**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---o0o---

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: HỆ CHUYÊN GIA**

**ĐỀ TÀI**: Nhận dạng loài vật

**Giảng viên hướng dẫn**: ThS. Trần Thanh Hùng.

Lớp KHMT3K5 Nhóm

|  |  |  |
| --- | --- | --- |
| **STT** | **Mã sinh viên** | **Họ tên** |
| 1 | 0541060190 | Mai Đăng Tùng |
| 2 |  | Hoàng Đức Thiện |
| 3 | 0541060192 | Nguyễn Cao Cường |

*Hà Nội 8-2013*

Hà Nội 12-2012

Mục lục

[Lời nói đầu 3](#_Toc365054518)

[**I.** **Giới thiệu hệ chuyên gia.** 4](#_Toc365054519)

[**1.** **Hệ chuyên gia là gì?** 4](#_Toc365054520)

[**2.** **Đặc trưng và ưu điểm của hệ chuyên gia** 6](#_Toc365054521)

[**3.** **Sự phát triển của công nghệ hệ chuyên gia** 7](#_Toc365054522)

[**II.** **Kiến trúc tổng quát của hệ chuyên gia** 8](#_Toc365054523)

[**1.** **Những thành phần cơ bản của một hệ chuyên gia** 8](#_Toc365054524)

[**2.** **Một số mô hình kiến trúc hệ chuyên gia** 10](#_Toc365054525)

[**3.** **Biểu diễn tri thức trong các hệ chuyên gia** 11](#_Toc365054526)

[**III.** **Kỹ thuật suy diễn tiến trong hệ chuyên gia** 16](#_Toc365054527)

[**IV.** **Xây dựng hệ chuyên gia chuẩn đoán loài vật.** 16](#_Toc365054528)

[**1.** **Giới thiệu về hệ chuyên gia chuẩn đoán loài vật.** 17](#_Toc365054529)

[**2.** **Giao diện chương trình.** 17](#_Toc365054530)

[**3.** **Code mã chương trình**. 19](#_Toc365054531)

# Lời nói đầu

Ngày nay không ai có thể phủ nhận vai trò cực kỳ quan trọng của máy tính trong nghiên cứu khoa học kỹ thuật cũng như đời sống.Máy tính đã làm được những điều kỳ diệu và giải được những vấn đề tưởng chừng như nan giải.Càng ngày càng có nhiều người tự hỏi ,liệu máy tính có khả năng suy nghĩ như con người hay chưa,chúng ta không trả lời câu hỏi đấy.Thay vào đó,chúng ta sẽ nêu ra những khác biệt chủ yếu giữa cách làm việc của máy tính và bộ óc con người.

Một máy tính dù có mạnh đến đâu đi nữa,đều phải làm việc theo một chương trình chính xác đã được hoạch định sẵn bởi các chuyên gia.Bài toán càng phức tạp thì việc lập trình càng công phu.Trong khi đó người làm việc bằng cách học hỏi và rèn luyện ,khi làm việc con người có khả năng liên tưởng,kết nối sự việc này với sự việc khác và quan trọng hơn hết họ có thể sáng tạo.

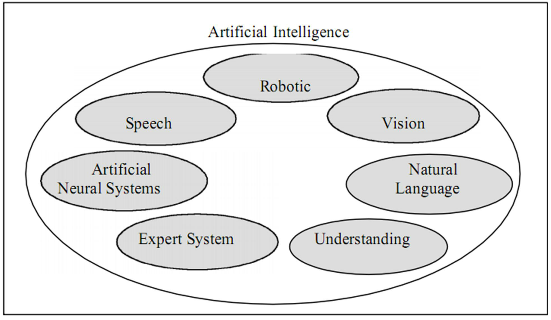
Do khả năng liên tưởng ,con người có thể dễ dàng làm nhiều điều mà việc lập trình máy tính đòi hỏi mất nhiều công sức.Chẳng hạn như việc chuẩn đoán bệnh .Một chuyên gia có thể phấn đoán được bênh nhân đó mắc bệnh gì . Một người bình thường cũng có thể đoán nhận được bệnh đó là bệnh gì thông qua học hỏi.Nhưng thật khó dạy cho máy tính làm được những việc ấy .Từ lâu các nhà khoa học đã nhận thấy những ưu điểm của bộ óc con người và tìm cách bắt chước để thực hiện trên những máy tính ,tạo cho nó có khả năng nhận biết, chuẩn đoán như các chuyên gia.Các phương pháp suy diễn đã ra đời từ nỗ lực đó.Nó thực sự được chú ý nhanh chóng trở thành một hướng nghiên cứu đầy triển vọng trong mục đích xây dựng các máy thông minh tiến gần tới trí tuệ con người .Đặc biệt là lĩnh vực chuẩn đoán, hệ chuyên gia.

