

**TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN**

**Môn: SEMINAR CHUYÊN ĐỀ**

**Chuyên đề : LẬP TRÌNH LOGIC VÀ ỨNG DỤNG**

*Đề tài : Xây dựng công cụ phân tích cú pháp*

**Lớp môn học: DKP1201**

**Năm học : 2024-2025**

**Sinh viên thực hiện:** Nguyễn Văn Ninh

**Mã số sinh viên:** 3121410374

**Lớp:** DCT1213

**Giáo viên hướng dẫn:** PGS.TS. Nguyễn Tuấn Đăng

***TP. HỒ CHÍ MINH, THÁNG 10 NĂM 2024***

[illegible]

## LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công nghệ thông tin, trường Đại học Sài Gòn đã tạo điều kiện cho em được tiếp cận và tìm hiểu để hoàn thành môn học lần này. Chúng em cũng xin chân thành cảm ơn Thầy PGS.TS. Nguyễn Tuấn Đăng đã giảng dạy những kiến thức cần thiết trong môn học, những kiến thức ấy giúp cho chúng em rất nhiều trong thời gian thực hiện tiểu luận môn học. Em xin cảm ơn Thầy đã luôn tận tình giảng dạy và trang bị những kiến thức cần thiết trong môn học.

Trong quá trình nghiên cứu và làm bài, em đã làm bài với nỗ lực cao nhất, nhưng do kiến thức cũng như kinh nghiệm thực tế còn nhiều hạn chế nên tiểu luận không thể tránh khỏi những sai sót nhất định, em rất mong nhận được sự cảm thông, chia sẻ và tận tình đóng góp chỉ bảo từ Thầy để em học hỏi được nhiều kỹ năng, kinh nghiệm và sẽ hoàn thành tốt hơn trong những bài sắp tới sắp tới.

Em xin chân thành cảm ơn thầy !

## Mục Lục

<b>I.</b>	<b>Giới thiệu đề tài .....</b>	<b>5</b>
1.	Giới thiệu .....	5
2.	Mục tiêu .....	5
3.	Phạm vi .....	6
4.	Đối tượng nghiên cứu .....	6
<b>II.</b>	<b>Lịch sử nghiên cứu phân tích cú pháp trong khoa học máy tính .....</b>	<b>6</b>
1.	Alan Mathison Turing (1936) .....	6
2.	Nguồn gốc của phân tích cú pháp trong toán học và logic .....	6
3.	Lý thuyết ngôn ngữ hình thức và công trình của Noam Chomsky (1956) .....	7
4.	Phát triển đầu tiên của các thuật toán phân tích cú pháp (1960s) .....	8
5.	Thập niên 1970: Phát triển của các công cụ phân tích cú pháp tự động .....	9
6.	Thập niên 1980: Sự phổ biến của các trình biên dịch và ngôn ngữ lập trình mới .....	10
7.	Thập niên 1990 và 2000: Phân tích cú pháp trong xử lý ngôn ngữ tự nhiên .....	11
8.	Thập niên 2010 và hiện tại: Học sâu và phân tích cú pháp .....	11
<b>III.</b>	<b>Cách tiếp cận phân tích cú pháp dựa trên luật và lập trình logic .....</b>	<b>11</b>
1.	Cú pháp dựa trên luật .....	12
1.1.	Khái niệm ngữ pháp dựa trên luật .....	12
1.2.	Syntax Tree .....	12
1.3.	Definite Clause Grammars .....	14
2.	Ngôn ngữ lập trình logic .....	14
2.1.	Prolog .....	15
2.2.	DCG Syntax .....	17
<b>IV.</b>	<b>Xây dựng công cụ phân tích cú pháp dựa trên luật .....</b>	<b>18</b>
1.	Dữ liệu .....	18
1.1.	Dữ liệu gốc .....	18
1.2.	Các câu phái sinh .....	18
1.3.	Các câu phái sinh từ các câu a,b,c,d,e .....	20
2.	Văn phạm DCG ( Definite Clause Grammar ) .....	32
2.1.	Bảng quy tắc cho văn phạm DCG .....	33
2.2.	Văn phạm DCG cho câu gốc và câu phái sinh .....	34
2.3.	Văn phạm DCG chung cho các câu a, b, c, d, e và phái sinh .....	48
<b>V.</b>	<b>Thử nghiệm .....</b>	<b>51</b>
1.	Cấu trúc thử nghiệm .....	51
2.	Thử nghiệm với Prolog .....	51

2.1.	Các câu gốc .....	51
2.2.	Các câu phái sinh từ a, b.....	65
2.3.	Các câu phái sinh từ câu a, c.....	70
2.4.	Các câu phái sinh từ câu a, d.....	87
2.5.	câu phái sinh từ câu a, e.....	101
VI.	Tài liệu tham khảo:.....	118

## I. Giới thiệu đề tài

### 1. Giới thiệu

Lập trình logic là một trong những lĩnh vực quan trọng của khoa học máy tính. Trong lập trình logic, chúng ta sử dụng các công cụ và phương pháp để tạo ra các chương trình có tính logic và khả năng suy luận. Đề tài "Lập trình logic và ứng dụng" tập trung vào việc tìm hiểu và áp dụng các kỹ thuật lập trình logic trong ngôn ngữ Prolog, đặc biệt là về Tiny-Parser và văn phạm DCG (Definite Clause Grammar). Cuối cùng, có thể xây dựng một Tiny-Parser dựa trên văn phạm DCG và ngôn ngữ lập trình Prolog để phân tích cú pháp một vài câu tiếng Việt và tất cả các câu phái sinh trong tập ngữ liệu.

### 2. Mục tiêu

Mục tiêu của bài tiểu luận này là xây dựng một Tiny-Parser dựa trên Văn phạm DCG để phân tích cú pháp. Parser sẽ cho phép phân tích các câu đơn giản tiếng Việt và trả về cây phân tích cú pháp tương ứng. Các chức năng cơ bản sẽ bao gồm:

1. Phân tích cú pháp các câu tiếng Việt và tất cả các câu phái sinh phát sinh trong tập ngữ liệu đã cho.
2. Trình bày cây phân tích cú pháp tương ứng.

Ngoài ra thông qua xây dựng Tiny-Parser ta sẽ tìm hiểu về DCG và cách xây dựng trên Prolog

### **3. Phạm vi**

Phạm vi của đề tài "Xây dựng Tiny-Parser dựa trên văn phạm DCG để phân tích cú pháp" bao gồm:

1. Tìm hiểu và nghiên cứu về ngôn ngữ Prolog và văn phạm DCG.
2. Xây dựng một bộ phân tích cú pháp đơn giản dựa trên văn phạm DCG.

### **4. Đối tượng nghiên cứu**

1. Ngôn ngữ Prolog
2. Văn phạm DCG, cây cú pháp
3. Phương pháp phân tích với một câu tiếng việt

## **II. Lịch sử nghiên cứu phân tích cú pháp trong khoa học máy tính**

### **1. Alan Mathison Turing (1936)**

Alan Mathison Turing OBE FRS (23 tháng 6 năm 1912 – 7 tháng 6 năm 1954) là một nhà toán học, logic học và mật mã học người Anh, được xem là một trong những nhà tiên phong của ngành khoa học máy tính và A.I (Trí tuệ nhân tạo). Phép thử Turing (Turing test) là một trong những cống hiến lớn nhất của ông trong ngành trí tuệ nhân tạo: thử thách này đặt ra câu hỏi rằng máy móc có khi nào đạt được ý thức và có thể suy nghĩ được hay không. Ông đã hình thức hóa khái niệm thuật toán và tính toán với máy Turing, đồng thời đưa ra phiên bản của "Turing", mà ngày nay được đông đảo công chúng chấp nhận, về luận đề Church-Turing, một luận đề nói rằng tất cả những gì tính được bằng thuật toán đều có thể tính được bằng máy Turing.

### **2. Nguồn gốc của phân tích cú pháp trong toán học và logic**

Toán logic là một phân nhánh của Toán học, tập trung vào việc nghiên cứu các quy luật logic và quy tắc suy luận. Nó liên quan đến việc sử dụng lý thuyết và quy tắc logic để giải quyết các vấn đề phức tạp. Không những thế, Toán logic đóng

vai trò quan trọng trong nhiều khía cạnh của cuộc sống như phân tích cú pháp về sau.

Phân tích cú pháp bắt đầu với sự phát triển của lý thuyết ngôn ngữ hình thức trong toán học và logic. Ý tưởng rằng ngôn ngữ có thể được biểu diễn thông qua các quy tắc hình thức đã tồn tại từ rất lâu, và các nhà toán học, như Leibniz và Boole, đã nghiên cứu cách biểu diễn logic dưới dạng các ngôn ngữ hình thức.

Một trong những nền tảng quan trọng nhất cho phân tích cú pháp là sự phát triển của lý thuyết ngôn ngữ hình thức và ngữ pháp hình thức vào giữa thế kỷ 20. Đây là những hệ thống quy tắc để mô tả cú pháp của một ngôn ngữ, đặc biệt là các ngôn ngữ lập trình.

### **3. Lý thuyết ngôn ngữ hình thức và công trình của Noam Chomsky (1956)**

**Avram Noam Chomsky** sinh ra trong một gia đình trí thức Nga gốc Do Thái. Ông sinh ngày 7-12-1928, tại East Oak Lane, vùng ngoại ô Philadelphia, bang Pennsylvania, Mỹ. Cha ông là William Chomsky, trốn quân dịch Sa hoàng và di cư khỏi Nga năm 1913, là một học giả tiếng Hebrew.

Trong quá trình nghiên cứu “Phương pháp ngôn ngữ học cấu trúc” của Harris (x. Harris, 1951) ông đã tìm ra phương pháp hình thức hoá chúng một cách triệt để. Từ 1955, ông đã hoàn thành công trình Cấu trúc logic của lý thuyết ngôn ngữ dày trên 900 trang [A, 1975a]. Chương Phương pháp phân tích biến đổi của sách này được đệ trình làm luận án tiến sĩ. Phương pháp biến đổi của N. Chomsky khác với phương pháp biến đổi của Z. Harris ở điểm căn bản sau: Quy tắc biến đổi của Harris là hai chiều, nghĩa là nếu cấu trúc A biến đổi thành cấu trúc B thì B cũng biến đổi thành A ( $A \leftrightarrow B$ ). Với Chomsky, chỉ có biến đổi một chiều.

Sự phát triển chính trong nghiên cứu ngôn ngữ hình thức và phân tích cú pháp bắt đầu với công trình của Noam Chomsky vào những năm 1950, đặc biệt là với bài báo nổi tiếng của ông năm 1956, “Three Models for the Description of Language”. Trong công trình này, Chomsky đã giới thiệu các khái niệm cơ bản của ngữ pháp phi ngữ cảnh (context-free grammar - CFG) và ngữ pháp có ngữ cảnh (context-sensitive grammar), cùng với các lớp ngôn ngữ hình thức khác nhau.

Ngữ pháp phi ngữ cảnh (CFG): Đây là loại ngữ pháp được sử dụng để mô tả cú pháp của hầu hết các ngôn ngữ lập trình hiện đại. Nó cho phép xác định cấu trúc phân cấp của một câu trong ngôn ngữ mà không phụ thuộc vào ngữ cảnh cụ thể.

Chomsky đã tạo ra phân cấp Chomsky, một phân loại các ngôn ngữ hình thức dựa trên mức độ phức tạp của ngữ pháp của chúng, bao gồm: ngôn ngữ chính quy, ngôn ngữ phi ngữ cảnh, ngôn ngữ có ngữ cảnh, và ngôn ngữ phi đệ quy. Công trình này đã trở thành nền tảng lý thuyết cho các nhà nghiên cứu trong khoa học máy tính, đặc biệt là trong phát triển trình biên dịch và các công cụ xử lý ngôn ngữ lập trình.

#### **4. Phát triển đầu tiên của các thuật toán phân tích cú pháp (1960s)**

Trong những năm 1960, các nhà khoa học máy tính đã bắt đầu phát triển các thuật toán phân tích cú pháp dựa trên ngữ pháp hình thức. Một số đóng góp chính trong giai đoạn này bao gồm:

##### **- Donald Knuth (1965)**

Donald Ervin Knuth (sinh ngày 10 tháng 1, năm 1938) là một nhà khoa học máy tính nổi tiếng hiện đang là giáo sư danh dự tại Đại học Stanford.

Knuth được biết đến nhiều nhất là tác giả của bộ sách Nghệ thuật lập trình máy tính (The Art of Computer Programming, TAOCPP), một trong những sách tham khảo được coi trọng nhất trong ngành khoa học máy tính. Ông được trao giải Turing năm 1974 cho bộ sách này. Ông đã tạo ra ngành phân tích thuật toán và đã đem lại nhiều cống hiến nền tảng cho ngành khoa học máy tính lý thuyết. Ông đã



tạo ra hệ thống sắp chữ TEX và hệ thống phát họa phông chữ METAFONT, và cũng là người khởi xướng khái niệm lập trình học thức (literate programming)

Knuth đã phát triển thuật toán phân tích cú pháp LR (Left-to-right, Rightmost derivation). Đây là một trong những phương pháp phân tích cú pháp từ dưới lên, có thể xử lý một lớp lớn hơn các ngữ pháp phi ngữ cảnh so với các phương pháp trước đó. LR parsing đặc biệt hữu ích trong việc xây dựng trình biên dịch và vẫn được sử dụng rộng rãi đến ngày nay.

#### - **Algol và Backus-Naur Form (BNF)**

Algol là một trong những ngôn ngữ lập trình đầu tiên có ảnh hưởng lớn và đã giúp định hình nhiều lý thuyết về ngôn ngữ lập trình và phân tích cú pháp. Backus-Naur Form (BNF), là một ký hiệu được sử dụng để mô tả cú pháp của ngôn ngữ lập trình hoặc các ngôn ngữ chính thức khác. Nó được phát triển bởi John Backus và Peter Naur. BNF có thể được mô tả như một ký hiệu siêu cú pháp cho ngữ pháp phi ngữ cảnh.

Dạng Backus–Naur được áp dụng ở bất cứ nơi nào cần mô tả chính xác về ngôn ngữ, chẳng hạn như trong các thông số kỹ thuật ngôn ngữ chính thức, trong các hướng dẫn và trong sách giáo khoa về lý thuyết ngôn ngữ lập trình. BNF có thể được sử dụng để mô tả các định dạng tài liệu, bộ hướng dẫn và giao thức truyền thông.

Theo thời gian, nhiều phần mở rộng và biến thể của ký hiệu Backus–Naur ban đầu đã được tạo ra; một số được định nghĩa chính xác, bao gồm dạng Backus–Naur mở rộng (EBNF) và dạng Backus–Naur tăng cường (ABNF).

### **5. Thập niên 1970: Phát triển của các công cụ phân tích cú pháp tự động**

Vào thập niên 1970, nhu cầu về các công cụ tự động hóa việc phân tích cú pháp đã thúc đẩy sự ra đời của nhiều phần mềm giúp tạo ra các bộ phân tích cú pháp. Một số công cụ nổi bật bao gồm:

**Yacc (Yet Another Compiler-Compiler):** Là phần mềm để phân tích ngữ pháp của phần mềm khác, dùng trong việc biên dịch từ ngôn ngữ bậc cao sang ngôn ngữ máy tính. Phần mềm này phân tích cách sắp xếp các ký tự được ghi trong ngôn ngữ bậc cao.

Được phát triển bởi Stephen C. Johnson vào năm 1970 tại Bell Labs, Yacc là một công cụ cho phép các nhà phát triển mô tả ngữ pháp của ngôn ngữ lập trình và tự động tạo ra bộ phân tích cú pháp. Yacc hỗ trợ phân tích cú pháp LALR (Lookahead LR), một biến thể của LR parsing có khả năng xử lý các ngữ pháp phi ngữ cảnh một cách hiệu quả hơn.

**Lex và Flex:** Là các công cụ dùng để tạo bộ phân tích từ vựng (lexer), thường kết hợp với Yacc để tạo thành các trình biên dịch hoàn chỉnh.

