

ỦY BAN NHÂN DÂN TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN



TÌỂU LUẬN

Môn: SEMINAR CHUYÊN ĐỀ

Chuyên đề: LẬP TRÌNH LOGIC VÀ ỨNG DỤNG

Đề tài: Xây dựng công cụ phân tích cú pháp

SINH VIÊN THỰC HIỆN : Lê Minh Phúc

MSSV : 3120410405

LỚP: DKP1201 – Mã học phần: 841482 (Nhóm 2)

Giảng viên hướng dẫn : PGS.TS. Nguyễn Tuấn Đăng

.....

TP. HCM tháng 10 /2024

NỘI DUNG

CHƯƠNG 1: GIỚI THIỆU	1
1.1. Giới thiệu về đề tài	1
1.2. Mục tiêu	1
1.3. Cách tiếp cận	1
1.4. Đối tượng nghiên cứu	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	3
2.1 Xử lý ngôn ngữ tự nhiên.....	3
2.1.1 Xử lý ngôn ngữ tự nhiên là gì.....	3
2.1.2 Lịch sử phát triển của phân tích cú pháp.....	3
2.1.3 Cây cú pháp	4
2.2 Văn phạm DCG (Definite Clause Grammar).....	5
2.3 Ngôn ngữ lập trình Prolog.....	5
CHƯƠNG 3: XÂY DỰNG CÔNG CỤ PHÂN TÍCH CÚ PHÁP.....	7
3.1 Vẽ cây cú pháp.....	7
3.1.1. Nam thường đến thư viện.....	7
3.1.2. Nam rất thích đọc sách ở thư viện	9
3.1.3. Nhà của Nam ở gần trường	11
3.1.4. Nam mới mua mấy cuốn sách mới.....	13
3.1.5. Nam tặng Lan một cuốn sách rất hay	14
3.2 Chuyển sang văn phạm DCG cho từng câu	16
3.2.1. Nam thường đến thư viện.....	16
3.2.2. Nam rất thích đọc sách ở thư viện	17
3.2.3. Nhà của Nam ở gần trường	18
3.2.4. Nam mới mua mấy cuốn sách mới.....	19
3.2.5. Nam tặng Lan một cuốn sách rất hay	20
3.3 Gộp văn phạm DCG:	21
CHƯƠNG 4: TỔNG KẾT	37
TÀI LIỆU THAM KHẢO.....	38

CHƯƠNG 1: GIỚI THIỆU

1.1. Giới thiệu về đề tài

Đề tài “Xây dựng công cụ phân tích cú pháp” trong chuyên đề “Lập trình logic và ứng dụng” tập trung vào việc phát triển một hệ thống có khả năng phân tích cấu trúc ngữ pháp của các câu trong ngôn ngữ tự nhiên hoặc ngôn ngữ lập trình bằng cách sử dụng các quy tắc logic. Dựa trên nền tảng của ngôn ngữ lập trình logic, đặc biệt là Prolog, đề tài này sẽ khai thác sức mạnh của các quy tắc suy luận để xác định mối quan hệ giữa các thành phần trong câu, từ đó xây dựng cây cú pháp cho câu đầu vào. Công cụ phân tích cú pháp này sẽ giúp hỗ trợ các ứng dụng xử lý ngôn ngữ tự nhiên (NLP), như kiểm tra ngữ pháp, hệ thống đối thoại thông minh và trích xuất thông tin, đồng thời mở rộng ứng dụng của lập trình logic trong các hệ thống suy luận tự động.

1.2. Mục tiêu

Mục tiêu của đề tài “Xây dựng công cụ phân tích cú pháp” là tạo ra một hệ thống có khả năng nhận diện và phân tích cấu trúc cú pháp của các câu dựa trên các quy tắc logic, nhằm hiểu rõ mối quan hệ giữa các thành phần trong câu. Sử dụng ngôn ngữ Prolog, đề tài hướng tới việc xây dựng một công cụ không chỉ chính xác trong phân tích cú pháp mà còn dễ dàng mở rộng, hỗ trợ các ứng dụng như kiểm tra ngữ pháp, trích xuất thông tin, và xây dựng hệ thống đối thoại.

1.3. Cách tiếp cận

Cách tiếp cận đề tài “Xây dựng công cụ phân tích cú pháp” dựa trên việc sử dụng các quy tắc DCG (Definite Clause Grammar) trong ngôn ngữ Prolog để thiết lập các quy tắc cú pháp và phân tích cấu trúc của câu. Đầu tiên, đề tài sẽ nghiên cứu các mô hình cú pháp phổ biến và lựa chọn các quy tắc phù hợp cho ngôn ngữ mục tiêu. Sau đó, các quy tắc này sẽ được mã hóa dưới dạng mệnh đề logic để Prolog có thể suy luận và xây dựng cây cú pháp tự động cho các câu đầu vào. Tiếp cận này giúp hệ thống phân tích cú pháp

linh hoạt, có thể mở rộng và điều chỉnh theo các cấu trúc ngữ pháp khác nhau, từ đó hỗ trợ nhiều ứng dụng trong xử lý ngôn ngữ tự nhiên.

1.4. Đối tượng nghiên cứu

Đối tượng nghiên cứu:

- a. Ngôn ngữ Prolog
- b. Văn phạm DCG, cây cú pháp
- c. Phương pháp phân tích cú pháp một câu tiếng Việt

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Xử lý ngôn ngữ tự nhiên

2.1.1 Xử lý ngôn ngữ tự nhiên là gì

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một lĩnh vực của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc giúp máy tính có thể hiểu, diễn giải và tương tác với ngôn ngữ của con người một cách tự nhiên. Mục tiêu của NLP là xây dựng các hệ thống có khả năng phân tích, xử lý và tạo ra ngôn ngữ một cách hiệu quả, như một người hiểu ngôn ngữ

Một ngôn ngữ tự nhiên thường tuân theo cấu trúc phân cấp và chứa các thành phần sau:

- Câu
- Mệnh đề
- Cụm từ
- Từ

2.1.2 Lịch sử phát triển của phân tích cú pháp

Lịch sử nghiên cứu và phát triển phân tích cú pháp (syntax analysis) trong khoa học máy tính là một hành trình từ những năm đầu của lập trình đến các hệ thống hiện đại sử dụng trí tuệ nhân tạo và học máy để hiểu và xử lý ngôn ngữ tự nhiên. Sau đây là các thời kì nghiên cứu và phát triển phân tích cú pháp:

- Thời kỳ khởi đầu (1950s - 1960s): Noam Chomsky giới thiệu lý thuyết ngữ pháp hình thức vào cuối thập niên 1950, với các lớp ngữ pháp như ngữ pháp phi ngữ cảnh (context-free grammar), đặt nền tảng cho phân tích cú pháp trong ngôn ngữ lập trình và ngôn ngữ tự nhiên
- Thời kỳ phát triển thuật toán (1970s - 1980s): Nhiều thuật toán phân tích cú pháp được nghiên cứu và phát triển, như LR parser của Donald Knuth,

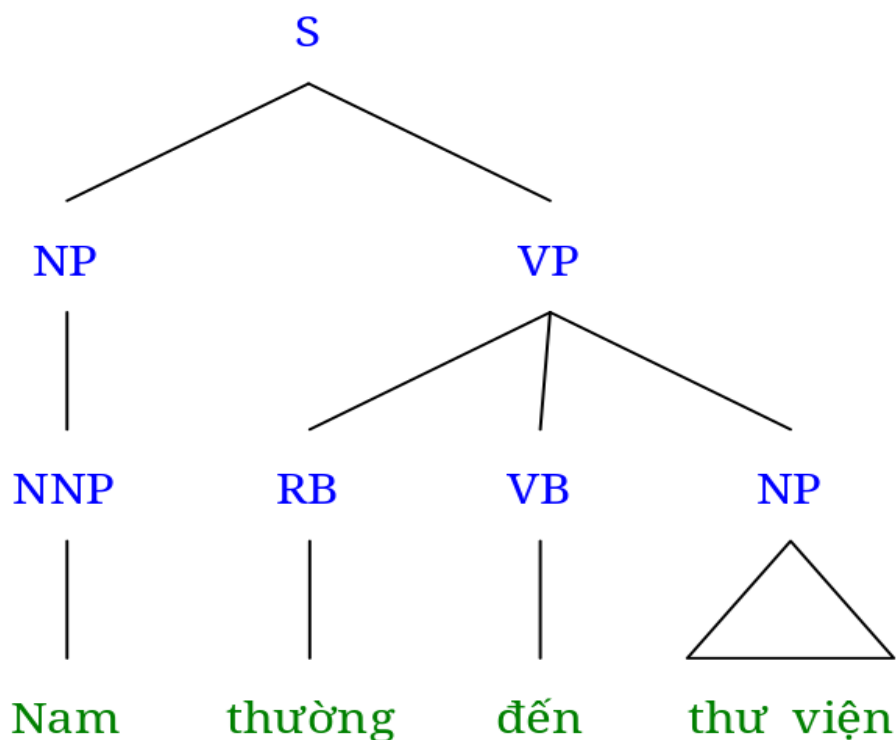
Earley parser và CYK (Cocke-Younger-Kasami) parser. Các thuật toán phân tích cú pháp hướng tới tối ưu hóa việc xây dựng trình biên dịch và dịch mã hiệu quả, đồng thời áp dụng trong một số hệ thống xử lý ngôn ngữ tự nhiên sơ khai.