Và bây giờ chúng ta sẽ đi tìm hiểu về việc làm sao mà máy tính có thể phán đoán ,tư duy như con người, với mức độ chinh xác cao đã được kiểm chứng.

1. **Giới thiệu hệ chuyên gia.**
2. **Hệ chuyên gia là gì?**

Theo E.Feigenbaum: Hệ chuyên gia(Expert System)là một chương trình máy tính thông minh sử dụng tri thức (Knowledge) và các thủ tục suy luận (inference procedures) để giải những bài toán tương đối khó khăn đòi hỏi những chuyên gia mới giải được.

Chuyên gia là một hệ thống tin học có thể mô phỏng (emulates) năng lực quyết đoán (decision) và hành động (making abilily) của một chuyên gia(con người).Hệ chuyên gia là một trong những lĩnh vực ứng dụng của trí tuệ nhân tạo (Artificial Intelligence) như hình dưới đây.



Hình 1.1. Một số lĩnh vực ứng dụng của trí tuệ nhân tạo

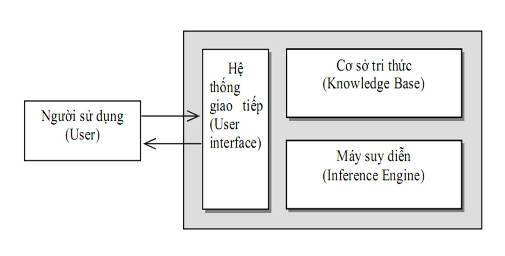
Hệ chuyên gia sử dụng các tri thức của những chuyên gia để giải quyết các vấn đề(bài toán) khác nhau thuộc mọi lĩnh vực.

Tri thức (knowledge) trong hệ chuyên gia phản ánh sự tinh thông được tích tụ từ sách vở ,tạp chí ,từ các chuyên gia hay các nhà bác học .Các thuật ngữ hệ chuyên gia ,hệ thống dựa trên tri thức (knowledge-based system) hay hệ chuyên gia dựa trên tri thức (knowledge-based expert system) thowngf có cùng nghĩa.

Một hệ chuyên gia bao gồm ba thành phần chính là cơ sở tri thức (knowledge based), máy suy diễn hay mô tơ suy diễn (inference engine),và hệ thống giao tiếp với người sử dụng (user interface).Cơ sở tri thức chứa các tri thức để từ đó ,máy suy diễn tạo ra câu trả lời cho người sử dụng thông qua hệ thống giao tiếp.

Người sử dung(user) cung cấp các sự kiện (facts) là những gì đã biết ,đã có thật hay những thông tin có ích cho hệ chuyên gia và nhận được những câu trả lời là những lời khuyên hay những gợi ý đúng đắn(expertise).

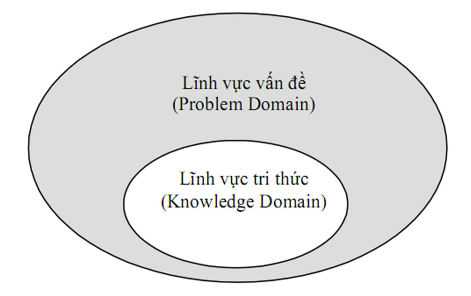
Hoạt động của một hệ chuyên gia dựa trên tri thức được minh họa như sau:



Hình 1.2. Hoạt động của hệ chuyên gia

Mỗi hệ chuyên gia chỉ đặc trưng cho một lĩnh vực vấn đề (problem domain) nào đó,như y học,tài chính,khoa học hay công nghệ,vv…,mà không phải cho bất cứ một lĩnh vực vấn đề nào.

Tri thức chuyên gia để giải quyết một vấn đề đặc trưng được gọi là lĩnh vực tri thức(knowledge domain).



Hình 1.3. Quan hệ giữa lĩnh vực vấn đề và lĩnh vực tri thức.