**LL Parsing:** Cũng trong thời gian này, các thuật toán phân tích cú pháp từ trên xuống như LL Parsing (Left-to-right, Leftmost derivation) cũng được phát triển, với công cụ nổi tiếng như Recursive Descent Parsing.

## **6. Thập niên 1980: Sự phổ biến của các trình biên dịch và ngôn ngữ lập trình mới**

Trong những năm 1980, sự phát triển của các ngôn ngữ lập trình hiện đại như C++, Ada, và Prolog đã thúc đẩy nghiên cứu về các kỹ thuật phân tích cú pháp phức tạp hơn.

**Công nghệ LL(k):** Các thuật toán LL Parsing cũng được mở rộng thành LL(k) để xử lý cú pháp phức tạp hơn với k token lookahead.

**ANTLR (Another Tool for Language Recognition):** Được phát triển vào cuối những năm 1980 bởi Terence Parr, ANTLR là một công cụ phân tích cú pháp

mạnh mẽ dựa trên LL(\*) Parsing, cho phép xử lý nhiều ngữ pháp phức tạp với hiệu suất cao.

## **7. Thập niên 1990 và 2000: Phân tích cú pháp trong xử lý ngôn ngữ tự nhiên**

Vào thập niên 1990 và 2000, sự phát triển của Xử lý ngôn ngữ tự nhiên (NLP) đã mở ra các hướng nghiên cứu mới về phân tích cú pháp. Phân tích cú pháp không chỉ còn tập trung vào ngôn ngữ lập trình mà còn được áp dụng rộng rãi để xử lý ngôn ngữ tự nhiên.

Các thuật toán thống kê: Các phương pháp thống kê và học máy bắt đầu được áp dụng vào phân tích cú pháp trong NLP. Các phương pháp này dựa trên việc học từ các tập dữ liệu lớn để xây dựng các mô hình phân tích cú pháp cho ngôn ngữ tự nhiên.

Cây phân tích cú pháp ngữ nghĩa (Semantic Parsing): Phân tích cú pháp trong NLP không chỉ dừng lại ở mức cú pháp mà còn mở rộng sang ngữ nghĩa, giúp máy tính hiểu được ý nghĩa của các câu văn trong ngôn ngữ tự nhiên.

## **8. Thập niên 2010 và hiện tại: Học sâu và phân tích cú pháp**

Với sự phát triển của học sâu (deep learning), các mô hình học máy như Transformer và BERT đã mở ra hướng mới trong việc phân tích cú pháp tự động. Các mô hình này có khả năng học các quy tắc ngôn ngữ từ dữ liệu lớn mà không cần phải xây dựng các ngữ pháp hình thức rõ ràng.

Phân tích cú pháp dựa trên mạng nơ-ron: Các mô hình học sâu có thể phân tích cú pháp hiệu quả bằng cách học các mẫu từ dữ liệu, từ đó cải thiện độ chính xác trong các ứng dụng xử lý ngôn ngữ tự nhiên như dịch máy và hỏi đáp tự động.

## **III. Cách tiếp cận phân tích cú pháp dựa trên luật và lập trình logic**

## 1. Cú pháp dựa trên luật

### 1.1. Khái niệm ngữ pháp dựa trên luật

Ngữ pháp dựa trên luật là cách tiếp cận phổ biến trong phân tích cú pháp, trong đó các câu được phân tích thông qua các quy tắc ngữ pháp (production rules). Mỗi quy tắc mô tả cách một cấu trúc ngữ pháp lớn hơn có thể được phân tích thành các thành phần nhỏ hơn. Ví dụ, một câu có thể được phân tích thành chủ ngữ và vị ngữ, và vị ngữ có thể được phân tích thành động từ và bổ ngữ.

Nguyên tắc cơ bản của ngữ pháp dựa trên luật:

Quy tắc ngữ pháp: Mỗi quy tắc mô tả một cách để phân tích một cấu trúc ngôn ngữ. Ví dụ, câu có thể được phân tích thành chủ ngữ và vị ngữ:

- $S \rightarrow NP + VP$  Trong đó:
- S (Sentence): Câu
- NP (Noun Phrase): Cụm danh từ (Chủ ngữ)
- VP (Verb Phrase): Cụm động từ (Vị ngữ)

Các quy tắc ngữ pháp phải định nghĩa rõ ràng các thành phần như danh từ (Noun), động từ (Verb), tính từ (Adjective), giới từ (Preposition), cụm danh từ (Noun Phrase), cụm động từ (Verb Phrase), và các cấu trúc khác trong ngôn ngữ.

### 1.2. Syntax Tree

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Vì hầu hết các thông tin quan trọng được viết ra bằng các ngôn ngữ tự nhiên như tiếng Anh, tiếng Pháp, tiếng Đức, ... nên NLP giúp máy tính giao tiếp với con người bằng ngôn ngữ của chúng và thực hiện các tác vụ liên quan đến ngôn ngữ

khác. Tóm lại, NLP giúp máy tính có thể đọc văn bản, nghe lời nói, diễn giải và nhận ra nó, hiểu tình cảm và xác định các phần quan trọng của văn bản hoặc bài phát biểu.

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

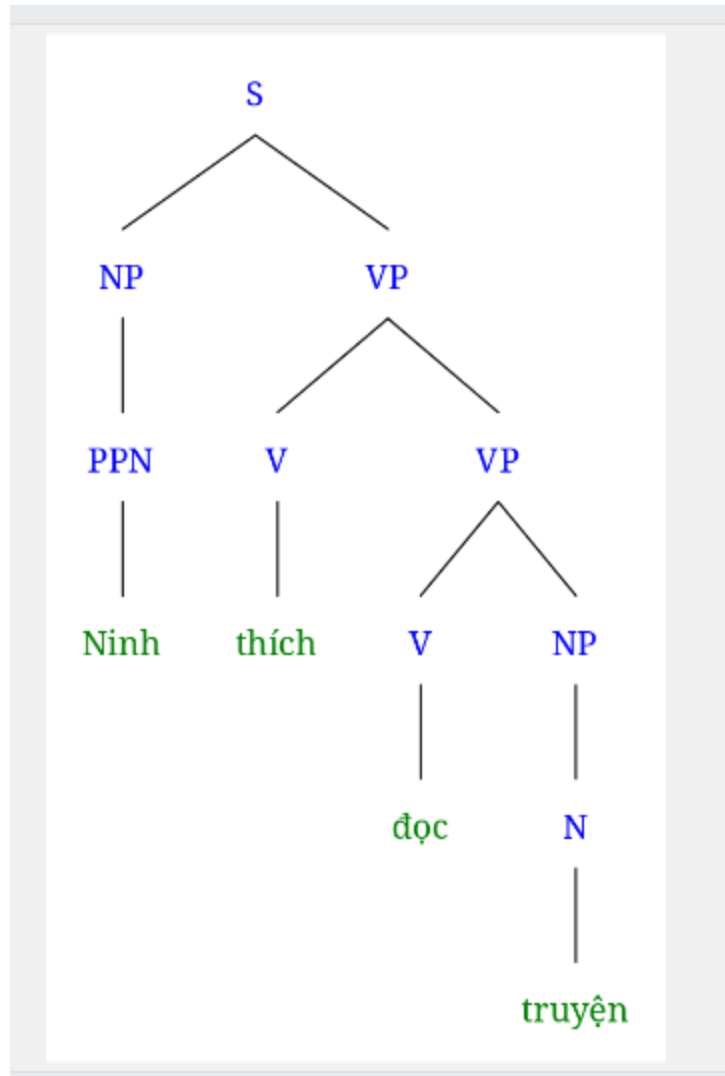
- Câu.
- Mệnh đề.
- Cụm từ.
- Từ.

Cú pháp (Syntax) đề cập đến tập hợp các quy tắc, nguyên tắc và quá trình điều chỉnh cấu trúc câu trong ngôn ngữ tự nhiên. Một mô tả cơ bản về cú pháp là cách các từ khác nhau như Chủ ngữ, Động từ, Danh từ, Cụm danh từ, vv. được sắp xếp trong một câu. Một số danh mục cú pháp của một ngôn ngữ tự nhiên như sau:

- Cụm danh từ (Noun Phrase - NP).
- Cụm động từ (Verb Phrase - VP).
- Cụm giới từ (Prepositional Phrase - PP).
- Động từ (Verb - V).
- Danh từ chung (Noun - N).
- Dang từ riêng (Proper N ).

Cây cú pháp (Syntax Tree) hay còn gọi là cây phân tích (parse tree) là một biểu diễn cây của các loại cú pháp khác nhau trong một câu. Nó giúp chúng ta hiểu cấu trúc cú pháp của một câu.

Ví dụ cây cú pháp cho câu Ninh thích đọc truyện:



Hình 1.1: Ví dụ cây cú pháp

### 1.3. Definite Clause Grammars

Ngữ pháp mệnh đề xác định ( DCG ) là một cách thể hiện ngữ pháp, cho ngôn ngữ tự nhiên hoặc ngôn ngữ hình thức , trong ngôn ngữ lập trình logic như Prolog . Nó liên quan chặt chẽ đến khái niệm ngữ pháp thuộc tính / ngữ pháp tiền tố . DCG thường được liên kết với Prolog, nhưng các ngôn ngữ tương tự như Mercury cũng bao gồm DCG. Chúng được gọi là ngữ pháp mệnh đề xác định vì chúng biểu diễn một ngữ pháp dưới dạng một tập hợp các mệnh đề xác định trong logic bậc nhất .

## 2. Ngôn ngữ lập trình logic

Lập trình logic là một mô hình lập trình , cơ sở dữ liệu và biểu diễn tri thức dựa trên logic hình thức . Một chương trình logic là một tập hợp các câu dưới dạng logic, biểu diễn tri thức về một số miền vấn đề. Tính toán được thực hiện bằng cách áp dụng lý luận logic vào tri thức đó, để giải quyết các vấn đề trong miền đó. Các họ ngôn ngữ lập trình logic chính bao gồm Prolog , Answer Set Programming (ASP) và Datalog .

## 2.1. Prolog

Prolog còn được gọi là ngôn ngữ lập trình ký hiệu (symbolic programming) tương tự các ngôn ngữ lập trình hàm (functional programming), hay lập trình phi số (nonnumerical programming). Prolog rất thích hợp để giải quyết các bài toán liên quan đến các đối tượng (object) và mối quan hệ (relation) giữa chúng.

Prolog được sử dụng nhiều trong các ứng dụng của trí tuệ nhân tạo và ngôn ngữ học trong khoa học máy tính (đặc biệt là trong ngành xử lý ngôn ngữ tự nhiên vì đây là mục tiêu thiết kế ban đầu của nó). Cú pháp và ngữ nghĩa của Prolog đơn giản và sáng sủa, nó được người Nhật coi là một trong những nền tảng để xây dựng máy tính thế hệ thứ năm mà ở đó, thay vì phải mô tả cách giải quyết một bài toán trên máy tính, con người chỉ cần mô tả bài toán và máy tính sẽ hỗ trợ họ nốt phần còn lại.

Nguyên lý lập trình logic dựa trên các mệnh đề Horn (Horn logic). Một mệnh đề Horn biểu diễn một sự kiện hay một sự việc nào đó là đúng hoặc không đúng, xảy ra hoặc không xảy ra (có hoặc không có, v.v...).

Chương trình Prolog thường bao gồm ba thành phần chính:

**Facts** (Sự kiện): Xác định những sự thật cụ thể.

Sự kiện trong Prolog được sử dụng để mô tả các mối quan hệ hoặc các đặc điểm của một thực thể. Một sự kiện là một mệnh đề logic đơn giản và luôn đúng. Chúng có dạng:

Prolog
parent(jonh, mary). parent(mary, susan).

**Rules** (Quy tắc): Xác định các quan hệ điều kiện giữa các sự kiện.

Quy tắc cho phép bạn định nghĩa các mối quan hệ phụ thuộc điều kiện, dưới dạng điều kiện "nếu ... thì". Một quy tắc có dạng:

Prolog
grandparent(X,Y):- parent(X,Z), parent(Z,Y).

**Queries** (Truy vấn): Đưa ra câu hỏi để Prolog tìm lời giải dựa trên các sự kiện và quy tắc đã định nghĩa.

Các truy vấn được sử dụng để kiểm tra xem một mệnh đề có đúng hay không dựa trên các sự kiện và quy tắc đã được định nghĩa. Một truy vấn có thể trả về:

true: Nếu Prolog tìm thấy lời giải thoả mãn truy vấn.

false: Nếu Prolog không tìm thấy lời giải.

Prolog
?- parent(jonh,mary). true.
?- grandparent(jonh,susan). true.
?- grandparent(jonh,susans). false.



## 2.2. DCG Syntax

Definite Clause Grammar (DCG) là một tiền xử lý trình thông dịch Prolog, nhận vào các quy tắc ngữ pháp DCG và thêm các danh sách khác biệt được liên kết vào các mục tiêu. DCG cung cấp cú pháp để viết các quy tắc phân tích cú pháp ngữ pháp dễ đọc hơn mà không bao gồm danh sách khác biệt liên kết. Tiền xử lý DCG chuyển đổi quy tắc DCG thành Prolog thuần, bổ sung danh sách khác biệt liên kết. Cú pháp DCG được biểu diễn như sau:

- Toán tử (`-->`) chỉ ra một quy tắc DCG, thay thế cho toán tử (`:-`) được sử dụng cho các mệnh đề Prolog. Ví dụ: `sentence --> subject, verb, object.`
- Mỗi mục tiêu được cho là đề cập đến phần đầu của một quy tắc DCG, và tiền xử lý thêm hai đối số khác nhau cho danh sách khác biệt.
- Ngoặc nhọn `{}` được sử dụng để cô lập các mục tiêu Prolog bình thường khỏi trình tiền xử lý DCG. Ví dụ: `subject -- modifier, noun, {write('found subject')}`.
- Ngoặc vuông `[]`, ký hiệu danh sách, được sử dụng để chỉ định các ký hiệu cuối của ngữ pháp. Ví dụ: `noun --> [cat].`

Sử dụng ký hiệu DCG, các quy tắc phân tích câu của phần trước có thể được viết lại như sau:

Prolog
<pre>% quy tac cho cau s --&gt; np,vp. % cum danh tu np --&gt; ppn. np --&gt; n. % cum dong tu vp --&gt; v,vp.</pre>

vp --> v,np.

% them du lieu

ppn --> [ninh].

v --> [thich].

v --> [doc].

n --> [truyen].

Kết quả :

Prolog

?- s([ninh,thich,doc,truyen],[]).

true

#### IV. Xây dựng công cụ phân tích cú pháp dựa trên luật

##### 1. Dữ liệu

##### 1.1. Dữ liệu gốc

- a. Nam thường đến thư viện.
- b. Nam rất thích đọc sách ở thư viện.
- c. Nhà của Nam ở gần trường.
- d. Nam mới mua mấy cuốn sách mới.
- e. Nam tặng Lan một cuốn sách rất hay.

##### 1.2. Các câu phái sinh

Các câu phái sinh từ tập dữ liệu gốc có thể được tạo ra bằng cách thay đổi vị trí các thành phần, nhưng vẫn giữ nguyên cấu trúc cú pháp cơ bản.

Câu a: Nam thường đến thư viện

Các câu phái sinh có thể là:

- Nam đến thư viện.
- Nam đến thư viện thường.
- Thư viện Nam đến.
- Thư viện Nam thường đến.

Câu b: Nam rất thích đọc sách ở thư viện.

Các câu phái sinh có thể là:

- Nam thích đọc sách ở thư viện.
- Nam đọc sách ở thư viện.
- Nam rất thích sách ở thư viện.
- Nam thích sách ở thư viện
- Nam rất thích ở thư viện.
- Nam thích ở thư viện.
- Nam thích sách.
- Nam rất thích sách.
- Nam thích đọc sách.
- Nam rất thích đọc sách.

Câu c: Nhà của Nam ở gần trường.

Các câu phái sinh có thể là:

- Nhà Nam ở gần trường.
- Nhà Nam gần trường.
- Nhà của Nam gần trường.
- Nhà của Nam ở trường
- Nhà Nam ở trường
- Nhà ở gần trường Nam
- Nhà gần trường Nam

Câu d: Nam mới mua mấy cuốn sách mới.