- Thời kỳ ứng dụng xác suất (1990s): Các phương pháp phân tích cú pháp dựa trên xác suất ra đời, giúp xử lý tốt hơn sự mơ hồ trong ngôn ngữ tự nhiên.
- Thời kỳ học máy và học sâu (2000s - 2010s): Học máy được áp dụng trong phân tích cú pháp, sử dụng các mô hình như SVM, CRF và các mô hình mạng nơ-ron như RNN và LSTM để xử lý ngữ cảnh câu.
- Thời kỳ Transformer và AI hiện đại (2010s - nay): Các mô hình này học từ dữ liệu lớn và ngữ cảnh phong phú, giúp cải thiện đáng kể độ chính xác trong các ứng dụng như dịch máy, chatbot, và trợ lý ảo. Phân tích cú pháp hiện nay tích hợp sâu vào các mô hình học sâu hiện đại, giúp xử lý ngôn ngữ tự nhiên ở cấp độ cao và phức tạp hơn

2.1.3 Cây cú pháp

Cây cú pháp (syntax tree) là một cấu trúc phân cấp biểu diễn cách các từ trong câu liên kết với nhau theo quy tắc ngữ pháp. Mỗi nút trong cây đại diện cho một thành phần ngữ pháp, như cụm danh từ (NP), cụm động từ (VP), động từ (V), hoặc danh từ (N). Gốc của cây thường là câu (S), từ đó các nhánh phân ra đến các từ cụ thể trong câu

Ví dụ về cây cú pháp:



2.2 Văn phạm DCG (Definite Clause Grammar)

Văn phạm các mệnh đề xác định (Definite Clause Grammar - DCG) là một loại ngữ pháp ngữ cảnh được sử dụng trong lập trình logic, đặc biệt phổ biến trong ngôn ngữ Prolog. DCG cho phép mô tả cấu trúc ngữ pháp của câu bằng các quy tắc logic dưới dạng mệnh đề, giúp định nghĩa mối quan hệ giữa các thành phần trong câu.

DCG rất hiệu quả trong việc xử lý cú pháp của ngôn ngữ tự nhiên nhờ khả năng linh hoạt và rõ ràng, giúp hỗ trợ các ứng dụng như phân tích cú pháp tự động, trích xuất thông tin và xây dựng cây cú pháp.

2.3 Ngôn ngữ lập trình Prolog

Prolog (Programming in Logic) là một ngôn ngữ lập trình logic, ra đời vào đầu những năm 1970, được thiết kế để hỗ trợ các ứng dụng yêu cầu suy luận tự động và ra quyết định. Khác với các ngôn ngữ lập trình truyền thống dựa trên việc mô tả các bước thực

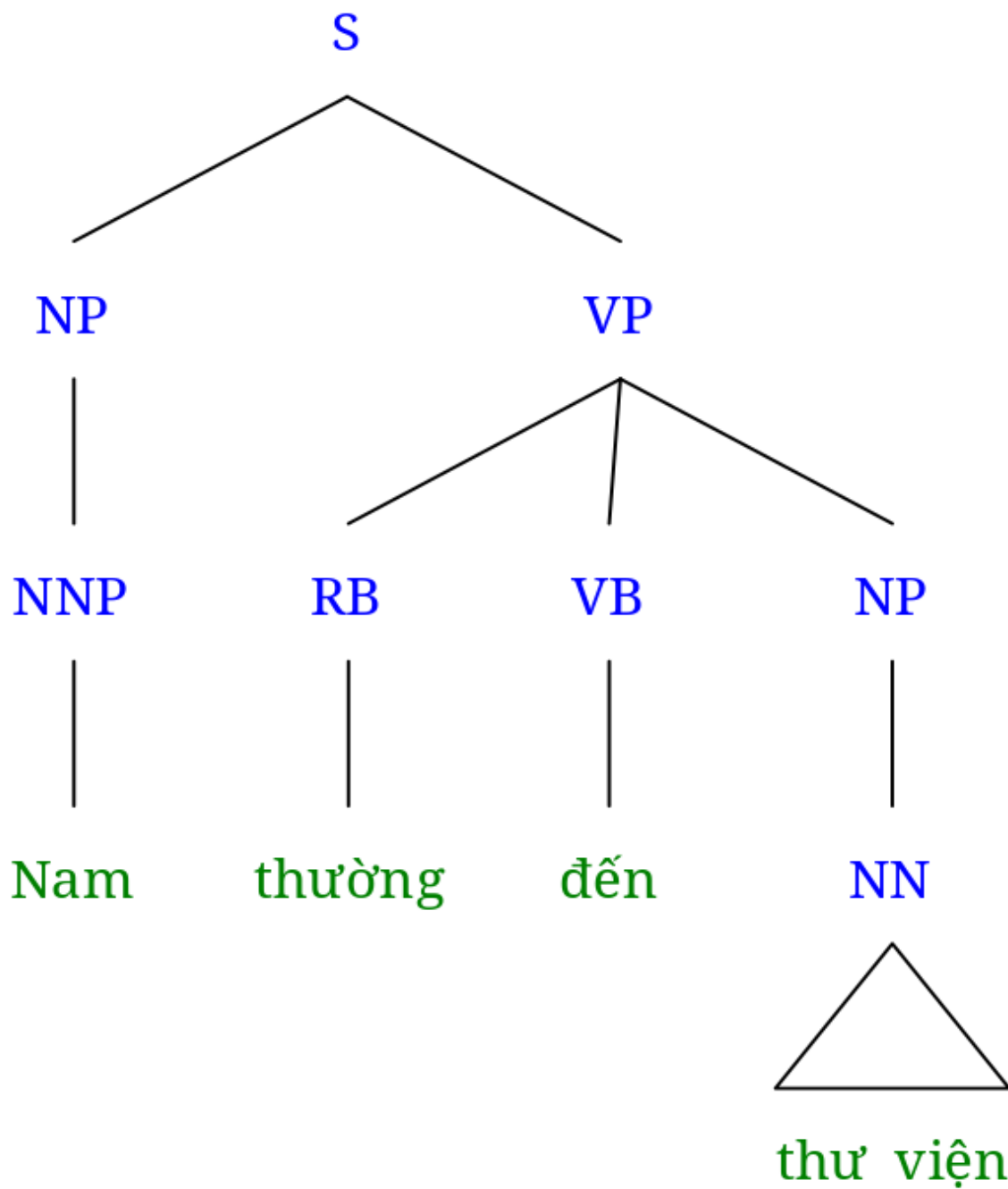
hiện cụ thể, Prolog cho phép lập trình viên định nghĩa các quan hệ và quy tắc logic, từ đó hệ thống tự tìm kiếm lời giải dựa trên những quy tắc này.

Prolog đặc biệt hữu ích trong các lĩnh vực như trí tuệ nhân tạo, xử lý ngôn ngữ tự nhiên, hệ thống chuyên gia và tìm kiếm đường đi, nhờ khả năng mô tả bài toán một cách tự nhiên qua các điều kiện và quan hệ logic. Trong Prolog, các câu lệnh thường được viết dưới dạng mệnh đề và quy tắc, cho phép hệ thống suy luận dựa trên các câu hỏi hoặc truy vấn được đặt ra.

