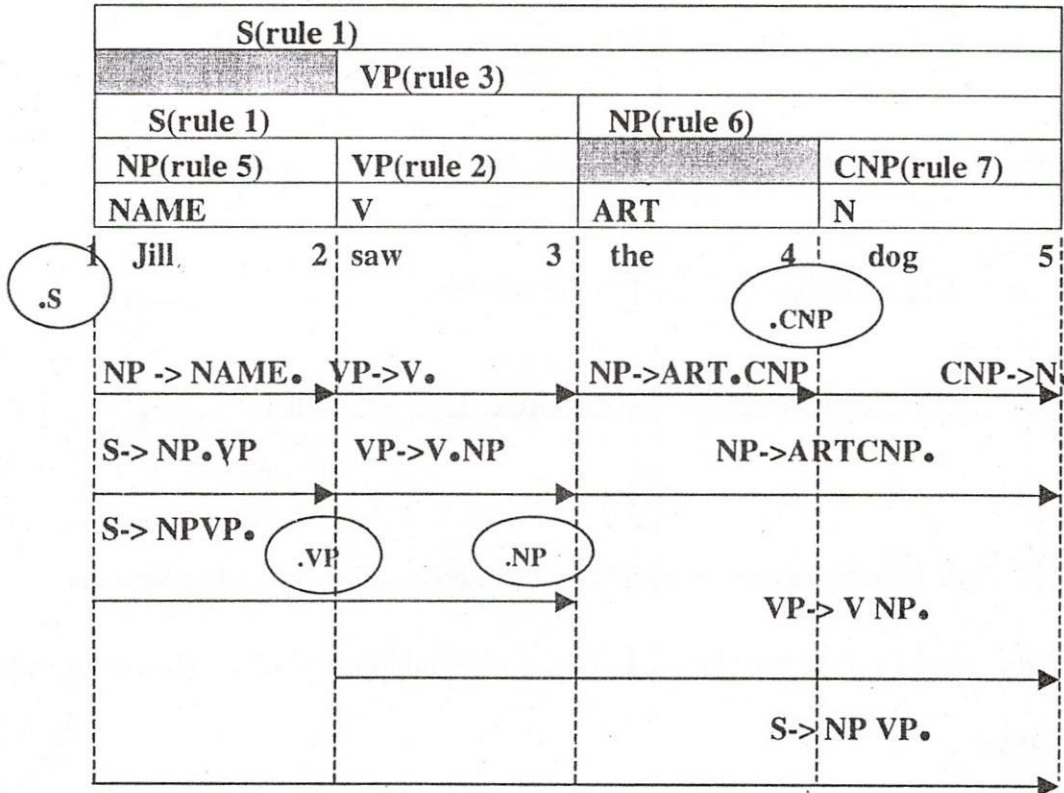
- 1.S(?idRule, ?entity) \rightarrow NP(?idRule, ?entity)VP(?idRule, ?entity)
- 2.VP(?idRule, ?entity) \rightarrow V
- 3.VP(?idRule, ?entity) \rightarrow V NP(?idRule, ?entity)
- 4.NP(?idRule, ?entity) \rightarrow PRO
- 5.NP(?idRule, ?entity) \rightarrow NAME
- 6.NP(?idRule, ?entity) \rightarrow ART CNP(?idRule, ?entity)
- 7.CNP(?idRule, ?entity) \rightarrow N