Các câu phái sinh có thể là:

- Nam mua mấy cuốn sách mới.
- Nam mới mua mấy cuốn sách.
- Nam mua cuốn sách mới.
- Nam mới mua cuốn sách.
- Nam mua cuốn sách.
- Nam mua sách.
- Nam mua sách mới.
- Nam mới mua sách.
- Nam mới mua sách mới.

Câu e: Nam tặng Lan một cuốn sách rất hay.

Các câu phái sinh có thể là:

- Nam tặng Lan cuốn sách.
- Nam tặng Lan một cuốn sách.
- Nam tặng Lan cuốn sách rất hay.
- Lan tặng Nam cuốn sách.
- Lan tặng Nam một cuốn sách.
- Lan tặng Nam cuốn sách rất hay.
- Nam tặng Lan sách
- Nam tặng Lan sách rất hay.
- Lan tặng Nam sách.
- Lan tặng Nam sách rất hay.

### **1.3. Các câu phái sinh từ các câu a,b,c,d,e**

Các câu phái sinh từ câu a, b :

- Nam đến đọc sách ở thư viện
- Nam thích đến đọc sách ở thư viện
- Nam rất thích đến đọc sách ở thư viện
- Nam thường thích đến thư viện
- Nam thường rất thích đến thư viện
- Nam thường đến đọc sách ở thư viện
- Nam thường thích đọc sách ở thư viện
- Nam thường rất thích đọc sách ở thư viện
- Nam thường rất thích sách ở thư viện
- Nam thường thích sách ở thư viện
- Nam thích đến ở thư viện
- Nam rất thích đến ở thư viện
- Nam thường thích đến ở thư viện
- Nam thường rất thích đến ở thư viện
- Nam thường thích sách
- Nam thường rất thích sách
- Nam thường đọc sách
- Nam thường thích đọc sách
- Nam thường rất thích đọc sách

Các câu phái sinh từ câu a, c :

- Nam đến thư viện ở gần trường
- Nam đến thư viện gần trường
- Nam đến thư viện trường
- Nam đến nhà thư viện
- Nam đến nhà trường
- Nam đến gần trường

- Nam đến ở gần trường.
- Nam đến ở trường
- Nam thường đến thư viện trường
- Nam thường đến thư viện ở trường
- Nam thường đến gần trường
- Nam thường đến ở trường
- Nam thường đến trường
- Nam thường ở trường
- Nam thường ở gần trường
- Nam thường đến ở gần trường
- Nam thường đến nhà thư viện
- Nam thường đến nhà trường
- Nam thường đến nhà
- Nam thường ở nhà
- Nam thường ở thư viện
- Nam thường ở nhà thư viện
- Nam thường ở nhà trường
- Nam ở nhà
- Nam ở thư viện
- Nam ở trường
- Nam ở nhà thư viện
- Nam ở nhà trường
- Nam gần trường
- Nam gần thư viện
- Nam gần nhà
- Nhà Nam ở gần thư viện
- Nhà Nam gần thư viện

- Nhà Nam thường ở gần thư viện
- Nhà Nam thường gần thư viện
- Nhà Nam đến thư viện
- Nhà Nam đến trường
- Nhà Nam đến gần thư viện
- Nhà Nam đến gần trường
- Nhà Nam đến ở gần thư viện
- Nhà Nam đến ở gần trường
- Nhà Nam thường đến thư viện
- Nhà Nam thường đến trường
- Nhà Nam thường đến gần thư viện
- Nhà Nam thường đến gần trường
- Nhà Nam thường đến ở gần thư viện
- Nhà Nam thường đến ở gần trường
- Nhà của Nam ở gần thư viện
- Nhà của Nam gần thư viện
- Nhà của Nam thường ở gần thư viện
- Nhà của Nam thường gần thư viện
- Nhà của Nam đến thư viện
- Nhà của Nam đến trường
- Nhà của Nam đến gần thư viện
- Nhà của Nam đến gần trường
- Nhà của Nam đến ở gần thư viện
- Nhà của Nam đến ở gần trường
- Nhà của Nam thường đến thư viện
- Nhà của Nam thường đến trường
- Nhà của Nam thường đến gần thư viện

- Nhà của Nam thường đến gần trường
- Nhà của Nam thường đến ở gần thư viện
- Nhà của Nam thường đến ở gần trường

Các câu phái sinh từ câu a, d :

- Nam đến thư viện mua sách
- Nam đến thư viện mua cuốn sách
- Nam đến thư viện mua mấy cuốn sách
- Nam đến thư viện mua sách mới
- Nam đến thư viện mua cuốn sách mới
- Nam đến thư viện mua mấy cuốn sách mới
- Nam mua sách ở thư viện
- Nam mua cuốn sách ở thư viện
- Nam mua mấy cuốn sách ở thư viện
- Nam mua sách mới ở thư viện
- Nam mua cuốn sách mới ở thư viện
- Nam mua mấy cuốn sách mới ở thư viện
- Nam ở thư viện mua sách
- Nam ở thư viện mua cuốn sách
- Nam ở thư viện mua mấy cuốn sách
- Nam ở thư viện mua sách mới
- Nam ở thư viện mua cuốn sách mới
- Nam ở thư viện mua mấy cuốn sách mới
- Nam thường đến thư viện mua sách
- Nam thường đến thư viện mua cuốn sách
- Nam thường đến thư viện mua mấy cuốn sách
- Nam thường đến thư viện mua sách mới



- Nam thường đến thư viện mua cuốn sách mới
- Nam thường đến thư viện mua mấy cuốn sách mới
- Nam thường mua sách ở thư viện
- Nam thường mua cuốn sách ở thư viện
- Nam thường mua mấy cuốn sách ở thư viện
- Nam thường mua sách mới ở thư viện
- Nam thường mua cuốn sách mới ở thư viện
- Nam thường mua mấy cuốn sách mới ở thư viện
- Nam thường ở thư viện mua sách
- Nam thường ở thư viện mua cuốn sách
- Nam thường ở thư viện mua mấy cuốn sách
- Nam thường ở thư viện mua sách mới
- Nam thường ở thư viện mua cuốn sách mới
- Nam thường ở thư viện mua mấy cuốn sách mới
- Nam mới mua sách ở thư viện
- Nam mới mua cuốn sách ở thư viện
- Nam mới mua mấy cuốn sách ở thư viện
- Nam mới mua sách mới ở thư viện
- Nam mới mua cuốn sách mới ở thư viện
- Nam mới mua mấy cuốn sách mới ở thư viện
- Nam mới ở thư viện mua sách
- Nam mới ở thư viện mua cuốn sách
- Nam mới ở thư viện mua mấy cuốn sách
- Nam mới ở thư viện mua sách mới
- Nam mới ở thư viện mua cuốn sách mới
- Nam mới ở thư viện mua mấy cuốn sách mới
- Nam mới đến thư viện mua sách

- Nam mới đến thư viện mua cuốn sách
- Nam mới đến thư viện mua mấy cuốn sách
- Nam mới đến thư viện mua sách mới
- Nam mới đến thư viện mua cuốn sách mới
- Nam mới đến thư viện mua mấy cuốn sách mới

Các câu phái sinh từ câu a, e :

- Nam đến thư viện tặng lan cuốn sách
- Nam đến thư viện tặng lan cuốn sách hay
- Nam đến thư viện tặng lan cuốn sách rất hay
- Nam đến thư viện tặng lan một cuốn sách
- Nam đến thư viện tặng lan một cuốn sách hay
- Nam đến thư viện tặng lan một cuốn sách rất hay
- Cuốn sách Nam tặng Lan ở thư viện
- Cuốn sách hay Nam tặng Lan ở thư viện
- Cuốn sách rất hay Nam tặng Lan ở thư viện
- Cuốn sách rất hay Nam đến tặng Lan
- Cuốn sách hay Nam đến tặng Lan
- Cuốn sách rất hay Nam đến tặng Lan
- Cuốn sách rất hay Nam đến tặng Lan ở thư viện
- Cuốn sách hay Nam đến tặng Lan ở thư viện
- Cuốn sách rất hay Nam đến tặng Lan ở thư viện
- Lan đến thư viện tặng Nam cuốn sách
- Lan đến thư viện tặng Nam cuốn sách hay
- Lan đến thư viện tặng Nam cuốn sách rất hay
- Lan đến thư viện tặng Nam một cuốn sách
- Lan đến thư viện tặng Nam một cuốn sách hay

- Lan đến thư viện tặng Nam một cuốn sách rất hay
- Cuốn sách Lan tặng Nam ở thư viện
- Cuốn sách hay Lan tặng Nam ở thư viện
- Cuốn sách rất hay Lan tặng Nam ở thư viện
- Cuốn sách rất hay Lan đến tặng Nam
- Cuốn sách hay Lan đến tặng Nam
- Cuốn sách rất hay Lan đến tặng Nam
- Cuốn sách rất hay Lan đến tặng Nam ở thư viện
- Cuốn sách hay Lan đến tặng Nam ở thư viện
- Cuốn sách rất hay Lan đến tặng Nam ở thư viện
- Nam thường đến thư viện tặng Lan cuốn sách
- Nam thường đến thư viện tặng Lan cuốn sách hay
- Nam thường đến thư viện tặng Lan cuốn sách rất hay
- Nam thường đến thư viện tặng Lan một cuốn sách
- Nam thường đến thư viện tặng Lan một cuốn sách hay
- Nam thường đến thư viện tặng Lan một cuốn sách rất hay
- Cuốn sách Nam thường tặng Lan ở thư viện
- Cuốn sách hay Nam thường tặng Lan ở thư viện
- Cuốn sách rất hay Nam thường tặng Lan ở thư viện
- Cuốn sách rất hay Nam thường đến tặng Lan
- Cuốn sách hay Nam thường đến tặng Lan
- Cuốn sách rất hay Nam thường đến tặng Lan
- Cuốn sách rất hay Nam thường đến tặng Lan ở thư viện
- Cuốn sách hay Nam thường đến tặng Lan ở thư viện
- Cuốn sách rất hay Nam thường đến tặng Lan ở thư viện
- Lan thường đến thư viện tặng Nam cuốn sách
- Lan thường đến thư viện tặng Nam cuốn sách hay

- Lan thường đến thư viện tặng Nam cuốn sách rất hay
- Lan thường đến thư viện tặng Nam một cuốn sách
- Lan thường đến thư viện tặng Nam một cuốn sách hay
- Lan thường đến thư viện tặng Nam một cuốn sách rất hay
- Cuốn sách Lan thường tặng Nam ở thư viện
- Cuốn sách hay Lan thường tặng Nam ở thư viện
- Cuốn sách rất hay Lan thường tặng Nam ở thư viện
- Cuốn sách ở thư viện Lan thường tặng Nam
- Cuốn sách hay ở thư viện Lan thường tặng Nam
- Cuốn sách rất hay ở thư viện Lan thường tặng Nam
- Cuốn sách rất hay Lan thường đến tặng Nam
- Cuốn sách hay Lan thường đến tặng Nam
- Cuốn sách rất hay Lan thường đến tặng Nam
- Cuốn sách rất hay Lan thường đến tặng Nam ở thư viện
- Cuốn sách hay Lan thường đến tặng Nam ở thư viện
- Cuốn sách rất hay Lan thường đến tặng Nam ở thư viện

Các câu phái sinh từ câu b, c :

- Nam đọc sách ở trường
- Nam thích đọc sách ở trường
- Nam rất thích đọc sách ở trường
- Nam đọc sách ở thư viện trường
- Nam thích đọc sách ở thư viện trường
- Nam rất thích đọc sách ở thư viện trường
- Nam đọc sách ở gần trường
- Nam thích đọc sách ở gần trường
- Nam rất thích đọc sách ở gần trường

- Nam đọc sách ở gần thư viện trường
- Nam thích đọc sách ở gần thư viện trường
- Nam rất thích đọc sách ở gần thư viện trường
- Nhà Nam ở thư viện
- Nhà Nam ở gần thư viện
- Nhà Nam ở rất gần thư viện
- Nhà Nam ở gần thư viện trường
- Nhà Nam ở rất gần thư viện trường
- Nam rất thích ở thư viện
- Nam thích ở thư viện
- Nam rất thích ở thư viện trường
- Nam thích ở thư viện trường
- Nam thích sách ở thư viện
- Nam rất thích sách ở thư viện

Các câu phái sinh từ câu b, d :

- Nam mua sách mới ở thư viện
- Nam thích mua sách mới ở thư viện
- Nam rất thích mua sách mới ở thư viện
- Nam mua sách ở thư viện
- Nam thích mua sách ở thư viện
- Nam rất thích mua sách ở thư viện
- Nam mua cuốn sách mới ở thư viện
- Nam thích mua cuốn sách mới ở thư viện
- Nam rất thích mua cuốn sách mới ở thư viện
- Nam mua mấy cuốn sách ở thư viện
- Nam thích mấy mua cuốn sách ở thư viện

- Nam rất thích mua mấy cuốn sách ở thư viện
- Nam mua mấy cuốn sách mới ở thư viện
- Nam thích mua mấy cuốn sách mới ở thư viện
- Nam rất thích mua mấy cuốn sách mới ở thư viện
- Nam mua mấy cuốn sách ở thư viện
- Nam thích mua mấy cuốn sách ở thư viện
- Nam rất thích mua mấy cuốn sách ở thư viện
- Nam mới mua cuốn sách mới ở thư viện
- Nam mới mua mấy cuốn sách ở thư viện
- Nam mới mua sách ở thư viện

Các câu phái sinh từ c, d:

- Nam mua sách mới ở trường
- Nam thích mua sách mới ở trường
- Nam rất thích mua sách mới ở trường
- Nam mua sách ở trường
- Nam thích mua sách ở trường
- Nam rất thích mua sách ở trường
- Nam mua cuốn sách mới ở trường
- Nam thích mua cuốn sách mới ở trường
- Nam rất thích mua cuốn sách mới ở trường
- Nam mua mấy cuốn sách ở trường
- Nam thích mua mấy cuốn sách ở trường
- Nam rất thích mua mấy cuốn sách ở trường
- Nam mua mấy cuốn sách mới ở trường
- Nam thích mua mấy cuốn sách mới ở trường
- Nam rất thích mua mấy cuốn sách mới ở trường

- Nam mua mấy cuốn sách ở trường
- Nam thích mua mấy cuốn sách ở trường
- Nam rất thích mua mấy cuốn sách ở trường
- Nam mới mua cuốn sách mới ở trường
- Nam mới mua mấy cuốn sách ở trường
- Nam mới mua sách ở trường

Các câu phái sinh từ c, e:

- Cuốn sách Nam tặng Lan ở trường
- Cuốn sách hay Nam tặng Lan ở trường
- Cuốn sách rất hay Nam tặng Lan ở trường
- Cuốn sách rất hay Nam đến tặng Lan ở trường
- Cuốn sách hay Nam đến tặng Lan ở trường
- Cuốn sách rất hay Nam đến tặng Lan ở trường
- Lan đến trường tặng Nam cuốn sách
- Lan đến trường tặng Nam cuốn sách hay
- Lan đến trường tặng Nam cuốn sách rất hay
- Lan đến trường tặng Nam một cuốn sách
- Lan đến trường tặng Nam một cuốn sách hay
- Lan đến trường tặng Nam một cuốn sách rất hay
- Cuốn sách Lan tặng Nam ở trường
- Cuốn sách hay Lan tặng Nam ở trường
- Cuốn sách rất hay Lan tặng Nam ở trường
- Cuốn sách rất hay Lan đến tặng Nam ở trường
- Cuốn sách hay Lan đến tặng Nam ở trường
- Cuốn sách rất hay Lan đến tặng Nam ở trường
- Nam thường đến trường tặng Lan cuốn sách

- Nam thường đến trường tặng Lan cuốn sách hay
- Nam thường đến trường tặng Lan cuốn sách rất hay
- Nam thường đến trường tặng Lan một cuốn sách
- Nam thường đến trường tặng Lan một cuốn sách hay
- Nam thường đến trường tặng Lan một cuốn sách rất hay
- Cuốn sách Nam thường tặng Lan ở trường