CHƯƠNG 3: XÂY DỰNG CÔNG CỤ PHÂN TÍCH CÚ PHÁP

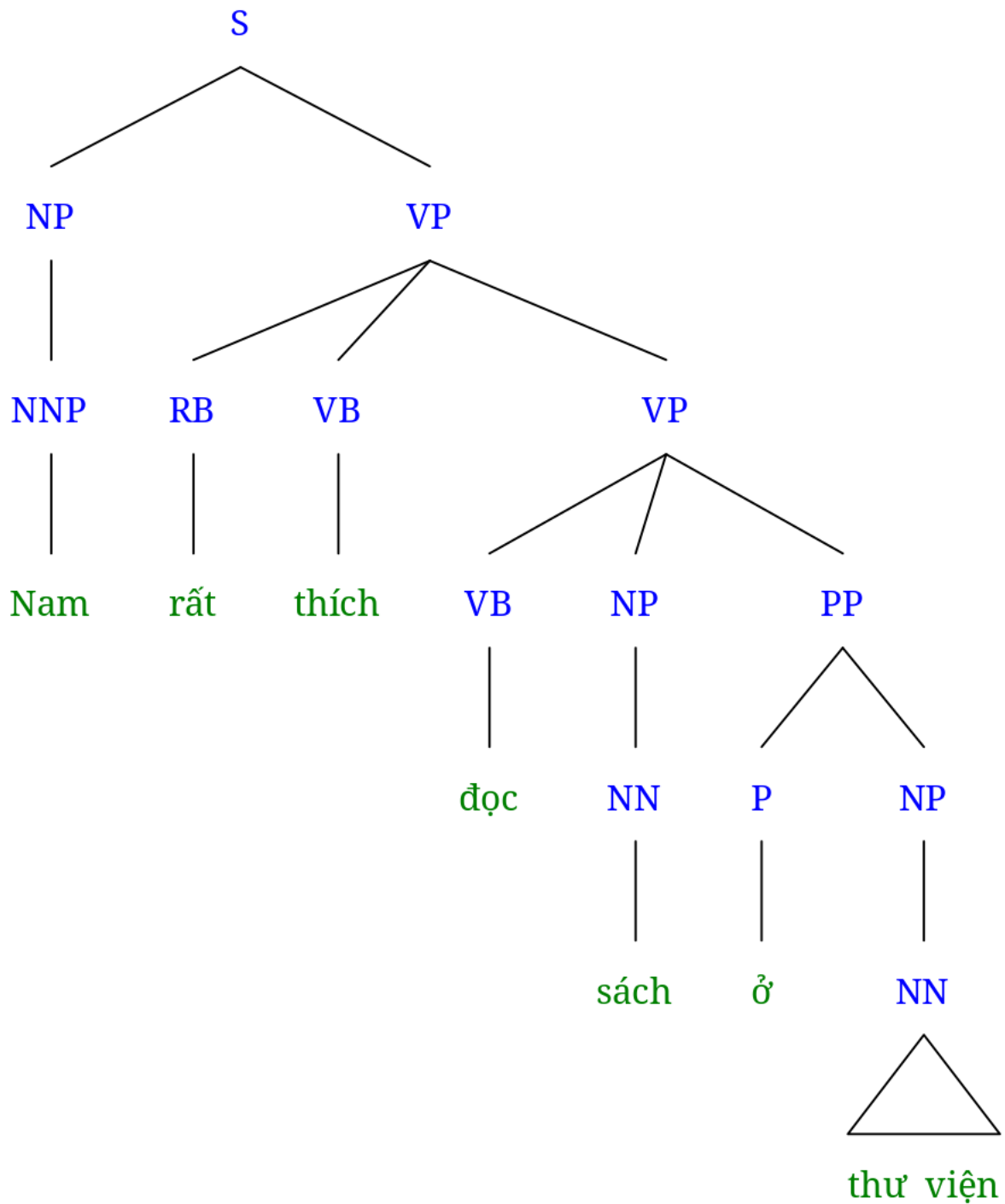
3.1 Vẽ cây cú pháp

3.1.1. Nam thường đến thư viện



- Giải thích cấu trúc:
- S: Câu gồm chủ ngữ (NP) và vị ngữ (VP).
- NP (Chủ ngữ): "Nam" (NNP) là người thực hiện hành động.
- VP (Vị ngữ): Bao gồm:
 - Trạng từ (RB): "thường" bổ nghĩa cho động từ "đến", chỉ rõ hành động này diễn ra theo một tần suất nhất định (thường xuyên).
 - Động từ (VB): "đến" là hành động chính của chủ ngữ.
 - Cụm danh từ (NP) có danh từ thường (NN) là "thư viện".

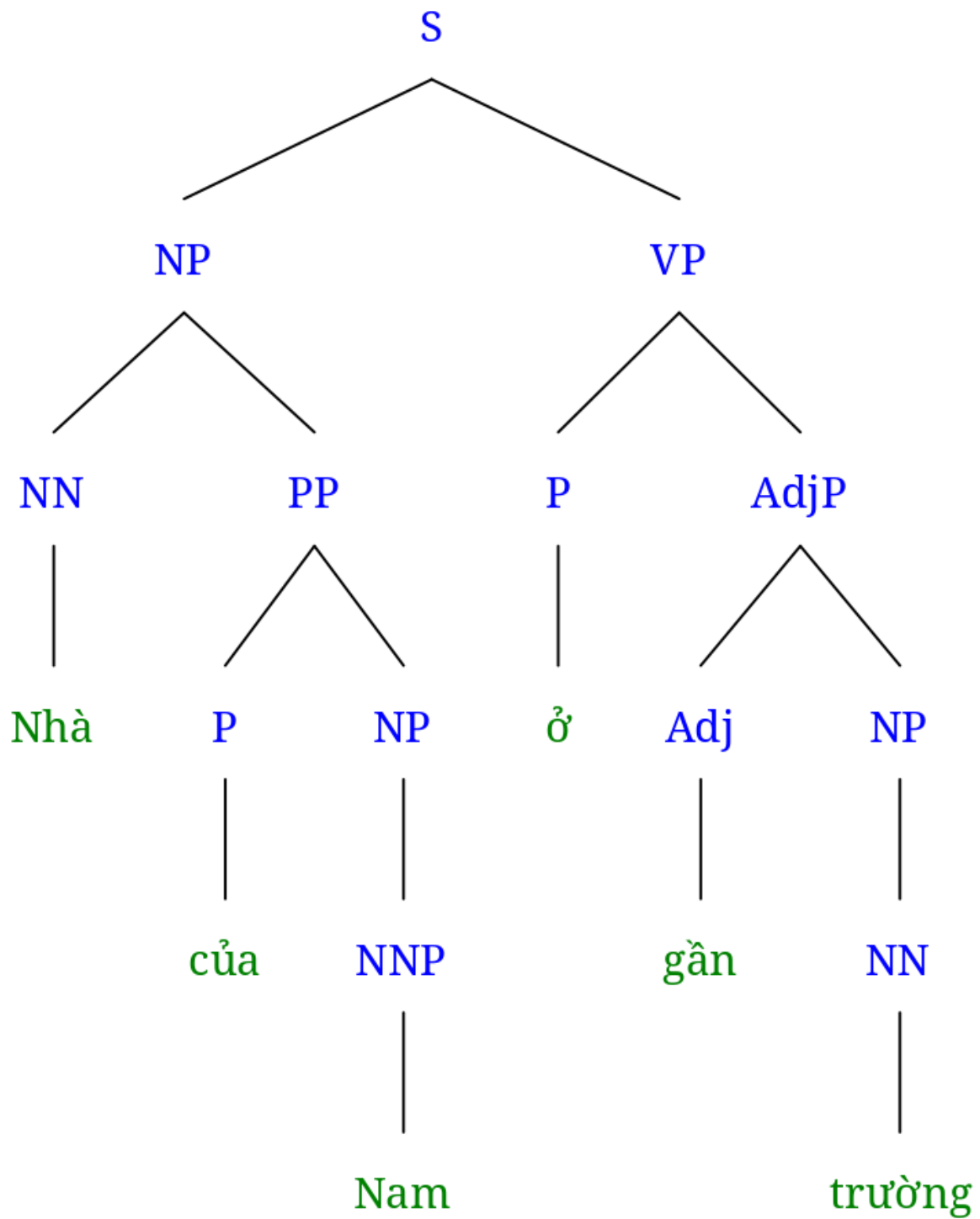
3.1.2. Nam rất thích đọc sách ở thư viện



- Giải thích cấu trúc:
- S: Gốc của câu (Sentence).

- NP: Cụm danh từ chủ ngữ (Noun Phrase) với danh từ riêng (NNP) là "Nam".
- VP: Cụm động từ chính (Verb Phrase), bao gồm:
 - RB (Adverb): "rất", trạng từ bổ nghĩa cho động từ "thích".
 - VB (Verb): "thích", động từ chính của câu.
- VP: Cụm động từ phụ, diễn tả hành động cụ thể "đọc sách ở thư viện".
 - VB: "đọc", động từ mô tả hành động.
 - NP: tân ngữ của động từ "đọc".
 - NN: "sách" là danh từ thường
- PP (Prepositional Phrase): Cụm giới từ chỉ địa điểm.
 - P (Preposition): "ở", giới từ chỉ địa điểm.
 - NP: tân ngữ
 - "thư viện", danh từ chỉ nơi hành động "đọc" diễn ra.