c) Cho văn phạm gia tố G_t

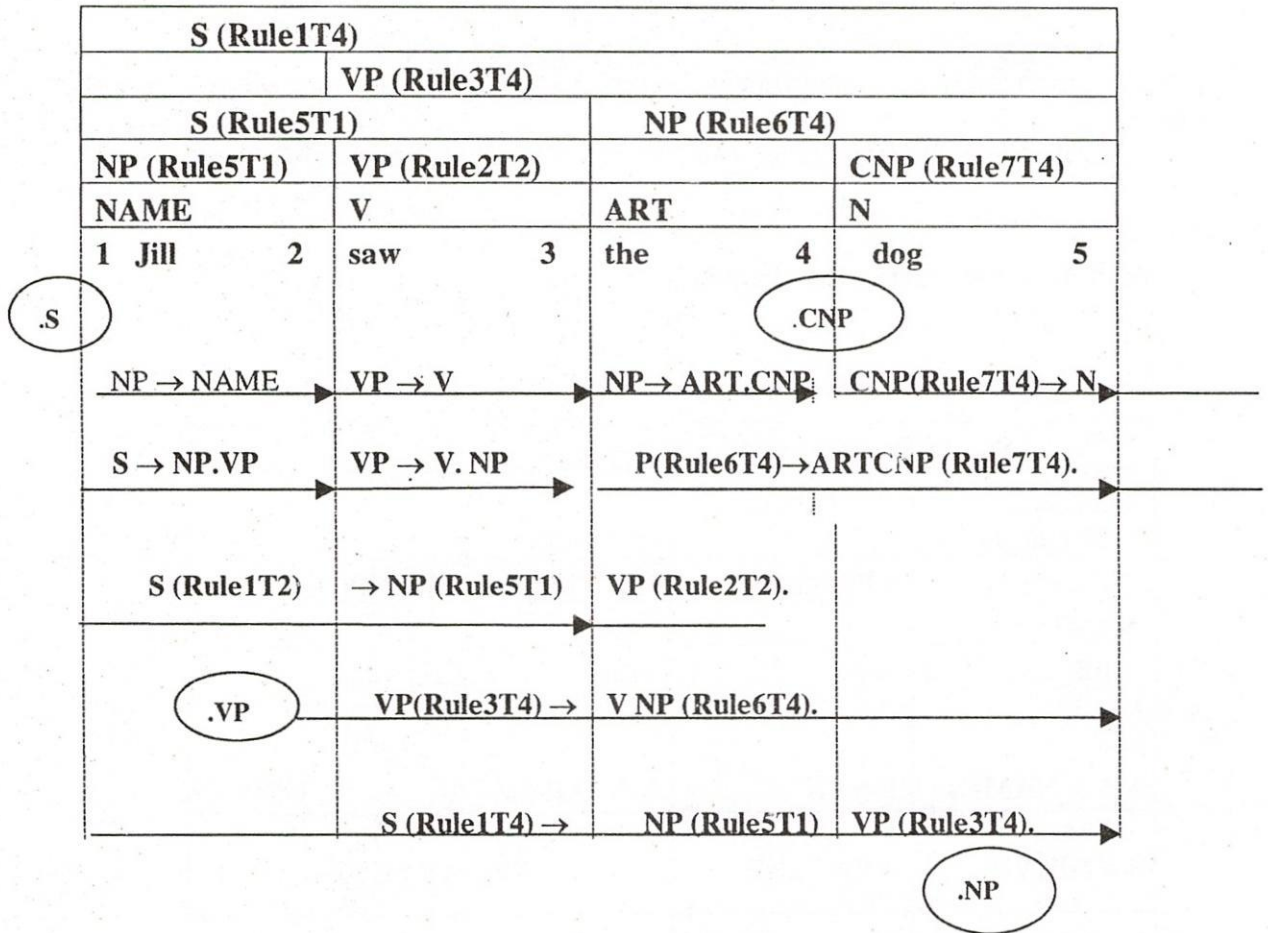
1. (S SEM (?semVP ?semNP)) \rightarrow (NP SEM ?semNP)(VP SEM ?semVP)
2. (VP SEM (lamdaX(?semV ?varV X)) VAR ?varV) \rightarrow
 (V_[none] SEM ?semV VAR ?varV)
3. (VP SEM (lamdaX(?semV ?varV X ?semNP)) VAR ?varV) \rightarrow
 (V_[np] SEM ?semV VAR ?varV)(NP SEM ?semNP VAR ?varNP)
4. (NP SEM (PRO ?varPRO ?semPRO) VAR ?varPRO) \rightarrow
 (PRO SEM ?semPRO VAR ?varPRO)
5. (NP SEM (NAME ?varNAME ?semNAME) VAR ?varNAME) \rightarrow
 (NAME SEM ?semNAME VAR ?varNAME)

- 6. (NP SEM (?semART ?varCNP ?semCNP) VAR ?varCNP) -> (ART SEM ?semART VAR ?varART)(CNP SEM ?semCNP VAR ?varCNP)
- 7. (CNP SEM (?semN ?varN) VAR ?varN) -> (N SEM ?semN VAR ?varN

Với từ điển ở H 3.2. ta có sơ đồ phân tích như sau :



H 4.1 .Quá trình phân tích cú pháp câu "Jill saw the dog" bằng giải thuật biểu đồ đi xuống



H.4.2 . Thực hiện phân tích cú pháp biểu đồ từ trên xuống trên văn phạm vết

Các tập thực thể của các ký hiệu không kết thúc được tính trước khi thực hiện phân tích:

.S:

- S -> . NP VP
- NP -> . PRO
- NP -> . NAME
- NP -> . ART CNP

.NP:

- NP -> . PRO
- NP -> . NAME
- NP -> . ART CNP

.VP:

- VP -> . V
- VP -> . V NP

.CNP:

- CNP -> . N

- Quá trình xác định dạng luận lý của chuỗi từ loại NAME V ART N của câu "Jill saw the dog"

Stack thu giảm	Stack nhập	Hành động	Bước
\$	NAME V ART N \$	Shift	1
\$ NAME	V ART N \$	Reduce NP->NAME	2

\$ NP(Rule5T1)	V ART N \$	Shift	3
\$ NP(Rule5T1) V	ART N \$	Shift	4
\$ NP(Rule5T1) V ART	N \$	Shift	5
\$ NP(Rule5T1) V ART N	\$	Reduce CNP->N	6
\$ NP(Rule5T1)V ART CNP (rule 7 T4)	\$	Reduce NP->ART CNP	7
\$ NP(Rule5T1)V NP(Rule6T4)	\$	Reduce VP->V NP	8
\$ NP(Rule5T1) VP(Rule3T4)	\$	Reduce S->NP VP	9
\$ S(Rule1T4)	\$	Accept	10

- Quá trình tạo SEM, VAR.