Ví dụ: hệ chuyên gia về lĩnh vực y học để phát hiện các căn bệnh lây nhiễm sẽ có nhiều tri thức về một số triệu chứng lây bệnh ,lĩnh vực tri thức y học bao gồm các căn bệnh ,triệu chứng và chữa trị.

Chú ý rằng lĩnh vực tri thức hoàn toàn nằm trong lĩnh vực vấn đề .Phần bên ngoài lĩnh vực tri thức nói lên rằng không phải là tri thức cho mọi vấn đề.

Tùy theo yêu cầu của người sử dụng mà có nhiều cách nhìn nhận khác nhau về một hệ chuyên gia.

|  |  |
| --- | --- |
| Loại người sử dụng | Vấn đề đặt ra |
| Người quản trị | Tôi có thể dùng nó để làm gì? |
| Kỹ thuật viên | Làm cách nào để tôi vận hành nó tốt nhất |
| Nhà nghiên cứu | Làm sao để tôi có thể mở rông nó? |
| Người sử dụng | Nó sẽ giúp tôi cái gì?  Nó có rắc rối và tốn kém không?  Nó có đáng tin cậy không? |

1. **Đặc trưng và ưu điểm của hệ chuyên gia**

Có bốn đặc trưng cơ bản của một hệ chuyên gia:

* Hiệu quả cao (high performance). Khả năng trả lời và mức độ tinh thông bằng hoặc cao hơn so với chuyên gia trong cùng lĩnh vực.
* Thời gian trả lời thỏa đáng (adequate response time). Thời gian trả lời hợp lý, bằng hoặc nhanh hơn so với chuyên gia để đi đến cùng một quyết định. Hệ chuyên gia là một hệ thống thời gian thực(real time system ).
* Độ tin cậy cao ( good reliability). Không thể xảy ra sự cố hoặc giảm sút độ tin cậy khi sử dụng .
* Dễ hiểu (understandable).Hệ chuyên gia giải thích các bước suy luận một cách dễ hiểu và nhất quán ,không giống như cách trả lời bí ẩn của các hộp đen(black box).

Những ưu điểm của hệ chuyên gia:

* Phổ cập(increased availability). Là sản phẩm chuyên gia ,được phát triển không ngừng với hiệu quả sử dụng là không phủ nhận.
* Giảm giá thành(reduced cost).
* Giảm rủi ro (reduced dangers).Giúp con người tránh được trong các môi trường rủi ro nguy hiểm.
* Tính thường trực(permanance ). Bất kể lúc nào cũng có thể khai thác sử dụng ,trong khi con người có thể mệt mỏi ,nghỉ ngơi hay vắng mặt.
* Đa lĩnh vực (multiple expertise ).Chuyên gia về nhiều lĩnh vực khác nhau và được khai thác đồng thời bất kể thời gian sử dụng.
* Độ tin cậy (increased relialility).Luôn đảm bảo độ tin cậy khi khai thác.
* Khả năng giảng giải (explanation ).Câu trả lời và mức độ tinh thông được giảng giải rõ ràng,chi tiết ,dễ hiểu.
* Khả năng trả lời (fast reponse).Trả lời theo thời gian thực,khách quan.
* Tính ổn định,suy luận có lý và đầy đủ mọi lúc mọi nơi (steady,une motional,and complete response at all times).
* Trợ giúp thông minh như một người hướng dẫn (intelligent – tutor).
* Có thể truy cập như một cơ sở dữ liêu thông minh (intelligent database ).

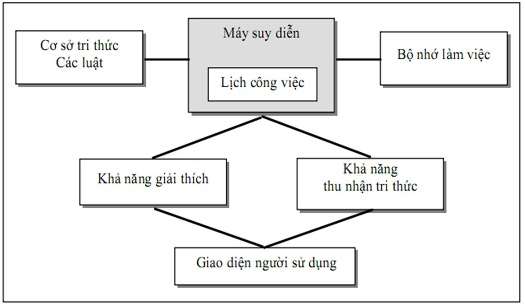
1. **Sự phát triển của công nghệ hệ chuyên gia**

Sau đây là một số sự kiện quan trọng trong lịch sự phát triển của công nghệ hệ chuyên gia (expert system technology).