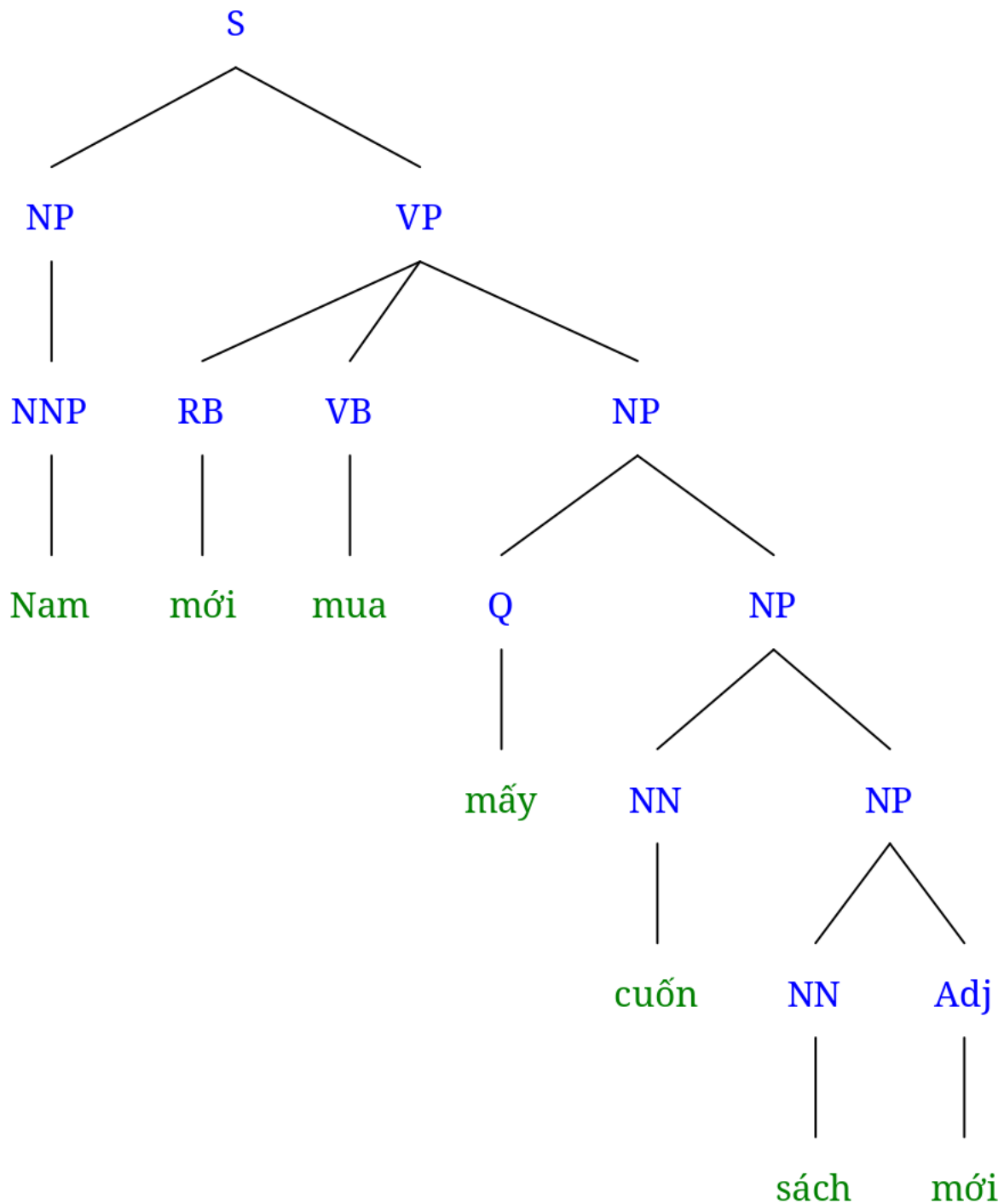
3.1.3. Nhà của Nam ở gần trường



- Giải thích cấu trúc:

- S (Sentence): Gốc của câu.
- NP (Noun Phrase): Cụm danh từ chủ ngữ gồm:
 - NN (Noun): "Nhà" là danh từ chính của chủ ngữ.
 - PP (Prepositional Phrase): Cụm giới từ bổ nghĩa cho "Nhà".
 - P (Preposition): "của", giới từ chỉ sở hữu.
 - NP: Cụm danh từ với NNP (Proper Noun) là "Nam", người sở hữu "Nhà".
- VP (Verb Phrase): Cụm động từ vị ngữ, gồm:
 - P (Preposition): "ở", giới từ chỉ địa điểm.
 - AdjP (Adjective Phrase): Cụm tính từ, mô tả vị trí "Nhà".
 - Adj: "gần", tính từ chỉ vị trí gần.
 - NP :
 - Cụm danh từ với NN (Noun) là "trường", nơi mà "Nhà" gần.

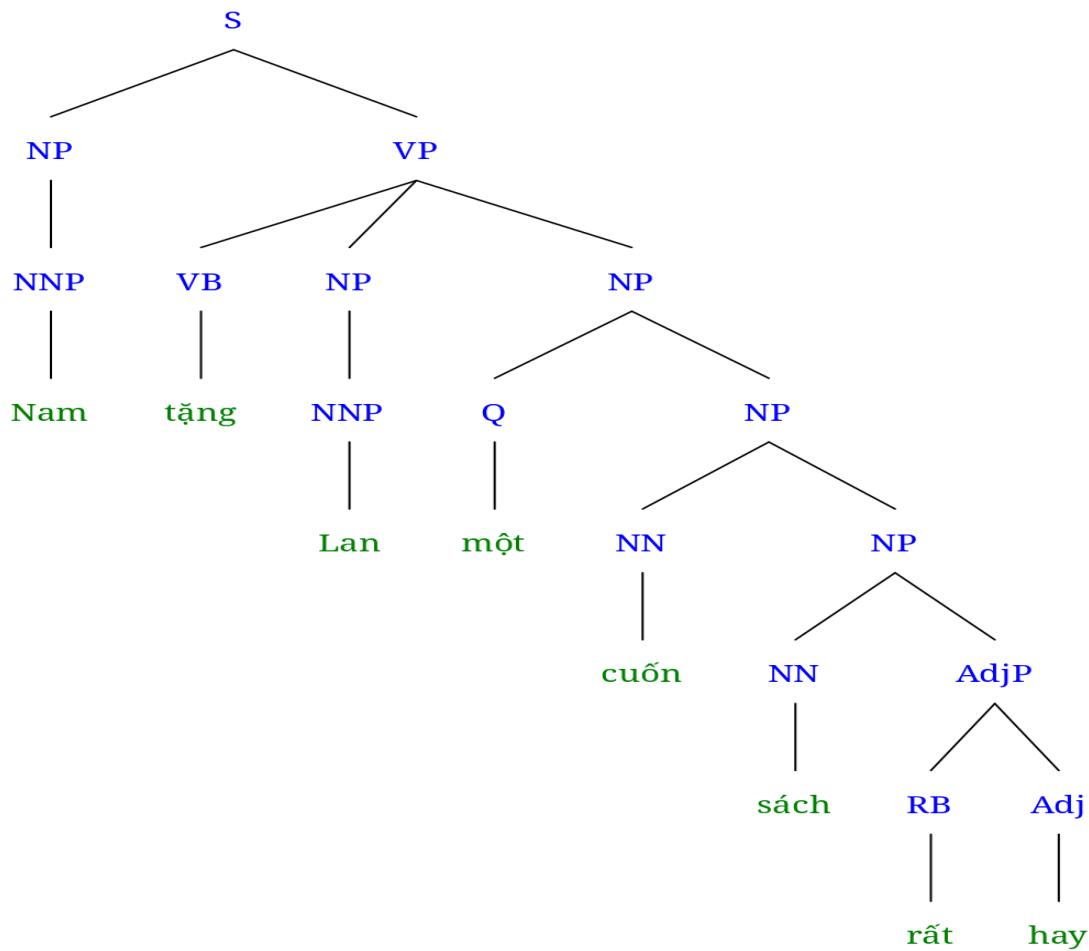
3.1.4. Nam mới mua mấy cuốn sách mới



- Giải thích cấu trúc
- S: Gốc của câu (Sentence), bao gồm cụm danh từ (NP) và cụm động từ (VP).

- NP: Cụm danh từ chủ ngữ, chứa:
 - NNP (Proper Noun): "Nam", danh từ riêng chỉ người thực hiện hành động.
- VP: Cụm động từ chính, chứa:
 - RB (Adverb): "mới", trạng từ bổ nghĩa cho động từ "mua", cho biết hành động vừa mới xảy ra.
 - VB (Verb): "mua", động từ chính của câu.
 - NP: Cụm danh từ tân ngữ, chỉ đối tượng của hành động "mua", bao gồm:
 - Q (Quantifier): "mấy", lượng từ bổ sung cho danh từ "cuốn sách".
 - NP: Cụm danh từ, chỉ đối tượng "cuốn sách mới".
 - NN (Noun): "cuốn", danh từ đơn vị.
 - NP: Cụm danh từ chỉ đối tượng "sách mới".
 - NN (Noun): "sách", danh từ chính.
 - Adj (Adjective): "mới", tính từ bổ nghĩa cho danh từ "sách".