Các câu phái sinh từ d, e:

- Nam mua tặng Lan cuốn sách mới
- Nam mua tặng Lan mấy cuốn sách mới
- Nam mua tặng Lan sách mới
- Nam mới mua tặng Lan sách mới
- Nam mới mua tặng Lan cuốn sách mới
- Nam mới mua tặng Lan mấy cuốn sách mới
- Lan mua tặng Nam cuốn sách mới
- Lan mua tặng Nam mấy cuốn sách mới
- Lan mua tặng Nam sách mới
- Lan mới mua tặng Nam sách mới
- Lan mới mua tặng Nam cuốn sách mới
- Lan mới mua tặng Nam mấy cuốn sách mới
- Nam mua tặng Lan một cuốn sách mới
- Nam mới mua tặng Lan một cuốn sách mới
- Lan mua tặng Nam một cuốn sách mới
- Lan mới mua tặng Nam một cuốn sách mới

## **2. Văn pháp DCG ( Definite Clause Grammar )**



### 2.1. Bảng quy tắc cho văn phạm DCG

Từ 5 câu trong tập dữ liệu trên có thể phân tích từ loại sau theo ngữ pháp tiếng việt:

Từ	Loại từ
Nam	Danh từ riêng
Lan	Danh từ riêng
Nhà	Danh từ chung
Trường	Danh từ chung
Thư viện	Danh từ chung
Sách	Danh từ chung
Đọc	Động từ
Mua	Động từ
Tặng	Động từ
Thích	Động từ
Máy	Tính từ
Mới	Tính từ
Hay	Tính từ
Của	Gới từ
ở	Giới từ

*Bảng 1: Từ và Loại từ*

Bảng quy tắc cho DCG:

Ký hiệu	Ý nghĩa sử dụng
S	Câu
NP	Dạng ngữ
VP	Ngữ động từ

ADVP	Trạng ngữ
ADJP	Tính ngữ
PP	Giới ngữ
N	Dạng từ chung
PPN (Proper N)	Dạng từ riêng
V	Động từ
ADJ	Tính từ
P	Giới từ
ADV	Trạng từ

*Bảng 2: Quy tắc DCG*

## 2.2. Văn phạm DCG cho câu gốc và câu phái sinh.

Văn phạm cho các câu gốc và phái sinh của câu:

**Câu a:** Nam thường đến thư viện

Câu phái sinh :

- Nam đến thư viện.
- Nam đến thư viện thường.
- Thư viện Nam đến.
- Thư viện Nam thường đến.

Văn phạm DCG cho câu gốc và câu phái sinh :

Prolog
:- discontinuous np/3. % cau goc $s(s(NP, VP)) \rightarrow np(NP), vp(VP).$ $np(np(PPN)) \rightarrow ppn(PPN).$ $vp(vp(ADVP, V, NP)) \rightarrow adv(ADVP), v(V), n(NP).$

```
% them cac cau phai sinh
vp(vp(V, NP)) --> v(V), np(NP).
vp(vp(V, NP, ADVP)) --> v(V), n(NP),adv(ADVP).
vp(vp(NP,ADVP,V)) --> np(NP), adv(ADVP), v(V).
vp(vp(NP,V)) --> np(NP), v(V).

% them du lieu cho cau
ppn(ppn(nam)) --> [nam].
adv(adv(thuong)) --> [thuong].
v(v(den)) --> [den].
np(np(N)) --> n(N).
n(n(thu_vien)) --> [thu], [vien].
```

Thực hiện truy vấn trên prolog:

```
?- s(Tree,[nam,thuong,den,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(adv(thuong), v(den), n(thu_vien))) .

?- s(Tree,[nam,den,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien)))) .

?- s(Tree,[nam,den,thu,vien,thuong],[]).
Tree = s(np(ppn(nam)), vp(v(den), n(thu_vien), adv(thuong)))

?- s(Tree,[thu,vien,nam,den],[]).
Tree = s(np(n(thu_vien)), vp(np(ppn(nam)), v(den))) .
```

?- s(Tree,[thu,vien,nam,thuong,den],[]).

Tree = s(np(n(thu\_vien)), vp(np(ppn(nam)), adv(thuong), v(den)))

**Câu b:** Nam rất thích đọc sách ở thư viện.

Các câu phái sinh :

- Nam thích đọc sách ở thư viện.
- Nam đọc sách ở thư viện.
- Nam rất thích sách ở thư viện.
- Nam thích sách ở thư viện
- Nam rất thích ở thư viện.
- Nam thích ở thư viện.
- Nam thích sách.
- Nam rất thích sách.
- Nam thích đọc sách.
- Nam rất thích đọc sách.

Văn phạm DCG cho câu gốc và câu phái sinh :

Prolog

:- discontinuous np/3.

:- discontinuous v/3.

:- discontinuous n/3.

:- discontinuous vp/3.

:- discontinuous s/3.

% cau goc

```
s(s(NP, VP, PP)) --> np(NP), vp(VP), pp(PP).  
np(np(PPN)) --> ppn(PPN).  
vp(vp(ADVP, V, VP)) --> advp(ADVP), v(V), vp(VP).  
pp(pp(P, NP)) --> p(P), np(NP).  
advp(advp(ADV)) --> adv(ADV).  
np(np(N)) --> n(N).
```

% them cau phai sinh

```
s(s(NP,VP)) --> np(NP), vp(VP).  
vp(vp(ADVP,V)) --> advp(ADVP), v(V).  
vp(vp(V)) --> v(V).  
vp(vp(V,V1,NP)) --> v(V),v(V1),np(NP).  
vp(vp(V, NP)) --> v(V), np(NP).  
vp(vp(NP)) --> np(NP).
```

% them du lieu

```
ppn(ppn(nam)) --> [nam].  
adv(adv(rat)) --> [rat].  
v(v(thich)) --> [thich].  
v(v(doc)) --> [doc].  
n(n(sach)) --> [sach].  
p(p(o)) --> [o].  
n(n(thu_vien)) --> [thu], [vien].
```

Thực hiện truy vấn trên prolog:

Prolog

?- s(Tree,[nam,rat,thich,doc,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(v(doc), np(sach))), pp(p(o), np(n(thu\_vien)))) .

?- s(Tree,[nam,thich,doc,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(thich), v(doc), np(sach)), pp(p(o), np(n(thu\_vien))))

?- s(Tree,[nam,doc,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(doc), np(sach)), pp(p(o), np(n(thu\_vien)))) .

?- s(Tree,[nam,rat,thich,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(np(sach))), pp(p(o), np(n(thu\_vien)))) .

?- s(Tree,[nam,rat,thich,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich)), pp(p(o), np(n(thu\_vien)))) .

?- s(Tree,[nam,thich,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(thich)), pp(p(o), np(n(thu\_vien))))

?- s(Tree,[nam,thich,sach],[]).

Tree = s(np(ppn(nam)), vp(v(thich), np(n(sach))))

?- s(Tree,[nam,rat,thich,sach],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(np(n(sach)))))

?- s(Tree,[nam,thich,doc,sach],[]).

Tree = s(np(ppn(nam)), vp(v(thich), v(doc), np(n(sach))))

?- s(Tree,[nam,rat,thich,doc,sach],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(v(doc), np(n(sach)))))

**Câu c:** Nhà của Nam ở gần trường.

Các câu phái sinh có thể là:

- Nhà Nam ở gần trường.
- Nhà Nam gần trường.
- Nhà của Nam gần trường.
- Nhà của Nam ở trường
- Nhà Nam ở trường
- Nhà ở gần trường Nam
- Nhà gần trường Nam

Văn phạm DCG cho các câu trên :

Prolog

:- discontiguous np/3.

:- discontiguous n/3.

:- discontiguous vp/3.

:- discontiguous pp/3.

:- discontiguous p/3.

% cau goc

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

np(np(N,PP)) --> n(N), pp(PP).

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

```

np(np(PPN)) --> ppn(PPN).
vp(vp(PP, NP)) --> pp(PP), np(NP).
pp(pp(P,P2)) --> p(P), p(P2).
np(np(N))--> n(N).

```

% them cau phai sinh

```

np(np(N,PPN)) --> n(N), ppn(PPN).
pp(pp(P)) --> p(P).

```

%them du lieu

```

n(n(nha)) --> [nha].
p(p(cua)) --> [cua].
ppn(ppn(nam)) --> [nam].
p(p(o)) --> [o].
p(p(gan)) --> [gan].
n(n(truong)) --> [truong].

```

Thực thiện truy vấn trên prolog:

Prolog

```

?- s(Tree,[nha,cua,nam,o,gan,truong],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(pp(p(o), p(gan)),
np(n(truong)))) .

?- s(Tree,[nha,nam,o,gan,truong],[]).
Tree = s(np(n(nha), ppn(nam)), vp(pp(p(o), p(gan)), np(n(truong)))) .

```



?- s(Tree,[nha,nam,gan,truong],[]).

Tree = s(np(n(nha), ppn(nam)), vp(pp(p(gan)), np(n(truong)))) .

?- s(Tree,[nha,cua,nam,gan,truong],[]).

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(pp(p(gan)), np(n(truong)))) .

?- s(Tree,[nha,cua,nam,o,truong],[]).

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(pp(p(o)), np(n(truong)))) .

?- s(Tree,[nha,nam,o,truong],[]).

Tree = s(np(n(nha), ppn(nam)), vp(pp(p(o)), np(n(truong)))) .

?- s(Tree,[nha,o,gan,truong,nam],[]).

Tree = s(np(n(nha), pp(p(o))), vp(pp(p(gan), np(n(truong))), np(ppn(nam))))) .

?- s(Tree,[nha,gan,truong,nam],[]).

Tree = s(np(n(nha)), vp(pp(p(gan), np(n(truong))), np(ppn(nam)))))

**Câu d:** Nam mới mua mấy cuốn sách mới.

Các câu phái sinh có thể là:

- Nam mua mấy cuốn sách mới.
- Nam mới mua mấy cuốn sách.
- Nam mua cuốn sách mới.
- Nam mới mua cuốn sách.
- Nam mua cuốn sách.

- Nam mua sách.
- Nam mua sách mới.
- Nam mới mua sách.
- Nam mới mua sách mới.

Văn phạm DCG các câu trên:

Prolog
<pre>:- discontiguous np/3. :- discontiguous adjp/3. :- discontiguous adj/3. % cau goc s(s(NP, VP)) --&gt; np(NP), vp(VP). np(np(PPN)) --&gt; ppn(PPN). vp(vp(ADVP,V,NP)) --&gt; advp(ADVP),v(V),np(NP). np(np(ADJP,N,ADJP2))--&gt; adjp(ADJP),n(N),adjp(ADJP2). advp(advp(ADV)) --&gt; adv(ADV). adjp(adjp(ADJ)) --&gt; adj(ADJ). np(np(N)) --&gt; n(N).  % them cau phai sinh vp(vp(V,NP)) --&gt; v(V),np(NP). np(np(ADJP,N)) --&gt; adjp(ADJP),n(N). np(np(N,ADJP)) --&gt; n(N), adjp(ADJP). n(n(sach)) --&gt; [sach].  % them du lieu ppn(ppn(nam)) --&gt; [nam].</pre>

adv(adv(moi)) --> [moi].  
 v(v(mua)) --> [mua].  
 adj(adj(may)) --> [may].  
 n(n(cuon\_sach))--> [cuon], [sach].  
 adj(adj(moi)) --> [moi].

Truy vấn trong prolog:

Prolog
<pre> ?- s(Tree,[nam,moi,mua,may,cuon,sach,moi],[]). Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may))), n(cuon_sach), adjp(adj(moi))))) .  ?- s(Tree,[nam,mua,may,cuon,sach,moi],[]). Tree = s(np(ppn(nam)), vp(v(mua), np(adjp(adj(may))), n(cuon_sach), adjp(adj(moi))))) .  ?- s(Tree,[nam,moi,mua,may,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may))), n(cuon_sach))))) .  ?- s(Tree,[nam,mua,may,cuon,sach,moi],[]). Tree = s(np(ppn(nam)), vp(v(mua), np(adjp(adj(may))), n(cuon_sach), adjp(adj(moi))))) . </pre>

?- s(Tree,[nam,moi,mua,cuon,sach],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(cuon\_sach)))) .

?- s(Tree,[nam,mua,cuon,sach],[]).

Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon\_sach))))

?- s(Tree,[nam,mua,cuon,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon\_sach), adjp(adj(moi)))))

?- s(Tree,[nam,mua,sach],[]).

Tree = s(np(ppn(nam)), vp(v(mua), np(n(sach)))) .

?- s(Tree,[nam,mua,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(v(mua), np(n(sach), adjp(adj(moi))))) .

?- s(Tree,[nam,moi,mua,sach],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach)))) .

?- s(Tree,[nam,moi,mua,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach), adjp(adj(moi)))))

**Câu e:** Nam tặng Lan một cuốn sách rất hay.

Các câu phái sinh có thể là:

- Nam tặng Lan cuốn sách.
- Nam tặng Lan một cuốn sách.
- Nam tặng Lan cuốn sách rất hay.
- Lan tặng Nam cuốn sách.

- Lan tặng Nam một cuốn sách.
- Lan tặng Nam cuốn sách rất hay.
- Nam tặng Lan sách
- Nam tặng Lan sách rất hay.
- Lan tặng Nam sách.
- Lan tặng Nam sách rất hay.

Văn phạm DCG các câu trên:

Prolog
<pre>:- discontinuous adjp/3. :- discontinuous adj/3. :- discontinuous np/3. :- discontinuous ppn/3. % cau goc s(s(NP, VP)) --&gt; np(NP), vp(VP). np(np(PPN)) --&gt; ppn(PPN). vp(vp(V, NP, NP2)) --&gt; v(V), np(NP), np(NP2). np(np(ADJP, N, ADJP2)) --&gt; adjp(ADJP), n(N), adjp(ADJP2). np(np(N)) --&gt; n(N). adjp(adjp(ADVP, ADJ)) --&gt; advp(ADVP), adj(ADJ).  % them cau phai sinh np(np(N, ADJP)) --&gt; n(N), adjp(ADJP). np(np(ADJP, N)) --&gt; adjp(ADJP), n(N). adjp(adjp(ADJ)) --&gt; adj(ADJ). advp(advp(ADV)) --&gt; adv(ADV).</pre>

```
% them du lieu
ppn(ppn(nam)) --> [nam].
ppn(ppn(lan)) --> [lan].
adj(adj(mot)) --> [mot].
v(v(tang)) --> [tang].
n(n(sach)) --> [sach].
n(n(cuon_sach)) --> [cuon], [sach].
adv(adv(rat)) --> [rat].
adj(adj(hay)) --> [hay].
```

Truy vấn trong prolog:

Prolog
<pre>?- s(Tree,[nam,tang,lan,mot,cuon,sach,rat,hay],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach), adjp(advp(adv(rat))), adj(hay)))) .</pre>
<pre>?- s(Tree,[nam,tang,lan,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach)))) .</pre>
<pre>?- s(Tree,[nam,tang,lan,mot,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach)))) .</pre>
<pre>?- s(Tree,[nam,tang,lan,cuon,sach,rat,hay],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach), adjp(advp(adv(rat))), adj(hay))))Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)),</pre>

np(n(cuon\_sach))))

?- s(Tree,[lan,tang,nam,cuon,sach],[]).

Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(n(cuon\_sach))))

?- s(Tree,[lan,tang,nam,mot,cuon,sach],[]).

Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(adjp(adj(mot)), n(cuon\_sach))))

?- s(Tree,[lan,tang,nam,cuon,sach,rat,hay],[]).

Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(n(cuon\_sach), adjp(advp(adv(rat)), adj(hay)))))

?- s(Tree,[nam,tang,lan,sach],[]).

Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan))), np(n(sach))))

?- s(Tree,[nam,tang,lan,sach,rat,hay],[]).

Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan))), np(n(sach), adjp(advp(adv(rat)), adj(hay)))))

?- s(Tree,[lan,tang,nam,sach],[]).

Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(n(sach))))

?- s(Tree,[lan,tang,nam,sach,rat,hay],[]).

Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(n(sach), adjp(advp(adv(rat)), adj(hay)))))

### 2.3. Văn phạm DCG chung cho các câu a, b, c, d, e và phái sinh

Văn phạm DCG chung cho các câu :

Prolog
<pre>:- discontinuous np/3. :- discontinuous vp/3. :- discontinuous advp/3. :- discontinuous adjp/3. :- discontinuous ppn/3. :- discontinuous n/3. :- discontinuous v/3. :- discontinuous adj/3. :- discontinuous adv/3. :- discontinuous pp/3. :- discontinuous p/3.  % cấu trúc câu s(s(NP, VP)) --&gt; np(NP), vp(VP). s(s(NP,VP,PP)) --&gt; np(NP), vp(VP),pp(PP).  % cụm danh từ np(np(PPN)) --&gt; ppn(PPN). np(np(N)) --&gt; n(N). np(np(N,NP)) --&gt; n(N), np(NP). np(np(N,PP)) --&gt; n(N), pp(PP). np(np(PPN,PP)) --&gt; ppn(PPN), pp(PP).</pre>



np(np(ADJP,N,ADJP2))--> adjp(ADJP),n(N),adjp(ADJP2).  
 np(np(ADJP,N)) --> adjp(ADJP),n(N).  
 np(np(N,ADVP)) --> n(N), advp(ADVP).  
 np(np(ADJP,N,ADVP)) --> adjp(ADJP), n(N), advp(ADVP).  
 np(np(ADJP,PPN)) --> adjp(ADJP), ppn(PPN).

% cụm động từ

vp(vp(V)) --> v(V).  
 vp(vp(V, NP)) --> v(V), np(NP).  
 vp(vp(PP,NP)) --> pp(PP),np(NP).  
 vp(vp(ADVP,V,NP)) --> advp(ADVP), v(V), np(NP).  
 vp(vp(ADVP,V,VP)) --> advp(ADVP), v(V), vp(VP).  
 vp(vp(ADVP,V)) --> advp(ADVP), v(V).  
 vp(vp(V,ADVP)) --> v(V),advp(ADVP).  
 vp(vp(V,NP,NP2)) --> v(V), np(NP), np(NP2).  
 vp(vp(V,NP,NP2,ADJP)) --> v(V), np(NP), np(NP2),adjp(ADJP).  
 vp(vp(ADVP,VP)) --> advp(ADVP), vp(VP).  
 vp(vp(ADVP)) --> advp(ADVP).  
 vp(vp(ADVP,V,NP,VP)) --> advp(ADVP), v(V), np(NP),vp(VP).  
 vp(vp(V,VP)) --> v(V), vp(VP).

% cụm giới từ

pp(pp(P)) --> p(P).  
 pp(pp(P,P2)) --> p(P), p(P2).  
 pp(pp(P,NP)) --> p(P), np(NP).  
 pp(pp(NP)) --> np(NP).

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

% cụm trạng từ

advp(advp(ADV)) --> adv(ADV).

advp(advp(ADV, NP)) --> adv(ADV), np(NP).

% cụm tính từ

adjp(adjp(ADJ)) --> adj(ADJ).

adjp(adjp(ADVP, ADJ)) --> advp(ADVP), adj(ADJ).

% them dữ liệu

ppn(ppn(nam)) --> [nam].

ppn(ppn(lan)) --> [lan].

n(n(thu\_vien)) --> [thu], [vien].

n(n(cuon\_sach)) --> [cuon], [sach].

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

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

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

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

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

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

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

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

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

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

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

```
adj(adj(mot)) --> [mot].
adj(adj(hay)) --> [hay].
adv(adv(thuong)) --> [thuong].
adv(adv(rat)) --> [rat].
p(p(gan)) --> [gan].
adv(adv(moi)) --> [moi].
p(p(o)) --> [o].
```

## V. Thử nghiệm

### 1. Cấu trúc thử nghiệm

#### Dữ liệu:

- Các câu gốc và phái sinh trong tập dữ liệu
- Các câu gốc và phái sinh kết hợp với nhau.

#### Kết quả chạy cho trạng thái :

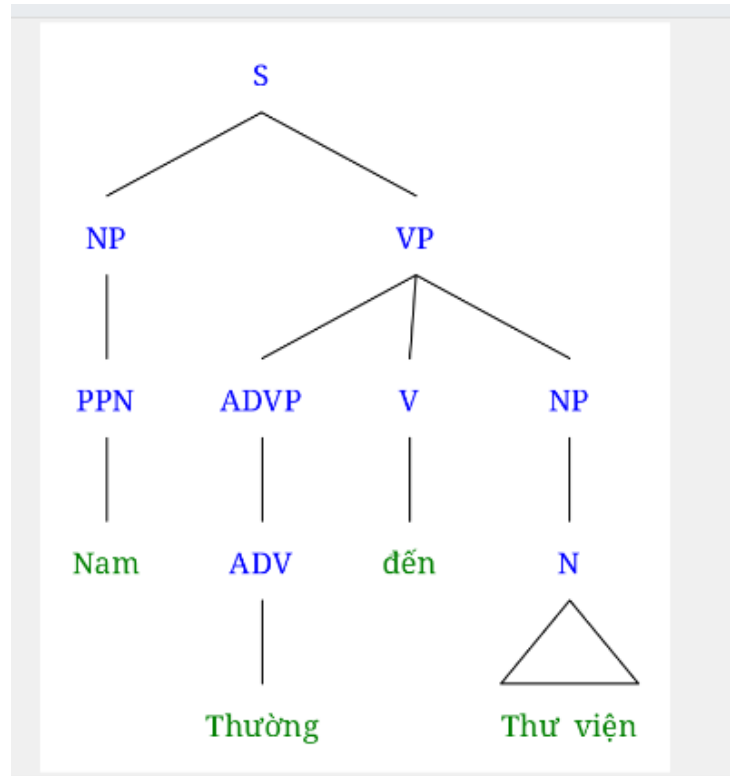
- Nếu câu đúng (**True**) in ra cây cấu trúc của câu.
- Nếu câu sai trả về kết quả **False**.

### 2. Thử nghiệm với Prolog

#### 2.1. Các câu gốc

**Câu a:** Nam thường đến thư viện.

Cấu trúc cây:



Hình 1.2. cây cú pháp cho câu a

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,den,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), np(n(thu_vien))))

### Các câu phái sinh:

- Nam đến thư viện.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien))))

- Nam đến thư viện thường.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,thuong],[]). Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien), advp(adv(thuong)))))

- Thư viện Nam đến.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[thu,vien,nam,den],[]). Tree = s(np(n(thu_vien), np(ppn(nam))), vp(v(den)))

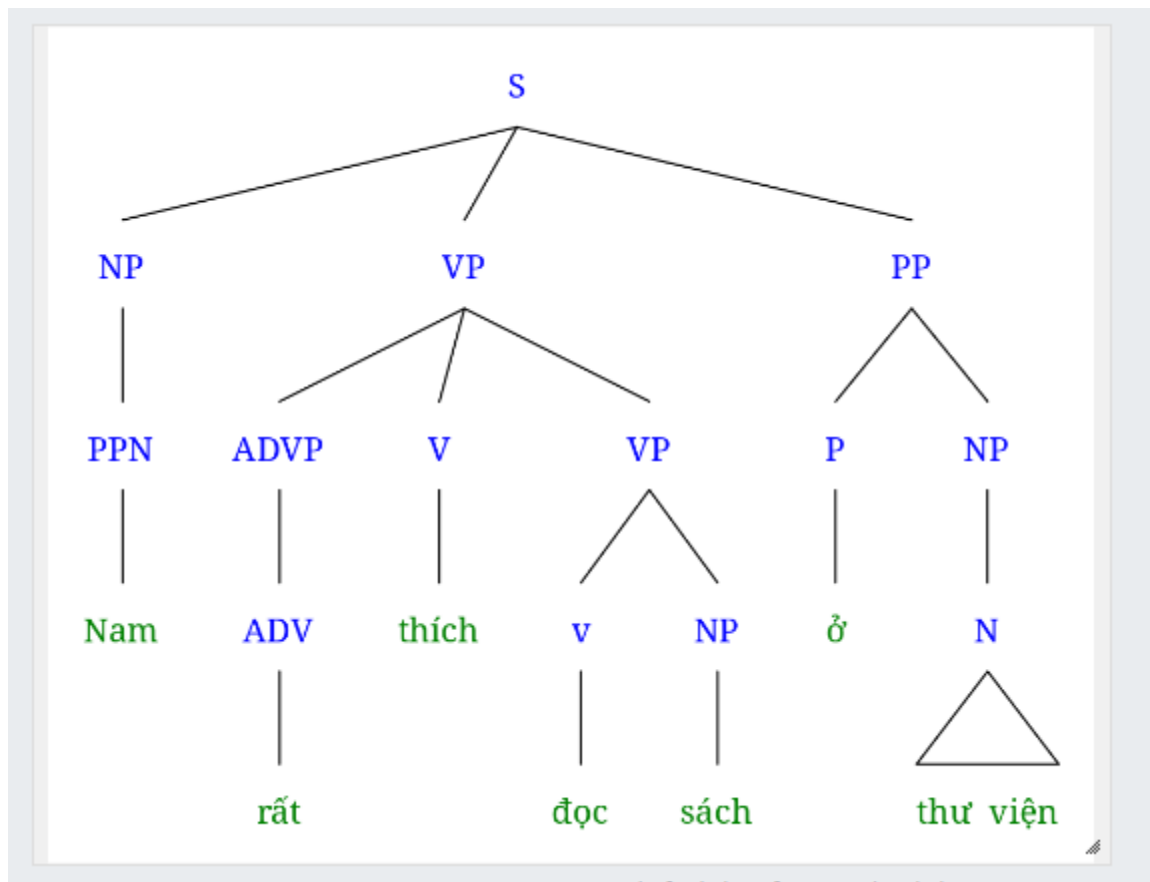
- Thư viện Nam thường đến.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[thu,vien,nam,thuong,den],[]). Tree = s(np(n(thu_vien), np(ppn(nam))), vp(advp(adv(thuong)), v(den)))

**Câu b:** Nam rất thích đọc sách ở thư viện.

Cấu trúc cây:



Hình 1.3. cây cú pháp cho câu b

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thich,doc,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(thich), vp(v(doc), np(n(sach)))), pp(p(o), np(n(thu\_vien)))))

**Các câu phái sinh:**

- Nam thích đọc sách ở thư viện.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thich,doc,sach,o,thu,vien],[]). Tree = s(np(ppn(nam)), vp(v(thich), vp(v(doc), np(n(sach)))), pp(p(o), np(n(thu_vien)))))

- Nam đọc sách ở thư viện.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,doc,sach,o,thu,vien],[]). Tree = s(np(ppn(nam)), vp(v(doc), np(n(sach))), pp(p(o), np(n(thu_vien)))))

- Nam rất thích sách ở thư viện.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,rat,thich,sach,o,thu,vien],[]). Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), np(n(sach)), vp(v(o), np(n(thu_vien)))))

- Nam rất thích ở thư viện.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,rat,thich,o,thu,vien],[]). Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(v(o), np(n(thu_vien)))))

- Nam thích ở thư viện.

Thực hiện truy vấn trên công cụ:

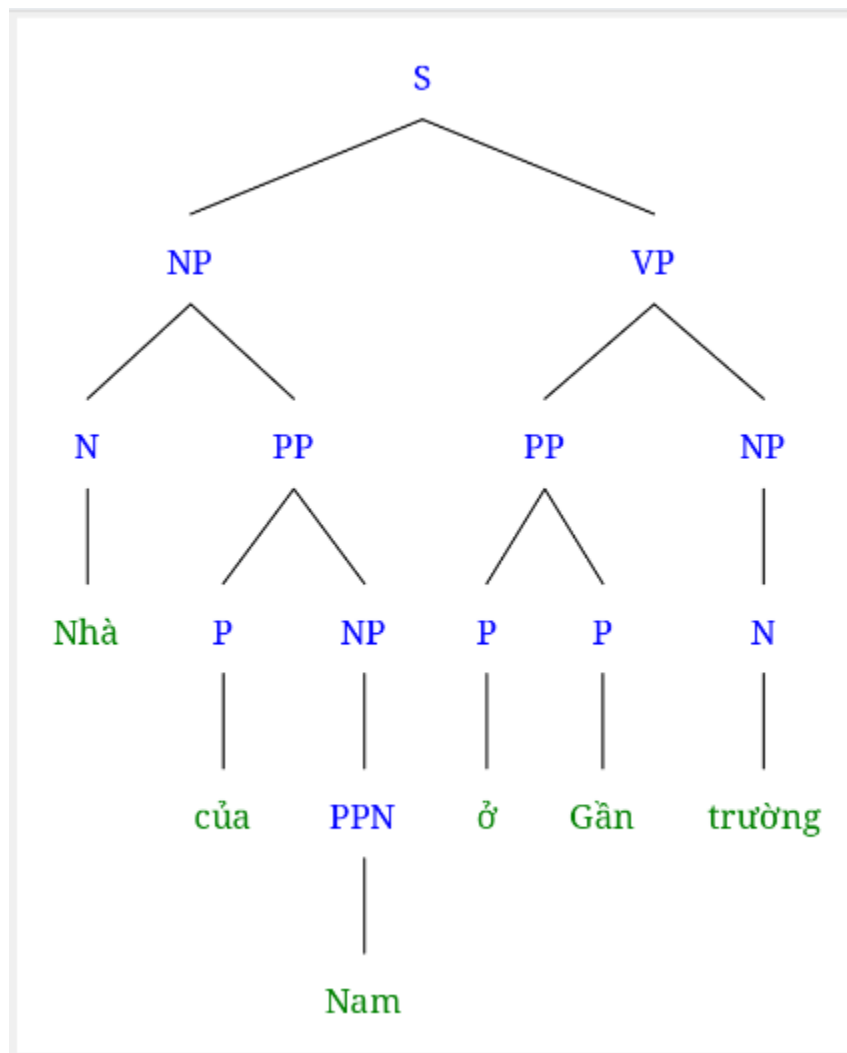
Prolog

?- s(Tree,[nam,thich,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(thich), vp(v(o), np(n(thu\_vien)))))

**Câu c:** Nhà của Nam ở gần trường.

Cấu trúc cây:



Hình 1.4. cây cú pháp cho câu c

Thực hiện truy vấn trên công cụ:

Prolog



?- s(Tree,[nha,cua,nam,o,gan,truong],[]).

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(pp(p(o), p(gan)), np(n(truong))))

### Các câu phái sinh:

- Nhà Nam ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nha,nam,o,gan,truong],[]).

Tree = s(np(n(nha), ppn(nam)), vp(pp(p(o), p(gan)), np(n(truong))))

- Nhà Nam gần trường.

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nha,nam,gan,truong],[]).

Tree = s(np(n(nha), ppn(nam)), vp(pp(p(gan)), np(n(truong))))

- Nhà của Nam gần trường.

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nha,cua,nam,gan,truong],[]).