|  |  |
| --- | --- |
| Năm | Các sự kiện |
| 1943 | Dịch vụ bưu điện, mô hình neural của Mc Culloch and Pitts Model |
| 1954 | Thuật toán Markov điều kiện thực thi các luật |
| 1956 | Hội thảo Dartmouth,lý luận logic,tìm kiếm nghiệm suy (heuristic search),thống nhất thuật ngữ trí tuệ nhân tạo. |
| 1957 | Rosenblatt phát minh khả năng nhận thức,Newell,Shaw và Simon đề suất giải bài toán tổng quát (GPS: Genenal Problem Solver ) |
| 1958 | Mc Carthy đề xuất ngôn ngữ trí tuệ nhân tạo LISA( LISA AI language) |
| 1962 | Nguyên lý Rosenblatt’s về chức năng thần kinh trong nhận thức (Rosenblatt’s Principles of Neurodynamicdynamics on Perceptions) |
| 1965 | Phương pháp hợp giải Robinson.Ứng dụng logic mờ (fuzzy logic) trong suy luận về các đối tượng mờ (fuzzy object) của Zadeh.Xây dựng hệ chuyên gia đầu tiên về nha khoa DENDRAL (Feigenbaum,Buchanan,et.al) |
| 1968 | Mạng ngữ nghĩa (semantic nets), mô hình bộ nhớ kết hợp (asociative memory model) của Quillian |
| 1969 | Hệ chuyên gia về toán học MACSYMA (Martin and Moses) |
| 1970 | Ứng dụng ngôn ngữ PROLOG(Colmerauser,Roussell,et,al.) |
| 1971 | Hệ chuyên gia HEARSAY I về nhận dạng tiếng nói (speech recognition). Xây dựng các luật giải bài toán con người (Human Problem Solving ppularizes rules (Newell abd Simon) |
| 1973 | Hệ chuyên gia MYCIN về chuẩn trị y học (Shortli ffe,et,al.) |
| 1975 | Lý thuyết khung (frames),biểu diễn tri thức (knowledge reresentation ) (Mínky) |
| 1976 | Toán nhân tạo (AM: Artificial Mathematician) (Lenat).Lý thuyết Dempster-Shafer về tính hiển nhiên của lập luận không chắc chắn (Dempster-Shafer theory of Evidence for reason ubder uncertainty).Ứng dụng hệ chuyên gia PROSPECTOR trong khai thác hầm mỏ (Duda,Har) |
| 1977 | Sử dụng ngôn ngữ chuyên gia OPS( OPS expert system shell ) trong hệ chuyên gia XCON/R1 (Forgy) |
| 1978 | Hệ chuyên gia XCON/R1 (McDermott ,DEC) để bảo trì hệ thống máy tính DEC (DEC computer systems) |
| 1979 | Thuật toán mạng về so khớp nhanh (rete algorithm for fast pattern matching) của Forgy; thương mại hóa các ứng dụng về trí tuệ nhân tạo |
| 1980 | Ký hiệu học (symbolics), Xây dựng các máy LISP (LISP machines) từ LMI. |
| 1982 | Hệ chuyên gia về toán học (SMP math system ) ; mạng nơ-ron Hopfield (Hopfield neural net ); dự án xây dựng máy tính thông minh thế hệ 5 tại nhật bản (Japanese Fifth Generation Project to develop intelligent computers) |
| 1983 | Bộ công cụ phục vụ hệ chuyên gia KEE (KEE expert system tool ) (intrlli Corp ) |
| 1985 | Bộ công cụ phục vụ hệ chuyên gia CLIPS (CLIPS expert system tool (NASA) |

1. **Kiến trúc tổng quát của hệ chuyên gia**
2. **Những thành phần cơ bản của một hệ chuyên gia**

Một hệ chuyên gia kiểu mẫu bao gồm bảy thành phần cơ bản sau:

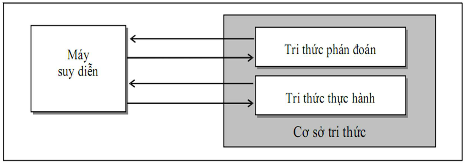


Hình 1.4. Những thành phần cơ bản của một hệ chuyên gia

* Cơ sở tri thức( knowledge base ).Gồm các phần tử hay đơn vị tri thức,thông thường được gọi là luật (rule),được tổ chức như một cơ sở dữ liệu.
* Máy suy diễn(inference engine).Công cụ (chương trình hay bộ xử lý)tạo ra sự suy luận bằng cách quyết định xem những luật nào sẽ làm thỏa mãn các sự kiện ,các đối tượng ,chọn ưu tiên các luật thỏa mãn,thực hiện các luật có tính ưu tiên cao nhất .
* Lịch công việc (agenda).Danh sách các luật ưu tiên do máy suy diễn tạo ra thỏa mãn các sự kiện,các đối tượng có mặt trong bộ nhớ làm việc.
* Bộ nhớ làm việc ( working memory). Cơ sở dữ liệu toàn cục chứ các sự kiện phục vụ cho các luật.
* Khả năng giải thích (explanation facility ). Giải nghĩa cách lập luận của hệ thống cho người sử dụng .
* Khả năng thu nhận tri thức (explanation facility). Cho phép người sử dụng bổ sung các tri thức vào hệ thống một cách tự động thay vì tiếp nhận tri thức bằng cách mã hóa tri thức một cách tường minh.Khả năng thu nhận tri thức là yếu tố mặc nhiên của nhiều hệ chuyên gia.
* Giao diện người sử dụng (user interface ). Là nơi mà người sử dụng và hệ chuyên gia trao đổi với nhau .

Cơ sở tri thức còn được gọi là bộ nhớ sản xuất (production memeory ) trong hệ chuyên gia.Trong một cơ sở tri thức, người ta thường phân biệt hai loại tri thức là tri thức phân đoán(assertion knowledge) và tri thức thực hành (operating knowledge).

Các tri thức phán đoán mô tả các tình huống đã được thiết lập hoặc sẽ được thiết lập .Các tri thức thực hành thể hiện những hậu quả rút ra hay những thao tác cần phải hoàn thiện khi một tình huống đã được thiết lập hoặc sẽ được thiết lập trong lĩnh vực đang xét. Các tri thức thực hành thường được thể hiện bởi các biểu thức dễ hiểu và dễ triển khai thao tác đối với người sử dụng .



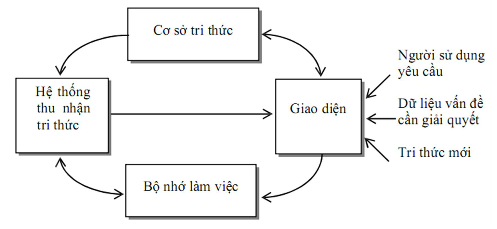
Hình 1.5. Quan hệ giữa máy suy diễn và cơ sở tri thức

Từ việc phân biệt hai loại tri thức ,người ta nói máy suy diễn là công cụ triển khai các cơ chế (hay kỹ thuật) tổng quát để tổ hợp các tri thức phán đoán và các tri thức thực hành .Hình trên mô tả quan hệ giữa máy suy diễn và cơ sở tri thức.

1. **Một số mô hình kiến trúc hệ chuyên gia**

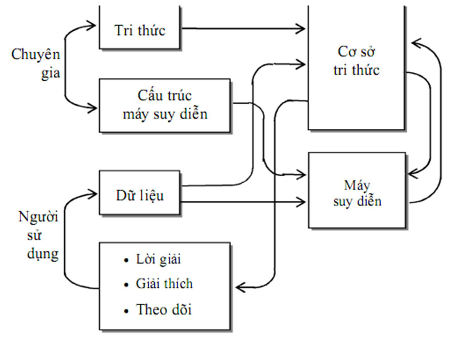
Có nhiều mô hình kiến trúc hệ chuyên gia theo tác giả khác nhau.Sau đây là một số mô hình.

1. Mô hình J.L.Ermine



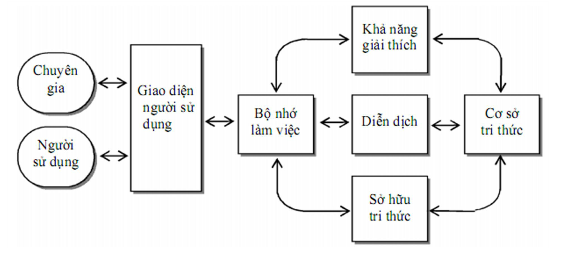
Hình 1.6.Kiến trúc hệ chuyên gia theo J.L.Ermine