3.1.5. Nam tặng Lan một cuốn sách rất hay



- Giải thích cấu trúc:

- S: Gốc của câu (Sentence), bao gồm cụm danh từ (NP) và cụm động từ (VP).
- NP: Cụm danh từ chủ ngữ, chứa:
 - NNP (Proper Noun): "Nam", danh từ riêng, người thực hiện hành động "tặng".
- VP: Cụm động từ chính, chứa:
 - VB (Verb): "tặng", động từ chính của câu.
 - NP: Cụm danh từ chỉ tân ngữ đầu tiên (người nhận quà), chứa:
 - NNP (Proper Noun): "Lan", danh từ riêng, người nhận món quà từ Nam.
 - NP: Cụm danh từ tân ngữ thứ hai (món quà được tặng), chứa:

- Q (Quantifier): "một", lượng từ chỉ số lượng cho danh từ "cuốn sách".
- NP: Cụm danh từ chỉ đối tượng "cuốn sách rất hay".
 - NN (Noun): "cuốn", danh từ đơn vị
 - NP: Cụm danh từ chỉ đối tượng "sách rất hay".
 - NN (Noun): "sách", danh từ chính.
 - AdjP (Adjective Phrase): Cụm tính từ, bổ nghĩa cho danh từ "sách".
 - RB (Adverb): "rất", trạng từ bổ nghĩa cho tính từ "hay".
 - Adj (Adjective): "hay", tính từ mô tả đặc điểm của "sách".

3.2 Chuyển sang văn phạm DCG cho từng câu

3.2.1. Nam thường đến thư viện

```

s(s(NP, VP)) --> np(NP), vp(VP).

np(np(NNP)) --> nnp(NNP).

np(np(NN)) --> nn(NN).

vp(vp(RB, VB, NP)) --> rb(RB), vb(VB), np(NP).

nnp(nnp('Nam')) --> ['Nam'].

vb(vb(den)) --> [den].

nn(nn('thu_vien')) --> ['thu_vien'].

rb(rb(thuong)) --> [thuong].

```

- **Kết quả phân tích câu:**

```
s(T, ['Nam', 'thuong', 'den', 'thu_vien'], []).
```

T = s(np(nnp('Nam')),vp(rb(thuong),vb(den),np(nn(thu_vien))))

Next 10 100 1,000 Stop

```
?- s(T, ['Nam', 'thuong', 'den', 'thu_vien'], []).
```

3.2.2. Nam rất thích đọc sách ở thư viện

s(s(NP, VP)) --> np(NP), vp(VP).

np(np(NNP)) --> nnp(NNP).

np(np(NN)) --> nn(NN).

vp(vp(RB, VB, VP)) --> rb(RB), vb(VB), vp(VP).

vp(vp(VB, NP, PP)) --> vb(VB), np(NP), pp(PP).

pp(pp(P, NP)) --> p(P), np(NP).

nnp(nnp('Nam')) --> ['Nam'].

vb(vb(thich)) --> [thich].

vb(vb(doc)) --> [doc].

rb(rb(rat)) --> [rat].

p(p(o)) --> [o].

nn(nn(sach)) --> [sach].

nn(nn(thu_vien)) --> [thu_vien].

- **Kết quả phân tích câu:**

```

s(T, ['Nam', rat, thích, doc, sach, o, thu_vien], []).

T = s(
  np(nnp('Nam')),
  vp(
    rb(rat),
    vb(thích),
    vp(vb(doc), np(nn(sach)), pp(p(o), np(nn(thu_vien)))))
  )
)

Next 10 100 1,000 Stop

?- s(T, ['Nam', rat, thích, doc, sach, o, thu_vien], []).

```

3.2.3. Nhà của Nam ở gần trường

$s(s(NP, VP)) \rightarrow np(NP), vp(VP).$
 $np(np(NN, PP)) \rightarrow nn(NN), pp(PP).$
 $np(np(NNP)) \rightarrow nnp(NNP).$
 $np(np(NN)) \rightarrow nn(NN).$
 $pp(pp(P, NP)) \rightarrow p(P), np(NP).$
 $vp(vp(P, AdjP)) \rightarrow p(P), adjp(AdjP).$
 $adjp(adjp(Adj, NP)) \rightarrow adj(Adj), np(NP).$
 $nnp(nnp('Nam')) \rightarrow ['Nam'].$
 $nn(nn(nha)) \rightarrow [nha].$
 $nn(nn(truong)) \rightarrow [truong].$
 $p(p(cua)) \rightarrow [cua].$
 $p(p(o)) \rightarrow [o].$
 $adj(adj(gan)) \rightarrow [gan].$

- **Kết quả phân tích câu:**

```

s(T, [nha, cua, 'Nam', o, gan, truong], []).
T = s(
  np(nn(nha),pp(p(cua),np(nnp('Nam')))),
  vp(p(o),adjp(adj(gan),np(nn(truong))))
)
Next 10 100 1,000 Stop
?- s(T, [nha, cua, 'Nam', o, gan, truong], []).

```

3.2.4. Nam mới mua mấy cuốn sách mới

$s(s(NP, VP)) \rightarrow np(NP), vp(VP).$
 $np(np(NNP)) \rightarrow nnp(NNP).$
 $np(np(Q, NP)) \rightarrow q(Q), np(NP).$
 $np(np(NN, NP)) \rightarrow nn(NN), np(NP).$
 $np(np(NN, Adj)) \rightarrow nn(NN), adj(Adj).$
 $np(np(NN)) \rightarrow nn(NN).$
 $vp(vp(RB, VB, NP)) \rightarrow rb(RB), vb(VB), np(NP).$
 $nnp(nnp('Nam')) \rightarrow ['Nam'].$
 $rb(rb(moi)) \rightarrow [moi].$
 $vb(vb(mua)) \rightarrow [mua].$
 $q(q(may)) \rightarrow [may].$
 $nn(nn(cuon)) \rightarrow [cuon].$
 $nn(nn(sach)) \rightarrow [sach].$
 $adj(adj(moi)) \rightarrow [moi].$

- **Kết quả phân tích câu:**

```
s(T, ['Nam', moi, mua, may, cuon, sach, moi], []).
```

```
T = s(
  np(nnp('Nam')),
  vp(
    rb(moi),
    vb(mua),
    np(q(may), np(nn(cuon), np(nn(sach), adj(moi))))
  )
)
```

Next 10 100 1,000 Stop

```
?- s(T, ['Nam', moi, mua, may, cuon, sach, moi], []).
```

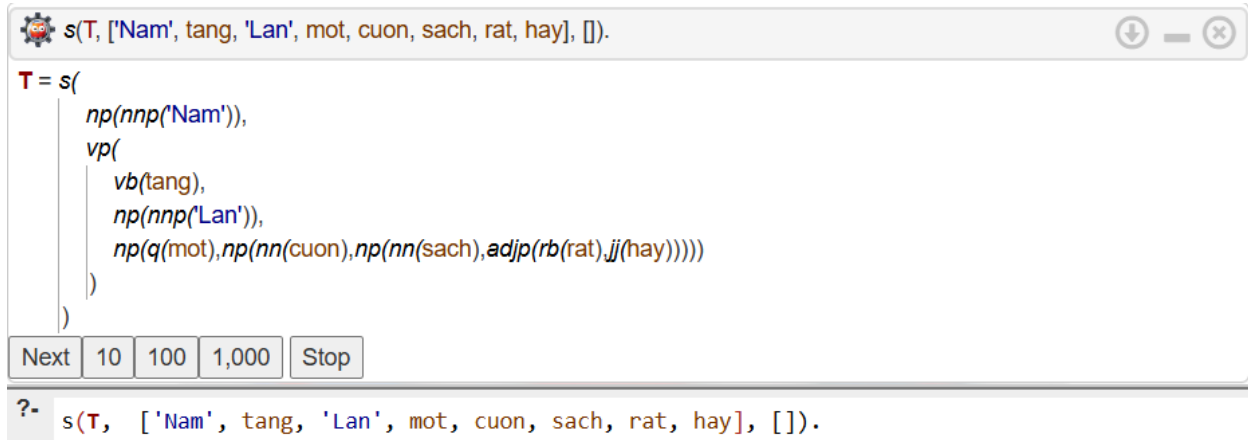
3.2.5. Nam tặng Lan một cuốn sách rất hay