Bước	SEM, VAR
2	NAME SEM "Jill" VAR j1
3	NP SEM (NAME j1 "Jill") VAR j1
4	V SEM <PAST SEE1> VAR ev1
5	ART SEM THE VAR t1
6	N SEM DOG1 VAR d1
7	CNP SEM DOG1 VAR d1
8	NP SEM (THE d1 : (DOG1 d1)) VAR d1
9	VP SEM (λX (<PAST SEE1>ev1) X(THE d1:(DOG1 d1))) VAR ev1
10	S SEM (<PAST SEE1> ev1 (NAME j1 "Jill") (THE d1 : (DOG1 d1))) ev1

- Stack lưu vết.

NP(Rule5T1) -> NAME.	
CNP(Rule7T4) -> N.	
NP(Rule6T4) -> ART CNP(Rule7T4).	
VP(Rule3T4) -> V NP(Rule6T4).	
S(Rule1T4) -> NP(Rule5T1)VP(Rule3T4).	

5. KẾT LUẬN:

Giải thuật tạo tự động dạng luận lý mà chúng tôi vừa trình bày là sự cải tiến của hai giải thuật biểu đồ đi xuống và giải thuật phân tích cú pháp thứ tự yếu [1, 3].

Qua thực nghiệm chúng tôi thấy rằng với thuật giải biểu đồ từ trên xuống, cho kết quả phân tích khá nhanh (chúng tôi đã thực nghiệm phân tích với số luật sinh khoảng 32.000 luật). Quá trình xác định dạng luận lý lại là quá trình tổng hợp từ dưới lên. Do vậy chúng tôi phải kết hợp sửa đổi giải thuật phân tích cú pháp thứ tự yếu để diễn dịch dạng luận lý. Hơn nữa chúng ta biết rằng bộ phân tích thứ tự yếu, chỉ phù hợp với lớp văn phạm có tính chất sau:

- Không có luật sinh nào có vế phải rỗng ($A \rightarrow \emptyset$).
- Không có hai ký hiệu không kết thúc đứng kề nhau ở vế phải.

Nên nó không phù hợp với lớp văn phạm của ngôn ngữ tự nhiên, vì vậy chúng tôi phải sử dụng hai giải thuật cải thiện trên theo trình tự sau. Trước tiên xác định câu nhập có đúng cú pháp không, chúng tôi dùng giải thuật phân tích biểu đồ từ trên xuống, từ đó lưu được các bước phân tích, mà ở đó có thực hiện thu giảm các luật sinh.

Dựa vào chuỗi luật sinh thu giảm, chúng tôi dùng cơ chế đẩy thu giảm của giải thuật phân tích cú pháp thứ tự yếu trên văn phạm gia tố để xác định dạng luận lý. Chương trình của chúng tôi đã thực hiện được các giai thuật nêu trên và cho kết quả đúng. Chúng tôi sẽ mở rộng văn phạm và từ điển để có thể xử lý các dạng câu phức tạp.

ESTABLISHMENT OF A LOGICAL FORM FOR NATURAL LANGUAGE IN PARSE PROCESSING

Phan Thi Tươi, Nguyen Chi Hieu

University of Technology – Vietnam National University HCM City

(Received 14 March 2002, Revised 29 March 2002)

ABSTRACT: The paper presents the basic ideas underlying theories of semantics. It introduces a level of context independent meaning called the logical form, which can be produced directly from the syntactic structure of sentence in natural language. Then the paper presents our improvements of top down chart algorithm for parsing. It also introduces our tracing grammar, which keeps the steps of parse processing. These steps are used to compute the logical form of sentence.

TÀI LIỆU THAM KHẢO

- [1] Alfred V.Aho, Ravisethi, Jeffrey D.Ullman, *Compiler principles, Techniques, and Tools*, AddisonWesley Publishing Company, 1986.
- [2] Phan Thị Tươi, “Trợ giúp bắt lỗi chính tả tiếng Việt tự động bằng máy tính (giai đoạn 2)”, đề tài cấp thành phố, 2001, Trường Đại học Bách Khoa TP HCM.
- [3] Phan Thị Tươi, *Trình biên dịch*, NXB Đại học Quốc gia TP HCM, 2001.
- [4] James Allen, *Natural Language Understanding*, the Bengamin / comings Publishing company, Inc, 1995.
- [5] Alfred V.Aho, Jeffred D.Ullman, *The Theory of Parsing, Translation, and Compiling Volume 1: Parsing*, Prentice – Hall, Inc., 1972 (320 – 330).
- [6] W.J. Hutchins, *Machine Translation, past, present, future*, Ellis Howood Limited Publishers, 1986. [285-287]
- [7] Dezyle W. Lonsdale, Alescander M. Franz and John R.R. Leavitl, *Large – Scale Machine Translation and Interlingua approach*, center for Machine system from R & D to Large – Scale Deployment, Carnegie Mellon University, 1998.
- [8] Eric Nyberg, *The Kant Translation System from R & D to Large – Scale Deployment*, Carnegie Mellon University, 1998