1. Mô hình C.Ernest



Hình 1.7.Kiến trúc hệ chuyên gia theo C.Ernest

1. Mô hình E.V.Popov



Hình 1.8.Kiến trúc hệ chuyên gia theo E.V.Popov

1. **Biểu diễn tri thức trong các hệ chuyên gia**

Tri thức của một hệ chuyên gia có thể được biểu diễn theo nhiều cách khác nhau.Thông thường người ta sử dụng các cách sau đây:

* Biểu diễn tri thức bởi các luật sản xuất
* Biểu diễn tri thức nhờ mệnh đề logic
* Biểu diễn tri thức nhờ mạng ngữ nghĩa
* Biểu diễn tri thức nhờ ngôn ngữ nhân tạo

Ngoài ra ,người ta thường sử dụng cách biểu diễn tri thức nhờ các sự kiện không chắc chắn,nhờ bộ ba: đối tượng thuộc tính và giá trị (O-A-V :Object-Attribute-Value), nhờ khung(frame), vv… Tùy theo từng hệ chuyên gia, người ta có thể sử dụng một cách hoặc đồng thời cả nhiều cách.

1. Biểu diễn tri thức bằng các luật sản xuất

Hiện nay ,hầu hết các hệ chuyên gia đều là các hệ thống dựa trên luật ,bởi lý do như sau:

* Bản chất đơn thể (modular nature). Có thể đóng gói tri thức và mở rộng hệ chuyên gia một cách dễ dàng.
* Khả năng diễn giải dễ dàng (explanation facilities). Dễ dàng dùng luật để diễn giải vấn đề nhờ các tiền đề đặc tả chính xác các yếu tố vận dụng luật ,từ đó rút ra được kết quả .
* Tương tự quá trình nhận thức của con người .Dựa trên các công trình của Newell và Simon,các luật được xây dựng từ cách con người giải quyết vấn đề .Cách biểu diễn luật nhớ IF THEN đơn giản cho phép giải thích dễ dàng cấu trúc tri thức cần trích lọc .

Luật là một kiểu sản xuất được nghiên cứu từ những năm 1940.Trong một hệ thông dựa trên luật ,công cụ suy luận sẽ xác định những luật nào là tiên đề thỏa mãn các sự việc.

Các luật sản xuất thường được viết dưới dạng IF THEN.Có hai dạng :

IF<điều kiện>THEN<hành động>

Hoặc

IF<điều kiện>THEN<kết luận>DO<hành động>

Tùy theo hệ chuyên gia cụ thể mà mỗi luật có thể được đặt tên.Chẳng hạn mỗi luật có dạng Rule:tên. Sau phần tên là phần IF của luật.

Phần giũa IF và THEN là phần trái luật (LHS: Left-Hand-Side), có nội dung được gọi theo nhiều tên khác nhau,như tiền đề(antecedent),điều kiện (conditional part ),mẫu so khớp (pattern part).

Phần sau THEN là kết luận hay hậu quả (consequent). Một số hệ chuyên gia có thêm phần hành động (action) được gọi là phần phải luật (RHS: Right –Hand –Side).

Ví dụ:

Rule: Đèn đỏ

IF

Đèn đỏ sáng

THEN

Dừng

Rule: Đèn xanh

IF

Đèn xanh sáng

THEN

Đi

1. Bộ sinh của hệ chuyên gia.

Bộ sinh của hệ chuyên gia (expert-system generator ) là hợp của:

* Một máy suy diễn
* Một ngôn ngữ thể hiện tri thức( bên ngoài)
* Một tập hợp các cấu trúc và quy ước thể hiện các tri thức (bên trong)

Theo cách nào đó,các cấu trúc và các quy ước này xác định ,một cơ sở tri thức rỗng (hay rỗng bộ phận).Nhờ các tri thức chuyên môn để định nghĩa một hệ chuyên gia,người ta đã tạo ra một bộ sinh để làm đầy cơ sở tri thức.

Chẳng hạn EMYCIN là tên của một bộ sinh của hệ chuyên gia MYCIN và được tiếp tục áp dụng cho một số lĩnh vực.Hệ chuyên gia R1 được xây dựng từ bộ sinh OSP( là hệ thống luật được phát triển bởi Charles Forgy năm 1975 tại Carnegie-Mellon University).

1. Soạn thảo kết hợp các luật

Nói chung,tùy theo hệ chuyên gia mà những quy ước để tạo ra luật cũng khác nhau.Sự giống nhau cơ bản giữa các hệ chuyên gia về mặt ngôn ngữ là cách soạn thảo kết hợp (associative writing ) các luật.