$s(s(NP, VP)) \rightarrow np(NP), vp(VP).$
 $np(np(NNP)) \rightarrow nnp(NNP).$
 $np(np(Q, NP)) \rightarrow q(Q), np(NP).$
 $np(np(NN, AdjP)) \rightarrow nn(NN), adjp(AdjP).$
 $np(np(NN)) \rightarrow nn(NN).$
 $np(np(NN, NP)) \rightarrow nn(NN), np(NP).$
 $vp(vp(VB, NP1, NP2)) \rightarrow vb(VB), np(NP1), np(NP2).$
 $adjp(adjp(RB, JJ)) \rightarrow rb(RB), jj(JJ).$
 $nnp(nnp('Nam')) \rightarrow ['Nam'].$
 $nnp(nnp('Lan')) \rightarrow ['Lan'].$
 $vb(vb(tang)) \rightarrow [tang].$
 $q(q(mot)) \rightarrow [mot].$
 $nn(nn(cuon)) \rightarrow [cuon].$
 $nn(nn(sach)) \rightarrow [sach].$

$rb(rb(rat)) \rightarrow [rat].$

$jj(jj(hay)) \rightarrow [hay].$

- **Kết quả phân tích câu:**



```
s(T, ['Nam', tang, 'Lan', mot, cuon, sach, rat, hay], []).  
  
T = s(  
  np(nnp('Nam')),  
  vp(  
    vb(tang),  
    np(nnp('Lan')),  
    np(q(mot), np(nn(cuon), np(nn(sach), adjp(rb(rat), jj(hay)))))  
  )  
)  
  
Next 10 100 1,000 Stop  
?- s(T, ['Nam', tang, 'Lan', mot, cuon, sach, rat, hay], []).
```

3.3 Gộp văn phạm DCG và thực thi:

Từ những văn phạm của các câu trên, ta tiến hành gộp chúng lại để tạo thành 1 công cụ phân tích cú pháp hoàn chỉnh

- **Công cụ phân tích cú pháp hoàn chỉnh:**

$s(s(NP, VP)) \rightarrow np(NP), vp(VP).$
 $s(s(NP, AdjP)) \rightarrow np(NP), adjp(AdjP).$
 $np(np(NNP)) \rightarrow nnp(NNP).$
 $np(np(NN)) \rightarrow nn(NN).$
 $np(np(NN, PP)) \rightarrow nn(NN), pp(PP).$
 $np(np(Q, NP)) \rightarrow q(Q), np(NP).$
 $np(np(NN, NP)) \rightarrow nn(NN), np(NP).$
 $np(np(NN, AdjP)) \rightarrow nn(NN), adjp(AdjP).$
 $np(np(Q, NN)) \rightarrow q(Q), nn(NN).$
 $np(np(NN, JJ)) \rightarrow nn(NN), jj(JJ).$

vp(vp(VB, NP)) --> vb(VB), np(NP).

vp(vp(VB, NP1, NP2)) --> vb(VB), np(NP1), np(NP2).

vp(vp(RB, VB, NP)) --> rb(RB), vb(VB), np(NP).

vp(vp(VB, NP, PP)) --> vb(VB), np(NP), pp(PP).

vp(vp(RB, VB, VP)) --> rb(RB), vb(VB), vp(VP).

vp(vp(VB, VP)) --> vb(VB), vp(VP).

vp(vp(VB, PP)) --> vb(VB), pp(PP).

vp(vp(P, AdjP)) --> p(P), adjp(AdjP).

pp(pp(P, NP)) --> p(P), np(NP).

adjp(adjp(RB, JJ)) --> rb(RB), jj(JJ).

adjp(adjp(Adj, NP)) --> adj(Adj), np(NP).

nnp(nnp('Nam')) --> ['Nam'].

nnp(nnp('Lan')) --> ['Lan'].

vb(vb(den)) --> [den].

vb(vb(thich)) --> [thich].

vb(vb(doc)) --> [doc].

vb(vb(mua)) --> [mua].

vb(vb(tang)) --> [tang].

q(q(may)) --> [may].

q(q(mot)) --> [mot].

nn(nn('thu_vien')) --> ['thu_vien'].

nn(nn(nha)) --> [nha].

nn(nn(truong)) --> [truong].

nn(nn(cuon)) --> [cuon].

nn(nn(sach)) --> [sach].

rb(rb(thuong)) --> [thuong].

rb(rb(rat)) --> [rat].

rb(rb(moi)) --> [moi].

p(p(o)) --> [o].

p(p(cua)) --> [cua].

adj(adj(gan)) --> [gan].

jj(jj(moi)) --> [moi].

jj(jj(hay)) --> [hay].

- Thực thi công cụ phân tích cú pháp đối với các câu có trong tập ngữ liệu cho trước:

- Nam thường đến thư viện:

```
s(T, ['Nam', thuong, den, 'thu_vien'], []).  
T = s(np(np('Nam')), vp(rb(thuong), vb(den), np(nn(thu_vien))))  
?- s(T, ['Nam', thuong, den, 'thu_vien'], []).
```

- Nam rất thích đọc sách ở thư viện:

```
s(T, ['Nam', rat, thích, doc, sach, o, 'thu_vien'], []).
```

```
T = s(
  np(nnp('Nam')),
  vp(
    rb(rat),
    vb(thích),
    vp(vb(doc), np(nn(sach)), pp(p(o), np(nn(thu_vien))))
  )
)
```

```
?- s(T, ['Nam', rat, thích, doc, sach, o, 'thu_vien'], []).
```

- Nhà của Nam ở gần trường:

```
s(T, ['nha', cua, 'Nam', o, gan, truong], []).
```

```
T = s(
  np(nn(nha), pp(p(cua), np(nnp('Nam')))),
  vp(p(o), adjp(adj(gan), np(nn(truong))))
)
```

```
?- s(T, ['nha', cua, 'Nam', o, gan, truong], []).
```

- Nam mới mua mấy cuốn sách mới:

```
s(T, ['Nam', moi, mua, may, cuon, sach, moi], []).
```

```
T = s(
  np(nnp('Nam')),
  vp(
    rb(moi),
    vb(mua),
    np(q(may), np(nn(cuon), np(nn(sach), jj(moi))))
  )
)
```

```
?- s(T, ['Nam', moi, mua, may, cuon, sach, moi], []).
```

- Nam tặng Lan một cuốn sách rất hay:

```
s(T, ['Nam', tang, 'Lan', mot, cuon, sach, rat, hay], []).
```

```
T = s(
  np(nnp('Nam')),
  vp(
    vb(tang),
    np(nnp('Lan')),
    np(q(mot), np(nn(cuon), np(nn(sach), adjp(rb(rat), jj(hay)))))
  )
)
```

```
?- s(T, ['Nam', tang, 'Lan', mot, cuon, sach, rat, hay], []).
```

- Thực thi công cụ phân tích cú pháp đối với các câu phái sinh có thể có trong tập ngữ liệu:

- Các câu phái sinh từ câu “Nam thường đến thư viện”:

Nam đến thư viện:

```
s(T, ['Nam', den, 'thu_vien'], []).
```

```
T = s(np(nnp('Nam')), vp(vb(den), np(nn(thu_vien)))))
```

```
?- s(T, ['Nam', den, 'thu_vien'], []).
```

- Các câu phái sinh từ câu “Nam rất thích đọc sách ở thư viện”:

Nam thích sách thư viện:

```
s(T, ['Nam', thích, sach, 'thu_vien'], []).
```

```
T = s(np(nnp('Nam')), vp(vb(thích), np(nn(sach), np(nn(thu_vien)))))
```

```
?- s(T, ['Nam', thích, sach, 'thu_vien'], []).
```

Nam thích đọc sách ở thư viện:

```

s(T, ['Nam', thích, doc, sach, o, 'thu_vien'], []).
T = s(
  np(nnp('Nam')),
  vp(vb(thích),vp(vb(doc),np(nn(sach))),pp(p(o),np(nn(thu_vien))))
)
?- s(T, ['Nam', thích, doc, sach, o, 'thu_vien'], []).

```

Nam rất thích đọc sách:

```

s(T, ['Nam', rat, thích, doc, sach], []).
T = s(np(nnp('Nam')),vp(rb(rat),vb(thích),vp(vb(doc),np(nn(sach)))))
?- s(T, ['Nam', rat, thích, doc, sach], []).

```

Nam đọc sách ở thư viện:

```

s(T, ['Nam', doc, sach, o, 'thu_vien'], []).
T = s(
  np(nnp('Nam')),
  vp(vb(doc),np(nn(sach),pp(p(o),np(nn(thu_vien)))))
)
?- s(T, ['Nam', doc, sach, o, 'thu_vien'], []).