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(pp(p(gan)), np(n(truong))))

- Nhà của Nam ở trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,o,truong],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(pp(p(o)), np(n(truong))))

- Nhà Nam ở trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,o,truong],[]).
Tree = s(np(n(nha), ppn(nam)), vp(pp(p(o)), np(n(truong))))

- Nhà ở gần trường Nam

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,o,gan,truong,nam],[]).
Tree = s(np(n(nha), pp(p(o))), vp(pp(p(gan), np(n(truong))), np(ppn(nam))))

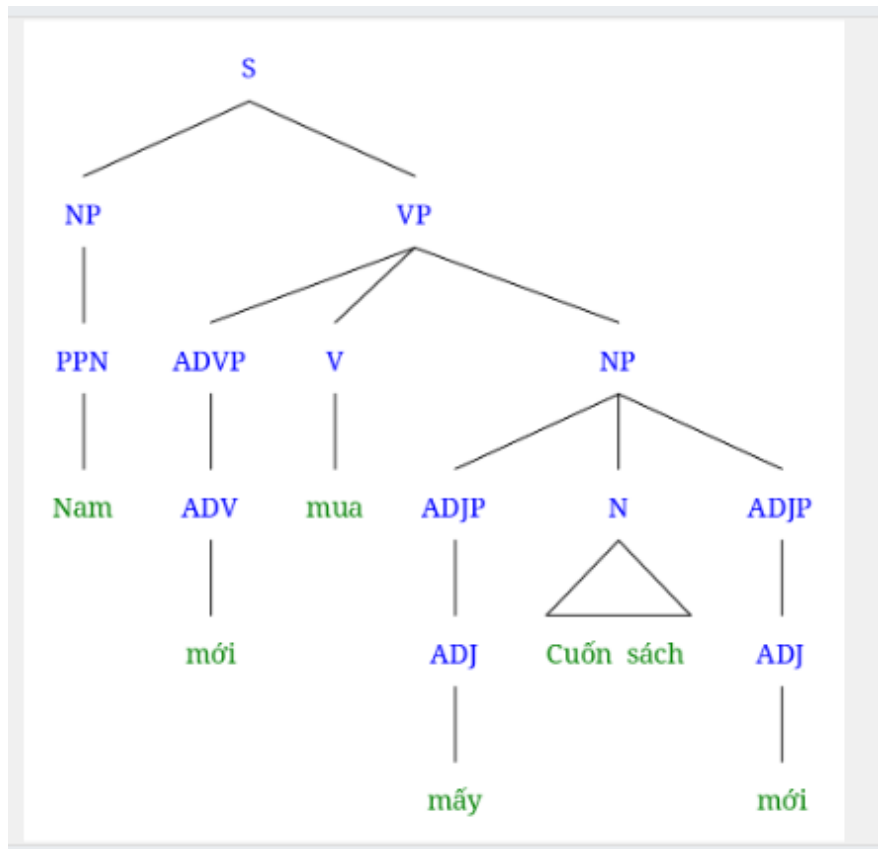
- Nhà gần trường Nam

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,gan,truong,nam],[]).
Tree = s(np(n(nha)), vp(pp(p(gan), np(n(truong))), np(ppn(nam))))

**Câu d:** Nam mới mua mấy cuốn sách mới.

Cấu trúc cây:



Hình 1.5. cây cú pháp cho câu d

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,moi,mua,may,cuon,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may)),  
n(cuon\_sach), adjp(adj(moi)))))

**Các câu phái sinh:**

- Nam mua mấy cuốn sách mới.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,may,cuon,sach,moi],[]). Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi)))))

- Nam mới mua mấy cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,may,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may)), n(cuon_sach))))

- Nam mua cuốn sách mới.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,cuon,sach,moi],[]). Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon_sach), advp(adv(moi)))))

- Nam mới mua cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(cuon_sach))))

- Nam mua cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon_sach))))

- Nam mua sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,sach],[]). Tree = s(np(ppn(nam)), vp(v(mua), np(n(sach))))

- Nam mua sách mới.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,sach,moi],[]). Tree = s(np(ppn(nam)), vp(v(mua), np(n(sach), advp(adv(moi)))))

- Nam mới mua sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,sach],[]). Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach))))

- Nam mới mua sách mới.

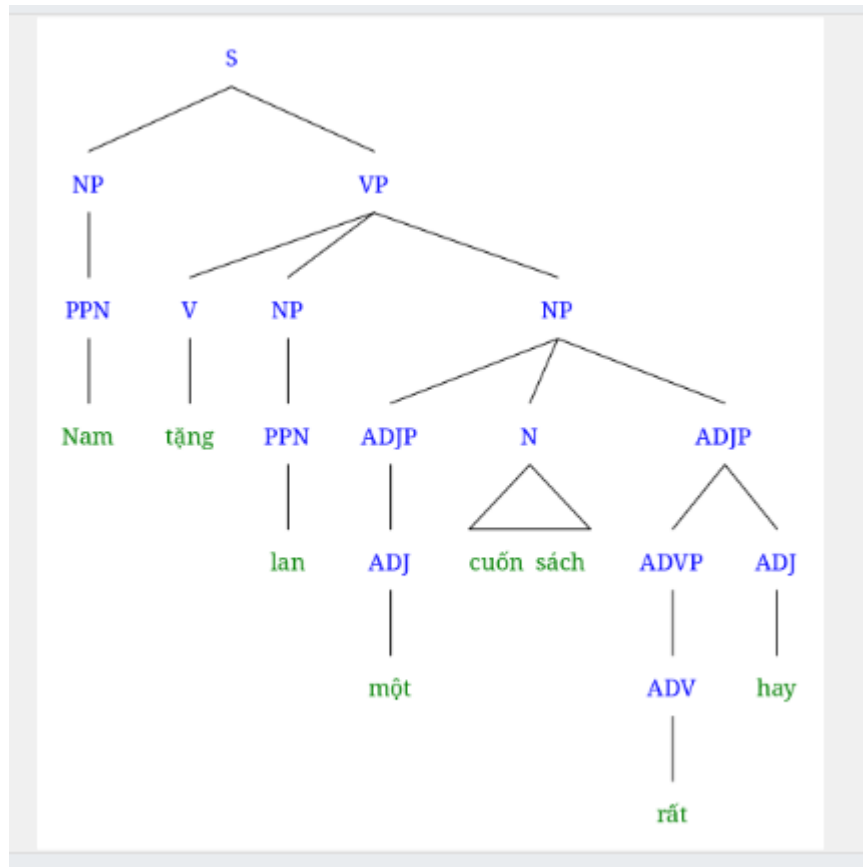
Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach), advp(adv(moi)))))

**Câu e:** Nam tặng Lan một cuốn sách rất hay.

Cấu trúc cây:



Hình 1.6. cây cú pháp cho câu e

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,tang,lan,mot,cuon,sach,rat,hay],[]).

Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon\_sach), adjp(advp(adv(rat)), adj(hay)))))

**Các câu phái sinh:**

- Nam tặng Lan cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,tang,lan,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach))))

- Nam tặng Lan một cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,tang,lan,mot,cuon,sach],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach))))

- Nam tặng Lan cuốn sách rất hay.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,tang,lan,cuon,sach,rat,hay],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach)), adjp(advp(adv(rat)), adj(hay))))

- Lan tặng Nam cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[lan,tang,lan,cuon,sach],[]). Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach))))

- Lan tặng Nam một cuốn sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[lan,tang,lan,mot,cuon,sach],[]). Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach))))

- Lan tặng Nam cuốn sách rất hay.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[lan,tang,lan,cuon,sach,rat,hay],[]). Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach), adjp(advp(adv(rat)), adj(hay))))

- Nam tặng Lan sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,tang,lan,sach],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(n(sach))))

-

- Nam tặng Lan sách rất hay.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,tang,lan,sach,rat,hay],[]). Tree = s(np(ppn(nam)), vp(v(tang), np(ppn(lan)), np(n(sach), adjp(advp(adv(rat)), adj(hay))))



- Lan tặng Nam sách.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[lan,tang,nam,sach],[]).
Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(n(sach))))

- Lan tặng Nam sách rat hay.

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[lan,tang,nam,sach,rat,hay],[]).
Tree = s(np(ppn(lan)), vp(v(tang), np(ppn(nam))), np(n(sach)), adjp(advp(adv(rat)), adj(hay))))

## 2.2. Các câu phái sinh từ a, b

- Nam đến đọc sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,doc,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(den), vp(v(doc), np(n(sach)))), pp(p(o), np(n(thu_vien)))))

- Nam thích đến đọc sách ở thư thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thich,den,doc,sach,o,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(v(thich), vp(v(den), vp(v(doc), np(n(sach))))), pp(p(o), np(n(thu_vien))))</pre>

- Nam rất thích đến đọc sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,rat,thich,den,doc,sach,o,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(v(den), vp(v(doc), np(n(sach))))), pp(p(o), np(n(thu_vien))))</pre>

- Nam thường thích đến thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), np(n(thu_vien))))</pre>

- Nam thường rất thích đến thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,rat,thich,den,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(advp(adv(rat)), v(thich), vp(v(den), np(n(thu_vien))))))</pre>

- Nam thường đến đọc sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,den,doc,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(doc), np(n(sach)))), pp(p(o), np(n(thu_vien)))))

- Nam thường thích đọc sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,thich,doc,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(thich), vp(v(doc), np(n(sach)))), pp(p(o), np(n(thu_vien)))))

- Nam thường rất thích đọc sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,rat,thich,doc,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(advp(adv(rat))), v(thich), vp(v(doc), np(n(sach)))), pp(p(o), np(n(thu_vien)))))

- Nam thường rất thích sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,rat,thich,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(advp(adv(rat)), v(thich), np(n(sach)), vp(p(o), np(n(thu\_vien))))))

- Nam thường thích sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thuong,thich,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(thich), np(n(sach)), vp(p(o), np(n(thu\_vien))))))

- Nam thích đến ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thich,den,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(thich), vp(v(den), vp(p(o), np(n(thu\_vien))))))

- Nam rất thích đến ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,rat,thich,den,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(rat)), v(thich), vp(v(den), vp(p(o), np(n(thu\_vien))))))

- Nam thường thích đến ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,thich,den,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(thich), vp(v(den), vp(p(o), np(n(thu_vien))))))

- Nam thường rất thích đến ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,rat,thich,den,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(advp(adv(rat)), v(thich), vp(v(den), vp(p(o), np(n(thu_vien))))))

- Nam thường thích sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,thich,sach],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(thich), np(n(sach))))

- Nam thường rất thích sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,rat,thich,sach],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(advp(adv(rat)), v(thich), np(n(sach))))

- Nam thường đọc sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,doc,sach],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(doc), np(n(sach))))

- Nam thường thích đọc sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,thich,doc,sach],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(thich), vp(v(doc), np(n(sach)))))

- Nam thường rất thích đọc sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,rat,thich,doc,sach],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(advp(adv(rat)), v(thich), vp(v(doc), np(n(sach)))))

### 2.3. Các câu phái sinh từ câu a, c

- Nam đến thư viện gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,gan,truong],[]).

Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien), advp(adv(gan), np(n(truong))))))
------------------------------------------------------------------------------------

- Nam đến thư viện trường

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,thu,vien,truong],[]).
------------------------------------------

Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien), np(n(truong))))))
---------------------------------------------------------------------

- Nam đến nhà thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,nha,thu,vien],[]).
---------------------------------------

Tree = s(np(ppn(nam)), vp(v(den), np(n(nha), np(n(thu_vien))))))
------------------------------------------------------------------

- Nam đến nhà trường

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,nha,truong],[]).
-------------------------------------

Tree = s(np(ppn(nam)), vp(v(den), np(n(nha), np(n(truong))))))
----------------------------------------------------------------

- Nam đến gần trường

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,gan,truong],[]).

Tree = s(np(ppn(nam)), vp(v(den), vp(pp(p(gan)), np(n(truong)))))

- Nam đến ở gần trường.

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,den,o,gan,truong],[]).

Tree = s(np(ppn(nam)), vp(v(den), vp(pp(p(o), p(gan)), np(n(truong)))))

- Nam đến ở trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,den,o,truong],[]).

Tree = s(np(ppn(nam)), vp(v(den), vp(pp(p(o)), np(n(truong)))))

- Nam thường đến thư viện trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thuong,den,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), np(n(thu\_vien)))))

- Nam thường đến thư viện ở trường

Thực hiện truy vấn trên công cụ:

Prolog



?- s(Tree,[nam,den,thuong,den,thu,vien,o,truong],[]).

Tree = s(np(ppn(nam)), vp(v(den), vp(advp(adv(thuong)), v(den), np(n(thu\_vien), pp(p(o), np(n(truong)))))))

- Nam thường đến gần trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thuong,den,gan,truong],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(pp(p(gan)), np(n(truong)))))

- Nam thường đến ở trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thuong,den,o,truong],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(pp(p(o)), np(n(truong)))))

- Nam thường đến trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,thuong,den,truong],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), np(n(truong)))))

- Nam thường ở trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,o,truong],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(pp(p(o)), np(n(truong)))))

- Nam thường ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,o,gan,truong],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(pp(p(o), p(gan)), np(n(truong)))))

- Nam thường đến ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,den,o,gan,truong],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(truong)))))

- Nam thường đến nhà thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,nha,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(den), np(n(nha), np(n(thu_vien)))))

- Nam thường đến nhà trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,den,nha,truong],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), np(n(nha), np(n(truong)))))

- Nam thường đến nhà

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,den,nha],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), np(n(nha))))

- Nam thường ở nhà

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,o,nha],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(pp(p(o))), np(n(nha)))))

- Nam thường ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(pp(p(o))), np(n(thu_vien)))))

- Nam thường ở nhà thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,o,nha,thu,vien],[]).  Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(pp(p(o)), np(n(nha), np(n(thu_vien))))))

- Nam thường ở nhà trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,o,nha,truong],[]).  Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(pp(p(o)), np(n(nha), np(n(truong))))))

- Nam ở nhà

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,o,nha],[]).  Tree = s(np(ppn(nam)), vp(pp(p(o)), np(n(nha))))

- Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,o,thu,vien],[]).  Tree = s(np(ppn(nam)), vp(pp(p(o)), np(n(thu_vien))))

- Nam ở trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,o,truong],[]).
Tree = s(np(ppn(nam)), vp(pp(p(o)), np(n(truong))))

- Nam ở nhà thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,nha,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(pp(np(n(nha))), np(n(thu_vien))))

- Nam ở nhà trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,nha,truong],[]).
Tree = s(np(ppn(nam)), vp(pp(np(n(nha))), np(n(truong))))

- Nam gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,gan,truong],[]).
Tree = s(np(ppn(nam)), vp(pp(p(gan)), np(n(truong))))

- Nam gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,gan,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(pp(p(gan)), np(n(thu_vien))))

- Nam gần nhà

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,gan,nha],[]).
Tree = s(np(ppn(nam)), vp(pp(p(gan)), np(n(nha))))

- Nhà Nam ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,o,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam))), vp(pp(p(o), p(gan)), np(n(thu_vien))))

- Nhà Nam gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam))), vp(pp(p(gan)), np(n(thu_vien))))

- Nhà Nam thường ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,thuong,o,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam))), vp(advp(adv(thuong)), vp(pp(p(o), p(gan)), np(n(thu_vien)))))

- Nhà Nam thường gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,thuong,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam))), vp(advp(adv(thuong)), vp(pp(p(gan)), np(n(thu_vien)))))

- Nhà Nam đến thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,den,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam))), vp(v(den), np(n(thu_vien)))))

- Nhà Nam đến trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,den,truong],[]).
Tree = s(np(n(nha), np(ppn(nam))), vp(v(den), np(n(truong)))))

- Nhà Nam đến gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,den,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(v(den), vp(pp(p(gan)), np(n(thu_vien))))

- Nhà Nam đến gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,den,gan,truong],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(v(den), vp(pp(p(gan)), np(n(truong))))

- Nhà Nam đến ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,den,o,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(v(den), vp(pp(p(o), p(gan)), np(n(thu_vien))))

- Nhà Nam đến ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog
--------



?- s(Tree,[nha,nam,den,o,gan,truong],[]).

Tree = s(np(n(nha), np(ppn(nam)))), vp(v(den), vp(pp(p(o), p(gan)), np(n(truong))))

- Nhà Nam thường đến thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nha,nam,thuong,den,thu,vien],[]).

Tree = s(np(n(nha), np(ppn(nam)))), vp(advp(adv(thuong)), v(den), np(n(thu\_vien)))

- Nhà Nam thường đến trường

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nha,nam,thuong,den,truong],[]).

Tree = s(np(n(nha), np(ppn(nam)))), vp(advp(adv(thuong)), v(den), np(n(truong)))

- Nhà Nam thường đến gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nha,nam,thuong,den,o,gan,thu,vien],[]).