```

Nam thích ở thư viện

```

s(T, ['Nam', thích, o, 'thu_vien'], []).
T = s(np(nnp('Nam')),vp(vb(thích),pp(p(o),np(nn(thu_vien)))))
?- s(T, ['Nam', thích, o, 'thu_vien'], []).

```

Nam rất thích thư viện

```

s(T, ['Nam', rat, thích, 'thu_vien'], []).
T = s(np(nnp('Nam')),vp(rb(rat),vb(thích),np(nn(thu_vien)))))
?- s(T, ['Nam', rat, thích, 'thu_vien'], []).

```

- Các câu phái sinh từ câu “Nhà của Nam ở gần trường”:

Nhà Nam ở gần trường

```

s(T, ['nha', 'Nam', o, gan, truong], []).

T = s(
  np(nn(nha),np(nnp('Nam'))),
  vp(p(o),adjp(adj(gan),np(nn(truong))))
)

?- s(T, ['nha', 'Nam', o, gan, truong], []).

```

Nam ở gần trường

```

s(T, ['Nam', o, gan, truong], []).

T = s(np(nnp('Nam')),vp(p(o),adjp(adj(gan),np(nn(truong)))))

?- s(T, ['Nam', o, gan, truong], []).

```

Nhà gần trường

```

s(T, ['nha', gan, truong], []).

T = s(np(nn(nha)),adjp(adj(gan),np(nn(truong)))))

?- s(T, ['nha', gan, truong], []).

```

Nhà của Nam gần trường:

```

s(T, ['nha', cua, 'Nam', gan, truong], []).

T = s(
  np(nn(nha),pp(p(cua),np(nnp('Nam')))),
  adjp(adj(gan),np(nn(truong)))
)

?- s(T, ['nha', cua, 'Nam', gan, truong], []).

```

- Các câu phải sinh từ câu “Nam mới mua mấy cuốn sách mới”:

Nam mua sách

```

s(T, ['Nam', mua, sach], []).

T = s(np(nnp('Nam')),vp(vb(mua),np(nn(sach)))))

?- s(T, ['Nam', mua, sach], []).

```

Nam mới mua sách

```

s(T, ['Nam', moi, mua, sach], []).
T = s(np(nnp('Nam')),vp(rb(moi),vb(mua),np(nn(sach))))
?- s(T, ['Nam', moi, mua, sach], []).

```

Nam mới mua cuốn sách

```

s(T, ['Nam', moi, mua, cuon, sach], []).
T = s(
  np(nnp('Nam')),
  vp(rb(moi),vb(mua),np(nn(cuon),np(nn(sach))))
)
?- s(T, ['Nam', moi, mua, cuon, sach], []).

```

Nam mua mấy cuốn sách

```

s(T, ['Nam', mua, may, cuon, sach], []).
T = s(
  np(nnp('Nam')),
  vp(vb(mua),np(q(may),np(nn(cuon),np(nn(sach))))
)
?- s(T, ['Nam', mua, may, cuon, sach], []).

```

Nam mua cuốn sách mới

```

s(T, ['Nam', mua, cuon, sach, moi], []).
T = s(np(nnp('Nam')),vp(vb(mua),np(nn(cuon),np(nn(sach)),jj(moi))))
?- s(T, ['Nam', mua, cuon, sach, moi], []).

```

Nam mới mua mấy cuốn sách

```

s(T, ['Nam', moi, mua, may, cuon, sach], []).
T = s(
  np(nnp('Nam')),
  vp(rb(moi),vb(mua),np(q(may),np(nn(cuon),np(nn(sach))))
)
?- s(T, ['Nam', moi, mua, may, cuon, sach], []).

```

Nam mới mua cuốn sách mới

```

s(T, ['Nam', moi, mua, cuon, sach, moi], []).

T = s(
  np(nnp('Nam')),
  vp(rb(moi), vb(mua), np(nn(cuon), np(nn(sach), jj(moi))))
)

?- s(T, ['Nam', moi, mua, cuon, sach, moi], []).

```

- Các câu phái sinh từ câu “Nam tặng Lan một cuốn sách rất hay”:

Nam tặng cuốn sách hay

```

s(T, ['Nam', tang, cuon, sach, hay], []).

T = s(np(nnp('Nam')), vp(vb(tang), np(nn(cuon), np(nn(sach), jj(hay)))))

?- s(T, ['Nam', tang, cuon, sach, hay], []).

```

Nam tặng Lan sách

```

s(T, ['Nam', tang, 'Lan', sach], []).

T = s(np(nnp('Nam')), vp(vb(tang), np(nnp('Lan')), np(nn(sach)))))

?- s(T, ['Nam', tang, 'Lan', sach], []).

```

Nam tặng cuốn sách

```

s(T, ['Nam', tang, cuon, sach], []).

T = s(np(nnp('Nam')), vp(vb(tang), np(nn(cuon), np(nn(sach)))))

?- s(T, ['Nam', tang, cuon, sach], []).

```

Nam tặng Lan cuốn sách hay

```

s(T, ['Nam', tang, 'Lan', cuon, sach, hay], []).

T = s(
  np(nnp('Nam')),
  vp(vb(tang), np(nnp('Lan')), np(nn(cuon), np(nn(sach), jj(hay)))))
)

?- s(T, ['Nam', tang, 'Lan', cuon, sach, hay], []).

```

Nam tặng Lan sách hay

```

s(T, ['Nam', tang, 'Lan', sach, hay], []).

T = s(
  np(nnp('Nam')),
  vp(vb(tang), np(nnp('Lan')), np(nn(sach), jj(hay)))
)

?- s(T, ['Nam', tang, 'Lan', sach, hay], []).

```

Nam tặng cuốn sách rất hay

```

s(T, ['Nam', tang, cuon, sach, rat, hay], []).

T = s(
  np(nnp('Nam')),
  vp(vb(tang), np(nn(cuon), np(nn(sach), adjp(rb(rat), jj(hay))))))
)

?- s(T, ['Nam', tang, cuon, sach, rat, hay], []).

```

Nam tặng Lan sách rất hay

```

s(T, ['Nam', tang, 'Lan', sach, rat, hay], []).

T = s(
  np(nnp('Nam')),
  vp(vb(tang), np(nnp('Lan')), np(nn(sach), adjp(rb(rat), jj(hay))))))
)

?- s(T, ['Nam', tang, 'Lan', sach, rat, hay], []).

```

Nam tặng sách hay

```

s(T, ['Nam', tang, sach, hay], []).

T = s(np(nnp('Nam')), vp(vb(tang), np(nn(sach), jj(hay))))

?- s(T, ['Nam', tang, sach, hay], []).

```

Nam tặng Lan cuốn sách

```

s(T, ['Nam', tang, 'Lan', cuon, sach], []).

T = s(
  np(nnp('Nam')),
  vp(vb(tang), np(nnp('Lan')), np(nn(cuon), np(nn(sach))))))
)

?- s(T, ['Nam', tang, 'Lan', cuon, sach], []).

```


Ngoài các câu phái sinh của riêng từng câu trên, khi kết hợp cả 5 câu trong tập ngữ liệu sẽ xuất hiện thêm nhiều câu phái sinh khác nhau:

Nam thường đến trường đọc sách

```
s(T, ['Nam', thuong, den, doc, sach], []).  
  
T = s(  
  np(nnp('Nam')),  
  vp(rb(thuong), vb(den), vp(vb(doc), np(nn(sach))))  
)  
  
?- s(T, ['Nam', thuong, den, doc, sach], []).
```

Nam rất thích đến thư viện

```
s(T, ['Nam', rat, thích, den, 'thu_vien'], []).  
  
T = s(  
  np(nnp('Nam')),  
  vp(rb(rat), vb(thích), vp(vb(den), np(nn(thu_vien))))  
)  
  
?- s(T, ['Nam', rat, thích, den, 'thu_vien'], []).
```

Nam thích đọc sách ở thư viện mới

```
s(T, ['Nam', thích, doc, sach, o, 'thu_vien', moi], []).  
  
T = s(  
  np(nnp('Nam')),  
  vp(  
    vb(thích),  
    vp(vb(doc), np(nn(sach), pp(p(o), np(nn(thu_vien), jj(moi)))))  
  )  
)  
  
?- s(T, ['Nam', thích, doc, sach, o, 'thu_vien', moi], []).
```

Nam thích đọc mấy cuốn sách ở thư viện

```

s(T, ['Nam', thích, doc, may, cuon ,sach, o, 'thu_vien'], []).

```

```

T = s(
  np(nnp('Nam')),
  vp(
    vb(thích),
    vp(
      vb(doc),
      np(q(may),np(nn(cuon),np(nn(sach),pp(p(o),np(nn(thu_vien)))))))
    )
  )
)

```

```

?- s(T, ['Nam', thích, doc, may, cuon ,sach, o, 'thu_vien'], []).