Tree = s(np(n(nha), np(ppn(nam)))), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(thu\_vien))))

- Nhà Nam thường đến gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,thuong,den,o,gan,truong],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(truong)))))

- Nhà Nam thường đến ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,thuong,den,o,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(thu_vien)))))

- Nhà Nam thường đến ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,thuong,den,o,gan,truong],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(truong)))))

- Nhà của Nam ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,nam,o,gan,thu,vien],[]).
Tree = s(np(n(nha), np(ppn(nam)))), vp(pp(p(o), p(gan)), np(n(thu_vien)))))

- Nhà của Nam gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,gan,thu,vien],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(pp(p(gan)), np(n(thu_vien))))

- Nhà của Nam thường ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,thuong,o,gan,thu,vien],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(advp(adv(thuong)), vp(pp(p(o), p(gan)), np(n(thu_vien)))))

- Nhà của Nam thường gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,thuong,gan,thu,vien],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(advp(adv(thuong)), vp(pp(p(gan)), np(n(thu_vien)))))

- Nhà của Nam đến thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,den,thu,vien],[]).
Tree = s(np(n(nha), np(adjp(adj(cua)), ppn(nam))), vp(v(den), np(n(thu_vien))))

- Nhà của Nam đến trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,den,truong],[]).
Tree = s(np(n(nha), np(adjp(adj(cua), ppn(nam)))), vp(v(den), np(n(truong))))

- Nhà của Nam đến gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,den,gan,thu,vien],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(v(den), vp(pp(p(gan)), np(n(thu_vien)))))

- Nhà của Nam đến gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,den,gan,truong],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam)))), vp(v(den), vp(pp(p(gan)), np(n(truong)))))

- Nhà của Nam đến ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,den,o,gan,thu,vien],[]).

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(v(den), vp(pp(p(o), p(gan)), np(n(thu_vien))))
---------------------------------------------------------------------------------------------------

- Nhà của Nam đến ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nha,cua,nam,den,o,gan,truong],[]).
-----------------------------------------------

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(v(den), vp(pp(p(o), p(gan)), np(n(truong))))
-------------------------------------------------------------------------------------------------

- Nhà của Nam thường đến thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nha,cua,nam,thuong,den,thu,vien],[]).
--------------------------------------------------

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(advp(adv(thuong)), v(den), np(n(thu_vien)))
------------------------------------------------------------------------------------------------

- Nhà của Nam thường đến trường

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nha,cua,nam,thuong,den,truong],[]).
------------------------------------------------

Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(advp(adv(thuong)), v(den), np(n(truong)))
----------------------------------------------------------------------------------------------

- Nhà của Nam thường đến gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,thuong,den,gan,thu,vien],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(advp(adv(thuong)), v(den), vp(pp(p(gan)), np(n(thu_vien)))))

- Nhà của Nam thường đến gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,thuong,den,gan,truong],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(advp(adv(thuong)), v(den), vp(pp(p(gan)), np(n(truong)))))

- Nhà của Nam thường đến ở gần thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,thuong,den,o,gan,thu,vien],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(thu_vien)))))

- Nhà của Nam thường đến ở gần trường

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nha,cua,nam,thuong,den,o,gan,truong],[]).
Tree = s(np(n(nha), pp(p(cua), np(ppn(nam))))), vp(advp(adv(thuong)), v(den), vp(pp(p(o), p(gan)), np(n(truong)))))

#### 2.4. Các câu phái sinh từ câu a, d

- Nam đến thư viện mua sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,mua,sach],[]).
Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(sach))))

- Nam đến thư viện mua cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,mua,cuon,sach],[]).
Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(cuon_sach))))

- Nam đến thư viện mua mấy cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,mua,may,cuon,sach],[]).
Tree = s(np(ppn(nam)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(adjp(adj(may)), n(cuon_sach))))

- Nam đến thư viện mua sách mới

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,thu,vien,mua,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(v(den), np(n(thu\_vien)), vp(v(mua)), np(n(sach), advp(adv(moi)))))

- Nam đến thư viện mua cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,den,thu,vien,mua,may,cuon,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(v(den), np(n(thu\_vien)), vp(v(mua)), np(adjp(adj(may)), n(cuon\_sach), adjp(adj(moi)))))

- Nam đến thư viện mua mấy cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,den,thu,vien,mua,may,cuon,sach,moi],[]).

Tree = s(np(ppn(nam)), vp(v(den), np(n(thu\_vien)), vp(v(mua)), np(adjp(adj(may)), n(cuon\_sach), adjp(adj(moi)))))

- Nam mua sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[nam,mua,sach,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(v(mua), np(n(sach), pp(p(o), np(n(thu\_vien)))))

- Nam mua cuốn sách ở thư viện

Thực hiện truy vấn trên công cụ:



Prolog
?- s(Tree,[nam,mua,cuon,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon_sach), pp(p(o), np(n(thu_vien))))))

- Nam mua mấy cuốn sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,cuon,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon_sach), pp(p(o), np(n(thu_vien))))))

- Nam mua sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,sach,moi,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(mua), np(n(sach), advp(adv(moi)))), pp(p(o), np(n(thu_vien))))

- Nam mua cuốn sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,cuon,sach,moi,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(v(mua), np(n(cuon_sach), advp(adv(moi)))), pp(p(o), np(n(thu_vien))))

- Nam mua mấy cuốn sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,mua,may,cuon,sach,moi,o,thu,vien],[]).  Tree = s(np(ppn(nam)), vp(v(mua), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi))))) , pp(p(o), np(n(thu_vien))))

- Nam ở thư viện mua sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,o,thu,vien,mua,sach],[]).  Tree = s(np(ppn(nam), pp(p(o), np(n(thu_vien))))) , vp(v(mua), np(n(sach))))

- Nam ở thư viện mua cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,o,thu,vien,mua,cuon,sach],[]).  Tree = s(np(ppn(nam), pp(p(o), np(n(thu_vien))))) , vp(v(mua), np(n(cuon_sach))))

- Nam ở thư viện mua mấy cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,o,thu,vien,mua,may,cuon,sach],[]).  Tree = s(np(ppn(nam), pp(p(o), np(n(thu_vien))))) , vp(v(mua), np(adjp(adj(may)), n(cuon_sach))))

- Nam ở thư viện mua sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,o,thu,vien,mua,sach,moi],[]).</pre> <pre>Tree = s(np(ppn(nam), pp(p(o), np(n(thu_vien)))), vp(v(mua), np(n(sach), advp(adv(moi)))))</pre>

- Nam ở thư viện mua cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,o,thu,vien,mua,cuon,sach,moi],[]).</pre> <pre>Tree = s(np(ppn(nam), pp(p(o), np(n(thu_vien)))), vp(v(mua), np(n(cuon_sach), advp(adv(moi)))))</pre>

- Nam ở thư viện mua mấy cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,o,thu,vien,mua,may,cuon,sach,moi],[]).</pre> <pre>Tree = s(np(ppn(nam), pp(p(o), np(n(thu_vien)))), vp(v(mua), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi)))))</pre>

- Nam thường đến thư viện mua sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,mua,sach],[]).</pre>

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(sach)))))
-----------------------------------------------------------------------------------------------------

- Nam thường đến thư viện mua cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,den,thu,vien,mua,cuon,sach],[]).
--------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(cuon_sach)))))
----------------------------------------------------------------------------------------------------------

- Nam thường đến thư viện mua mấy cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,den,thu,vien,mua,may,cuon,sach],[]).
------------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(adjp(adj(may)), n(cuon_sach)))))
--------------------------------------------------------------------------------------------------------------------------

- Nam thường đến thư viện mua sách mới

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,den,thu,vien,mua,sach,moi],[]).
-------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(sach), advp(adv(moi)))))
--------------------------------------------------------------------------------------------------------------------

- Nam thường đến thư viện mua cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,mua,cuon,sach,moi],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(v(den), np(n(thu_vien))),</pre> <pre>vp(v(mua)), np(n(cuon_sach), advp(adv(moi)))))</pre>

- Nam thường đến thư viện mua mấy cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,mua,may,cuon,sach,moi],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), vp(v(den), np(n(thu_vien))),</pre> <pre>vp(v(mua)), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi)))))</pre>

- Nam thường mua sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,mua,sach,o,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(mua), np(n(sach), pp(p(o),</pre> <pre>np(n(thu_vien)))))</pre>

- Nam thường mua cuốn sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,mua,cuon,sach,o,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(mua), np(n(cuon_sach), pp(p(o),</pre> <pre>np(n(thu_vien)))))</pre>

- Nam thường mua mấy cuốn sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,mua,may,cuon,sach,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(mua), np(adjp(adj(may)), n(cuon_sach)), vp(pp(p(o)), np(n(thu_vien)))))

- Nam thường mua sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,mua,sach,moi,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(mua), np(n(sach)), vp(advp(adv(moi)), vp(pp(p(o)), np(n(thu_vien)))))

- Nam thường mua cuốn sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,mua,cuon,sach,moi,o,thu,vien],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(mua), np(n(cuon_sach)), vp(advp(adv(moi)), vp(pp(p(o)), np(n(thu_vien)))))

- Nam thường mua mấy cuốn sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,thuong,mua,may,cuon,sach,moi,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(mua), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi))), vp(pp(p(o)), np(n(thu_vien)))))
----------------------------------------------------------------------------------------------------------------------------------------

- Nam thường ở thư viện mua sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,o,thu,vien,mua,sach],[]).
-------------------------------------------------

Tree = s(np(ppn(nam), pp(advp(adv(thuong)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(sach)))))
------------------------------------------------------------------------------------------------

- Nam thường ở thư viện mua cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,o,thu,vien,mua,cuon,sach],[]).
------------------------------------------------------

Tree = s(np(ppn(nam), pp(advp(adv(thuong)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(cuon_sach)))))
-----------------------------------------------------------------------------------------------------

- Nam thường ở thư viện mua mấy cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,o,thu,vien,mua,may,cuon,sach],[]).
----------------------------------------------------------

Tree = s(np(ppn(nam), pp(advp(adv(thuong)), p(o), np(n(thu_vien)))), vp(v(mua), np(adjp(adj(may)), n(cuon_sach)))))
---------------------------------------------------------------------------------------------------------------------

- Nam thường ở thư viện mua sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,o,thu,vien,mua,sach,moi],[]).</pre> <pre>Tree = s(np(ppn(nam), pp(advp(adv(thuong)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(sach), advp(adv(moi)))))</pre>

- Nam thường ở thư viện mua cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,o,thu,vien,mua,cuon,sach],[]).</pre> <pre>Tree = s(np(ppn(nam), pp(advp(adv(thuong)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(cuon_sach)))))</pre>

- Nam thường ở thư viện mua mấy cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,o,thu,vien,mua,may,cuon,sach],[]).</pre> <pre>Tree = s(np(ppn(nam), pp(advp(adv(thuong)), p(o), np(n(thu_vien)))), vp(v(mua), np(adjp(adj(may)), n(cuon_sach)))))</pre>

- Nam mới mua sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,moi,mua,sach,o,thu,vien],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach), pp(p(o), np(n(thu_vien)))))</pre>



- Nam mới mua cuốn sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,cuon,sach,o,thu,vien],[]).  Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(cuon_sach), pp(p(o), np(n(thu_vien))))))

- Nam mới mua mấy cuốn sách ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,may,cuon,sach,o,thu,vien],[]).  Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may)), n(cuon_sach)), vp(pp(p(o)), np(n(thu_vien))))))

- Nam mới mua sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,sach,moi,o,thu,vien],[]).  Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach), pp(advp(adv(moi)), p(o), np(n(thu_vien))))))

- Nam mới mua cuốn sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,mua,sach,cuon,sach,moi,o,thu,vien],[]).

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(n(sach), np(n(cuon_sach), pp(advp(adv(moi)), p(o), np(n(thu_vien)))))))
------------------------------------------------------------------------------------------------------------------------------

- Nam mới mua mấy cuốn sách mới ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,moi,mua,may,cuon,sach,moi,o,thu,vien],[]).
-----------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), v(mua), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi))), vp(pp(p(o), np(n(thu_vien))))))
-------------------------------------------------------------------------------------------------------------------------------------

- Nam mới ở thư viện mua sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,moi,o,thu,vien,mua,sach],[]).
----------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(p(o), np(n(thu_vien)), vp(v(mua)), np(n(sach)))))
------------------------------------------------------------------------------------------------

- Nam mới ở thư viện mua cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,moi,o,thu,vien,mua,sach],[]).
----------------------------------------------

Tree = s(np(ppn(nam), pp(advp(adv(moi)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(sach))))
--------------------------------------------------------------------------------------------

- Nam mới ở thư viện mua mấy cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,o,thu,vien,mua,may,cuon,sach],[]).
Tree = s(np(ppn(nam), pp(advp(adv(moi)), p(o), np(n(thu_vien)))), vp(v(mua), np(adjp(adj(may)), n(cuon_sach)))))

- Nam mới ở thư viện mua sách mới

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,o,thu,vien,mua,sach,moi],[]).
Tree = s(np(ppn(nam), pp(advp(adv(moi)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(sach), advp(adv(moi)))))

- Nam mới ở thư viện mua cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,o,thu,vien,mua,cuon,sach,moi],[]).
Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(o), np(n(thu_vien))), vp(v(mua)), np(n(cuon_sach), advp(adv(moi)))))

- Nam mới ở thư viện mua mấy cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,moi,o,thu,vien,mua,cuon,sach,moi],[]).
Tree = s(np(ppn(nam), pp(advp(adv(moi)), p(o), np(n(thu_vien)))), vp(v(mua), np(n(cuon_sach), advp(adv(moi)))))

- Nam mới đến thư viện mua sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,moi,den,thu,vien,mua,sach],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(den), np(n(thu_vien)),</pre> <pre>vp(v(mua)), np(n(sach)))))</pre>

- Nam mới đến thư viện mua cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,moi,den,thu,vien,mua,cuon,sach],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(den), np(n(thu_vien)),</pre> <pre>vp(v(mua)), np(n(cuon_sach)))))</pre>

- Nam mới đến thư viện mua mấy cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,moi,den,thu,vien,mua,may,cuon,sach],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(den), np(n(thu_vien)),</pre> <pre>vp(v(mua)), np(adjp(adj(may)), n(cuon_sach)))))</pre>

- Nam mới đến thư viện mua sách mới

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,moi,den,thu,vien,mua,sach,moi],[]).</pre>

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(sach), advp(adv(moi))))))
------------------------------------------------------------------------------------------------------------------

- Nam mới đến thư viện mua cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,moi,den,thu,vien,mua,cuon,sach,moi],[]).
---------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(n(cuon_sach), advp(adv(moi))))))
-----------------------------------------------------------------------------------------------------------------------

- Nam mới đến thư viện mua mấy cuốn sách mới

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,moi,den,thu,vien,mua,may,cuon,sach,moi],[]).
-------------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(moi)), vp(v(den), np(n(thu_vien)), vp(v(mua)), np(adjp(adj(may)), n(cuon_sach), adjp(adj(moi))))))
---------------------------------------------------------------------------------------------------------------------------------------

## 2.5. câu phái sinh từ câu a, e

- Nam đến thư viện tặng lan cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,thu,vien,tang,lan,cuon,sach],[]).
------------------------------------------------------

Tree = s(np(ppn(nam)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach)))))
-----------------------------------------------------------------------------------------------------

- Nam đến thư viện tặng lan cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,tang,lan,cuon,sach,hay],[]).  Tree = s(np(ppn(nam)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach))), adjp(adj(hay)))))

- Nam đến thư viện tặng lan cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,tang,lan,cuon,sach,rat,hay],[]).  Tree = s(np(ppn(nam)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach))), adjp(advp(adv(rat)), adj(hay)))))

- Nam đến thư viện tặng lan một cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,tang,lan,mot,cuon,sach],[]).  Tree = s(np(ppn(nam)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach)))))

- Nam đến thư viện tặng lan một cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[nam,den,thu,vien,tang,lan,mot,cuon,sach,hay],[]).

Tree = s(np(ppn(nam)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach), adjp(adj(hay))))))))
---------------------------------------------------------------------------------------------------------------------------------------

- Nam đến thư viện tặng lan một cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,den,thu,vien,tang,lan,mot,cuon,sach,rat,hay],[]).
------------------------------------------------------------------

Tree = s(np(ppn(nam)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach), adjp(advp(adv(rat)), adj(hay))))))))
-------------------------------------------------------------------------------------------------------------------------------------------------------

- Cuốn sách Nam tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,nam,tang,lan,o,thu,vien],[]).
----------------------------------------------------

Tree = s(np(n(cuon_sach)), vp(np(ppn(nam)), vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien))))))))
---------------------------------------------------------------------------------------------------------

- Cuốn sách hay Nam tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,hay,nam,tang,lan,o,thu,vien],[]).
--------------------------------------------------------

Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(nam)), vp(v(tang), vp(np(ppn(lan)), pp(p(o),np(n(thu_vien))))))))
------------------------------------------------------------------------------------------------------------------------

- Cuốn sách rất hay Nam tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,rat,hay,nam,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(nam)), vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien))))))</pre>

- Cuốn sách rất hay Nam đến tặng Lan

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,rat,hay,nam,den,tang,lan],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(nam)), vp(v(den), vp(v(tang), np(ppn(lan)))))</pre>

- Cuốn sách hay Nam đến tặng Lan

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,hay,nam,den,tang,lan],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(nam)), vp(v(den), vp(v(tang), np(ppn(lan)))))</pre>

- Cuốn sách rất hay Nam đến tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,rat,hay,nam,den,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(nam)), vp(v(den), vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien))))))</pre>



- Cuốn sách hay Nam đến tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,hay,nam,den,tang,lan,o,thu,vien],[]).</pre> <pre>Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay))), ppn(nam)), vp(v(den),</pre> <pre>vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien))))))</pre>

- Lan đến thư viện tặng Nam cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[lan,den,thu,vien,tang,nam,cuon,sach],[]).</pre> <pre>Tree = s(np(ppn(lan)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)),</pre> <pre>np(n(cuon_sach)))))</pre>

- Lan đến thư viện tặng Nam cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[lan,den,thu,vien,tang,nam,cuon,sach,hay],[]).</pre> <pre>Tree = s(np(ppn(lan)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)),</pre> <pre>np(n(cuon_sach)), adjp(adj(hay)))))</pre>

- Lan đến thư viện tặng Nam cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[lan,den,thu,vien,tang,nam,cuon,sach,rat,hay],[]).</pre>

Tree = s(np(ppn(lan)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)), np(n(cuon_sach))), adjp(advp(adv(rat)), adj(hay))))))
---------------------------------------------------------------------------------------------------------------------------------------

- Lan đến thư viện tặng Nam một cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[lan,den,thu,vien,tang,nam,mot,cuon,sach],[]).
----------------------------------------------------------

Tree = s(np(ppn(lan)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)), np(adjp(adj(mot)), n(cuon_sach))))))
----------------------------------------------------------------------------------------------------------------------

- Lan đến thư viện tặng Nam một cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[lan,den,thu,vien,tang,nam,mot,cuon,sach,hay],[]).
--------------------------------------------------------------

Tree = s(np(ppn(lan)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)), np(adjp(adj(mot)), n(cuon_sach), adjp(adj(hay))))))
-------------------------------------------------------------------------------------------------------------------------------------

- Lan đến thư viện tặng Nam một cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[lan,den,thu,vien,tang,nam,mot,cuon,sach,rat,hay],[]).
------------------------------------------------------------------

Tree = s(np(ppn(lan)), vp(v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)), np(adjp(adj(mot)), n(cuon_sach), adjp(advp(adv(rat)), adj(hay))))))
-----------------------------------------------------------------------------------------------------------------------------------------------------

- Cuốn sách Lan tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,lan,tang,nam,o,thu,vien],[]).
Tree = s(np(n(cuon_sach)), vp(np(ppn(lan)), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien)))))))

- Cuốn sách hay Lan tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,hay,lan,tang,nam,o,thu,vien],[]).
Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(lan)), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien)))))))

- Cuốn sách rất hay Lan tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,rat,hay,lan,tang,nam,o,thu,vien],[]).
Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat)), adj(hay)), ppn(lan)), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien)))))))

- Cuốn sách rất hay Lan đến tặng Nam

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,rat,hay,lan,den,tang,nam],[]).
Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat)), adj(hay)), ppn(lan)), vp(v(den), vp(v(tang), np(ppn(nam))))))

- Cuốn sách hay Lan đến tặng Nam

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,hay,lan,den,tang,nam],[]).
Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay))), ppn(lan)), vp(v(den), vp(v(tang), np(ppn(nam))))))

- Cuốn sách Lan đến tặng Nam

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,lan,den,tang,nam],[]).
Tree = s(np(n(cuon_sach)), vp(np(ppn(lan)), vp(v(den), vp(v(tang), np(ppn(nam))))))

- Cuốn sách rất hay Lan đến tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,rat,hay,lan,den,tang,nam,o,thu,vien],[]).
Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat))), adj(hay))), ppn(lan)), vp(v(den), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien)))))))

- Cuốn sách hay Lan đến tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,hay,lan,den,tang,nam,o,thu,vien],[]).

Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay))), ppn(lan)), vp(v(den), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien))))))))
-------------------------------------------------------------------------------------------------------------------------------------

- Cuốn sách Lan đến tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,lan,den,tang,nam,o,thu,vien],[]).
--------------------------------------------------------

Tree = s(np(n(cuon_sach)), vp(np(ppn(lan)), vp(v(den), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien))))))))
--------------------------------------------------------------------------------------------------------------------

- Nam thường đến thư viện tặng lan cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,den,thu,vien,tang,lan,cuon,sach],[]).
-------------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach))))))
-------------------------------------------------------------------------------------------------------------------------

- Nam thường đến thư viện tặng lan cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[nam,thuong,den,thu,vien,tang,lan,cuon,sach,hay],[]).
-----------------------------------------------------------------

Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(lan)), np(n(cuon_sach)), adjp(adj(hay))))))
-----------------------------------------------------------------------------------------------------------------------------------------

- Nam thường đến thư viện tặng lan cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,tang,lan,cuon,sach,rat,hay],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)),</pre> <pre>vp(v(tang), np(ppn(lan)), np(n(cuon_sach)), adjp(advp(adv(rat)), adj(hay))))))</pre>

- Nam thường đến thư viện tặng lan một cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,tang,lan,mot,cuon,sach],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)),</pre> <pre>vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach))))))</pre>

- Nam thường đến thư viện tặng lan một cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,tang,lan,mot,cuon,sach,hay],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)),</pre> <pre>vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach), adjp(adj(hay))))))</pre>

- Nam thường đến thư viện tặng lan một cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[nam,thuong,den,thu,vien,tang,lan,mot,cuon,sach,rat,hay],[]).</pre> <pre>Tree = s(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)),</pre> <pre>vp(v(tang), np(ppn(lan)), np(adjp(adj(mot)), n(cuon_sach), adjp(advp(adv(rat)),</pre> <pre>adj(hay))))))</pre>

- Cuốn sách Nam thường tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,nam,thuong,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(ppn(nam)), vp(advp(adv(thuong)), v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien)))))))

- Cuốn sách hay Nam thường tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,hay,nam,thuong,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(tang), vp(np(ppn(lan)), vp(v(o), np(n(thu_vien)))))))

- Cuốn sách rất hay Nam thường tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,rat,hay,nam,thuong,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat)), adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien)))))))

- Cuốn sách rất hay Nam thường đến tặng Lan

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,rat,hay,nam,thuong,den,tang,lan],[]).

Tree = s(np(n(cuon\_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(tang), np(ppn(lan))))))

- Cuốn sách hay Nam thường đến tặng Lan

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[cuon,sach,hay,nam,thuong,den,tang,lan],[]).

Tree = s(np(n(cuon\_sach)), vp(np(adjp(adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(tang), np(ppn(lan))))))

- Cuốn sách Nam thường đến tặng Lan

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[cuon,sach,nam,thuong,den,tang,lan],[]).

Tree = s(np(n(cuon\_sach)), vp(np(ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(tang), np(ppn(lan))))))

- Cuốn sách rất hay Nam thường đến tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[cuon,sach,rat,hay,nam,thuong,den,tang,lan,o,thu,vien],[]).

Tree = s(np(n(cuon\_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu\_vien))))))



- Cuốn sách hay Nam thường đến tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,hay,nam,thuong,den,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien))))))))))</pre>

- Cuốn sách rất hay Nam thường đến tặng Lan ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[cuon,sach,rat,hay,nam,thuong,den,tang,lan,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat)), adj(hay)), ppn(nam)), vp(advp(adv(thuong)), v(den), vp(v(tang), vp(np(ppn(lan)), pp(p(o), np(n(thu_vien))))))))))</pre>

- Lan thường đến thư viện tặng Nam cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog
<pre>?- s(Tree,[lan,thuong,den,thu,vien,tang,nam,cuon,sach],[]).  Tree = s(np(ppn(lan)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)), np(n(cuon_sach))))))</pre>

- Lan thường đến thư viện tặng Nam cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[lan,thuong,den,thu,vien,tang,nam,cuon,sach,hay],[]).

Tree = s(np(ppn(lan)), vp(advp(adv(thuong)), v(den), vp(np(n(thu\_vien)),  
vp(v(tang), np(ppn(nam)), np(n(cuon\_sach)), adjp(adj(hay))))))

- Lan đến thư viện tặng Nam thường cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[lan,thuong,den,thu,vien,tang,nam,cuon,sach,rat,hay],[]).

Tree = s(np(ppn(lan)), vp(advp(adv(thuong)), v(den), vp(np(n(thu\_vien)),  
vp(v(tang), np(ppn(nam)), np(n(cuon\_sach)), adjp(advp(adv(rat)), adj(hay))))))

- Lan thường đến thư viện tặng Nam một cuốn sách

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[lan,thuong,den,thu,vien,tang,nam,mot,cuon,sach],[]).

Tree = s(np(ppn(lan)), vp(advp(adv(thuong)), v(den), vp(np(n(thu\_vien)),  
vp(v(tang), np(ppn(nam)), np(adjp(adj(mot)), n(cuon\_sach))))))

- Lan thường đến thư viện tặng Nam một cuốn sách hay

Thực hiện truy vấn trên công cụ:

Prolog

?- s(Tree,[lan,thuong,den,thu,vien,tang,nam,mot,cuon,sach,hay],[]).

Tree = s(np(ppn(lan)), vp(advp(adv(thuong)), v(den), vp(np(n(thu\_vien)),  
vp(v(tang), np(ppn(nam)), np(adjp(adj(mot)), n(cuon\_sach), adjp(adj(hay))))))

- Lan thường đến thư viện tặng Nam một cuốn sách rất hay

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[lan,thuong,den,thu,vien,tang,nam,mot,cuon,sach,rat,hay],[]).  Tree = s(np(ppn(lan)), vp(advp(adv(thuong)), v(den), vp(np(n(thu_vien)), vp(v(tang), np(ppn(nam)), np(adjp(adj(mot)), n(cuon_sach), adjp(advp(adv(rat)), adj(hay)))))))))

- Cuốn sách Lan thường tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,lan,thuong,tang,nam,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(ppn(lan)), vp(advp(adv(thuong)), v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien)))))))))

- Cuốn sách hay Lan thường tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,hay,lan,thuong,tang,nam,o,thu,vien],[]).  Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(lan)), vp(advp(adv(thuong)), v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu_vien)))))))))

- Cuốn sách rất hay Lan thường tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

Prolog
?- s(Tree,[cuon,sach,rat,hay,lan,thuong,tang,nam,o,thu,vien],[]).

<p>Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(lan)), vp(advp(adv(thuong))), v(tang), vp(np(ppn(nam))), pp(p(o), np(n(thu_vien))))))</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Cuốn sách rất hay Lan thường đến tặng Nam
- Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,rat,hay,lan,thuong,den,tang,nam],[]).
------------------------------------------------------------

<p>Tree = s(np(n(cuon_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(lan)), vp(advp(adv(thuong))), v(den), vp(v(tang), np(ppn(nam))))))</p>
-------------------------------------------------------------------------------------------------------------------------------------------------

- Cuốn sách hay Lan thường đến tặng Nam

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,hay,lan,thuong,den,tang,nam],[]).
--------------------------------------------------------

<p>Tree = s(np(n(cuon_sach)), vp(np(adjp(adj(hay)), ppn(lan)), vp(advp(adv(thuong))), v(den), vp(v(tang), np(ppn(nam))))))</p>
--------------------------------------------------------------------------------------------------------------------------------

- Cuốn sách Lan thường đến tặng Nam

Thực hiện truy vấn trên công cụ:

Prolog
--------

?- s(Tree,[cuon,sach,lan,thuong,den,tang,nam],[]).
----------------------------------------------------

<p>Tree = s(np(n(cuon_sach)), vp(np(ppn(lan)), vp(advp(adv(thuong))), v(den), vp(v(tang), np(ppn(nam))))))</p>
----------------------------------------------------------------------------------------------------------------

- Cuốn sách rất hay Lan thường đến tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

### Prolog

?- s(Tree,[cuon,sach,rat,hay,lan,thuong,den,tang,nam,o,thu,vien],[]).

Tree = s(np(n(cuon\_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(lan)),  
vp(advp(adv(thuong))), v(den), vp(v(tang), vp(np(ppn(nam)), pp(p(o),  
np(n(thu\_vien)))))))))

- Cuốn sách hay Lan thường đến tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

### Prolog

?- s(Tree,[cuon,sach,hay,lan,thuong,den,tang,nam,o,thu,vien],[]).

Tree = s(np(n(cuon\_sach)), vp(np(adjp(adj(hay))), ppn(lan)), vp(advp(adv(thuong))),  
v(den), vp(v(tang), vp(np(ppn(nam)), pp(p(o), np(n(thu\_vien)))))))))

- Cuốn sách rất hay Lan thường đến tặng Nam ở thư viện

Thực hiện truy vấn trên công cụ:

### Prolog

?- s(Tree,[cuon,sach,rat,hay,lan,thuong,den,tang,nam,o,thu,vien],[]).

Tree = s(np(n(cuon\_sach)), vp(np(adjp(advp(adv(rat))), adj(hay)), ppn(lan)),  
vp(advp(adv(thuong))), v(den), vp(v(tang), vp(np(ppn(nam)), vp(v(o),  
np(n(thu\_vien)))))))))

## VI. Tài liệu tham khảo:

1. "Definite Clause Grammar for Natural Language Processing" – John L. Lloyd.
2. "Prolog Programming for Artificial Intelligence" – Ivan Bratko.
3. <https://vi.shaip.com/blog/what-is-nlp-how-it-works-benefits-challenges-examples/>
4. <https://mathnasium.edu.vn/goc-chia-se/toan-logic-la-gi/> (26/10/2024)
5. [http://vienngonnguhoc.gov.vn/bai-viet/avram-noam-chomsky-%E2%80%9Cnguoi-co-tri-tue-nhat-the-gioi%E2%80%9D\\_388.aspx](http://vienngonnguhoc.gov.vn/bai-viet/avram-noam-chomsky-%E2%80%9Cnguoi-co-tri-tue-nhat-the-gioi%E2%80%9D_388.aspx) (26/10/2024)
6. [https://vi.wikipedia.org/wiki/Donald\\_Knuth](https://vi.wikipedia.org/wiki/Donald_Knuth) (26/10/2024)
7. XỬ LÝ NGÔN NGỮ TỰ NHIÊN. Nguyễn Tuấn Đăng. 2017. Nhà xuất bản DHQP-HCM.
8. Definite Clause Grammars (DCGs) from  
[https://www.amzi.com/manuals/amzi/pro/ref\\_dcg.htm](https://www.amzi.com/manuals/amzi/pro/ref_dcg.htm) (28/10/2024)
9. Prolog  
[https://www.let.rug.nl/bos/lpn/?fbclid=IwY2xjawGRqBNleHRuA2FlbQIxMAABHYTR8qQYQVTPKtaGSrlcCdcGh5aVqSrzQrrynDnPXD4BmQMgmM\\_BMbeXEw\\_aem\\_BoHHgeXYADPtzRR\\_juooew](https://www.let.rug.nl/bos/lpn/?fbclid=IwY2xjawGRqBNleHRuA2FlbQIxMAABHYTR8qQYQVTPKtaGSrlcCdcGh5aVqSrzQrrynDnPXD4BmQMgmM_BMbeXEw_aem_BoHHgeXYADPtzRR_juooew) (28/10/2024)

Hết