```

Nhà Nam gần trường thư viện

```

s(T, ['nha', cua, 'Nam', o, gan, truong,'thu_vien'], []).

```

```

T = s(
  np(nn(nha),pp(p(cua),np(nnp('Nam')))),
  vp(p(o),adjp(adj(gan),np(nn(truong),np(nn(thu_vien))))))
)

```

```

?- s(T, ['nha', cua, 'Nam', o, gan, truong,'thu_vien'], []).

```

Nam thường đến thư viện gần trường

```

s(T, ['Nam', thường, đến, 'thu_vien',gan, truong], []).

```

```

T = s(
  np(nnp('Nam')),
  vp(
    rb(thường),
    vb(đến),
    np(nn(thu_vien),adjp(adj(gan),np(nn(truong))))
  )
)

```

```

?- s(T, ['Nam', thường, đến, 'thu_vien',gan, truong], []).

```

Nam tặng Lan sách mới

```
s(T, ['Nam', tang, 'Lan', sach, moi], []).
```

```
T = s(  
  np(nnp('Nam')),  
  vp(vb(tang), np(nnp('Lan')), np(nn(sach), jj(moi)))  
)
```

```
?- s(T, ['Nam', tang, 'Lan', sach, moi], []).
```

Nam mua sách hay ở trường

```
s(T, ['Nam', mua, sach, hay, o, truong], []).
```

```
T = s(  
  np(nnp('Nam')),  
  vp(vb(mua), np(nn(sach), jj(hay)), pp(p(o), np(nn(truong))))  
)
```

```
?- s(T, ['Nam', mua, sach, hay, o, truong], []).
```

Nam đọc một cuốn hay sách ở thư viện

```
s(T, ['Nam', doc, mot, cuon, sach, hay, o, 'thu_vien'], []).
```

```
T = s(  
  np(nnp('Nam')),  
  vp(  
    vb(doc),  
    np(q(mot), np(nn(cuon), np(nn(sach), jj(hay)))),  
    pp(p(o), np(nn(thu_vien)))  
  )  
)
```

```
?- s(T, ['Nam', doc, mot, cuon, sach, hay, o, 'thu_vien'], []).
```

Nam tặng Lan sách mấy cuốn sách rất hay

```
s(T, ['Nam', tang, 'Lan', may, cuon, sach, rat, hay], []).
```

```
T = s(
  np(nnp('Nam')),
  vp(
    vb(tang),
    np(nnp('Lan')),
    np(q(may),np(nn(cuon),np(nn(sach),adjp(rb(rat),jj(hay))))))
  )
)
```

```
?- s(T, ['Nam', tang, 'Lan', may, cuon, sach, rat, hay], []).
```

Nam thường đến trường

```
s(T, ['Nam', thuong, den, truong], []).
```

```
T = s(np(nnp('Nam')),vp(rb(thuong),vb(den),np(nn(truong))))
```

```
?- s(T, ['Nam', thuong, den, truong], []).
```

Nhà Nam gần thư viện

```
s(T, ['nha', 'Nam', o, gan, 'thu_vien'], []).
```

```
T = s(
  np(nn(nha),np(nnp('Nam'))),
  vp(p(o),adjp(adj(gan),np(nn(thu_vien))))
)
```

```
?- s(T, ['nha', 'Nam', o, gan, 'thu_vien'], []).
```

Nam mới mua cuốn sách gần trường

```
s(T, ['Nam', moi, mua,cuon,sach, gan, truong], []).
```

```
T = s(
  np(nnp('Nam')),
  vp(
    rb(moi),
    vb(mua),
    np(nn(cuon),np(nn(sach),adjp(adj(gan),np(nn(truong))))))
  )
)
```

```
?- s(T, ['Nam', moi, mua,cuon,sach, gan, truong], []).
```

Nam thích đọc sách mới

```

s(T, ['Nam', thích, doc, sach, moi], []).
T = s(np(nnp('Nam')),vp(vb(thích),vp(vb(doc),np(nn(sach),jj(moi)))))
?- s(T, ['Nam', thích, doc, sach, moi], []).

```

Nam mới mua mấy cuốn sách rất hay

```

s(T, ['Nam', moi, mua,may,cuon,sach, rat, hay], []).
T = s(
  np(nnp('Nam')),
  vp(
    rb(moi),
    vb(mua),
    np(q(may),np(nn(cuon),np(nn(sach),adjp(rb(rat),jj(hay)))))
  )
)
?- s(T, ['Nam', moi, mua,may,cuon,sach, rat, hay], []).

```

Nam mới mua sách rất hay

```

s(T, ['Nam', moi, mua, sach, rat, hay], []).
T = s(
  np(nnp('Nam')),
  vp(rb(moi),vb(mua),np(nn(sach),adjp(rb(rat),jj(hay)))))
)
?- s(T, ['Nam', moi, mua, sach, rat, hay], []).

```

Nam tặng mấy cuốn sách hay

```

s(T, ['Nam', tang, may, cuon, sach , hay], []).
T = s(
  np(nnp('Nam')),
  vp(vb(tang),np(q(may),np(nn(cuon),np(nn(sach),jj(hay)))))
)
Next 10 100 1,000 Stop
?- s(T, ['Nam', tang, may, cuon, sach , hay], []).

```

Nhà của Nam gần thư viện

```
s(T, ['nha', cua, 'Nam', gan, 'thu_vien'], []).
```

```
T = s(  
  np(nn(nha),pp(p(cua),np(np('Nam')))),  
  adjp(adj(gan),np(nn(thu_vien)))  
)
```

```
?- s(T, ['nha', cua, 'Nam', gan, 'thu_vien'], []).
```

Nam thích đến thư viện gần trường

```
s(T, ['Nam', thích, den, 'thu_vien', gan, truong], []).
```

```
T = s(  
  np(np('Nam'),  
    vp(  
      vb(thích),  
      vp(vb(den),np(nn(thu_vien),adjp(adj(gan),np(nn(truong)))))  
    )  
  )  
)
```

```
?- s(T, ['Nam', thích, den, 'thu_vien', gan, truong], []).
```

Nam rất thích sách mới

```
s(T, ['Nam', rat, thích, sach, moi], []).
```

```
T = s(np(np('Nam')),vp(rb(rat),vb(thích),np(nn(sach),jj(moi))))
```

```
?- s(T, ['Nam', rat, thích, sach, moi], []).
```

CHƯƠNG 4: TỔNG KẾT

Trong đồ án này, tôi đã giới thiệu cấu trúc của công cụ phân tích cú pháp dựa trên ngữ pháp DCG và ngôn ngữ lập trình Prolog để phân tích cú pháp các câu tiếng Việt trong tập dữ liệu. Đồ án đã hoàn thành mục tiêu là giới thiệu một phương pháp xử lý ngôn ngữ tự nhiên tiếng Việt bằng DCG và Prolog, giúp người đọc có cái nhìn tổng quan về các phương pháp phân tích cú pháp và ứng dụng chúng trong xây dựng hệ thống xử lý ngôn ngữ tự nhiên.

Đồ án không chỉ dừng lại ở việc định nghĩa các quy tắc cú pháp mà còn triển khai và kiểm thử các quy tắc này thông qua các truy vấn trong Prolog, từ đó minh chứng tính chính xác của công cụ phân tích cú pháp trong việc tự động phân tích và xây dựng cây cú pháp cho câu tiếng Việt. Kết quả đạt được cho thấy công cụ có khả năng mở rộng, cho phép xử lý nhiều cấu trúc câu phức tạp và phong phú trong ngữ pháp tiếng Việt.

Ưu điểm:

- Đề tài đã hoàn thành mục tiêu đề ra.
- Sử dụng công cụ và phần mềm hỗ trợ giúp tiết kiệm thời gian và công sức nghiên cứu.

Nhược điểm:

- Chưa bao hàm đầy đủ các cấu trúc câu có thể phát sinh từ ngữ liệu DCG tổng hợp.
- Cần nghiên cứu thêm để giảm thời gian chạy và tối ưu mã nguồn

TÀI LIỆU THAM KHẢO

- [1] PGS. TS. Nguyễn Tuấn Đăng, Bài giảng Xử lý ngôn ngữ tự nhiên, ĐH Sài Gòn
- [2] <https://vi.wikipedia.org/wiki/Prolog>
- [3] <https://ongthovuive.wordpress.com/2016/04/22/xu-ly-ngon-ngu-tu-nhien-phan-1/>
(1/5/2023)
- [4] <https://ongthovuive.wordpress.com/2016/04/22/xu-ly-ngon-ngu-tu-nhien-phan-2/>
(1/5/2023)
- [5] <https://ongthovuive.wordpress.com/2016/04/22/xu-ly-ngon-ngu-tu-nhien-phan-3/>
(1/5/2023)