Ở đây ,thuật ngữ soạn thảo kết hợp được chọn để gợi lên khái niệm về chế độ truy cập kết hợp (associative access) liên quan đến chế độ lưu trữ kết hợp (associative memory) là chế độ mà thông tin cần tìm kiếm được đọc không chỉ căn cứ vào địa chỉ đơn vị nhớ cụ thể mà còn căn cứ vào một thành phần nội dung của thông tin cần tìm kiếm chứa trong đó.

Soạn thảo kết hợp các luật gồm những quy ước như sau:

1. Mỗi luật do chuyên gia cung cấp phải định nghĩa được các điều kiện khởi động (tác nhân) hay tiền đề của luật ,nghĩa là các tình huống (được xác đinh bởi các quan hệ trên tập hợp dữ liêu đã cho) và hậu quả của luật ,để luật này có thể áp dụng.

Theo cách dùng thông thường ,người ta đặt tên riêng cho luật để chọn áp dụng,hoặc cung cấp một nhóm các sự kiện (fact) tương thích với điều kiện khởi động của luật .

1. Trong luật, không bao giờ người ta chỉ định một luật khác bởi tên riêng.

Ví dụ : Luật R sau đây tuân thủ hai đặc trưng:

IF bệnh nhân sốt AND tóc độ lắng huyết cầu trong máu tăng lên

THEN bệnh nhân nhiễm bệnh virut

Từ nội dung luật R người ta có thể vận dụng như sau:

* Khi xảy ra tình huống bệnh nhân bị sốt và tốc độ lắng huyết cầu trong máu tăng lên, thì “bệnh nhân sốt” và “tốc độ lắng huyết cầu trong máu tăng lên” là những điều kiện để khởi động luật .Hậu quả của luật là “Bệnh nhân nhiễm bệnh virut ”.Như vậy ,việc áp dụng luật sẽ dẫn đến một sự kiện mới được thiết lập từ đây trở đi :”Bệnh nhân nhiễm bệnh virut” .
* Khi muốn tạo sự kiện ”Bệnh nhân nhiễm bệnh virut”,thì điều kiện khởi động luật là :”Bệnh nhân nhiễm bệnh virut”.Hậu quả của luật sẽ là “bệnh nhân sốt” và “tốc độ lắng huyết cầu trong máu tăng lên”.Từ đây ,luật sẽ khởi động các sự kiện mới vừa được thiết lập “bệnh nhân sốt” và “tốc độ lắng huyết cầu trong máu tăng lên”.

Cách biểu diễn các điều kiện khởi động trong luật ohuf hợp với cách tư duy tự nhiên của các chuyên gia.Do vậy, người ta cũng dễ dàng thể hiện cũng như sửa đổi các tri thức tiếp nhận.

Như vậy, người ta không nhất thiết phải đặt tên cho các luật để có gọi đến khi cần,mà có thể khai thác thông tin từ các điều kiện khởi động của luật .Chẳng hạn từ luật R trên đây:

* Nếu tìm được các luật có khả năng thiết lập sự kiện “Bệnh nhân nhiễm bệnh virut ”,người ta sẽ để ý đến phần THEN của chúng như là các điều kiện khởi động.Luật R là một trong các luật có điều kiện khởi động tương ứng với lời gọi “Bệnh nhân nhiễm bệnh virut ”.
* Nếu tìm được các luật có khả năng đưa ra sự kiện “bệnh nhân sốt”,chỉ cần để ý đến phần IF của chúng là các điều kiện khởi động .Luật R là một trong các luật có điều kiện khởi động tương ứng với lời gọi “bệnh nhân sốt”.

Việc so sánh này giữa điều kiện khởi động các luật và sự kiện được xét tại một thời điểm đã cho (tùy theo trường hợp ,các sự kiện giả sử đã được thiết lập hay sẽ thiết lập) cho phép lọc (filter) các luật để giữ lại một số luật nào đó.Phần điều kiện của luật thường được gọi là bộ lọc,hay mẫu so khớp của luật đó.

Trong tin học cổ điển,mỗi thủ tục (đóng vai trò là một đơn vị ttri thức) thường được xác định và được gọi bởi tên của thủ tục .Lúc này, nếu muốn thêm vào hay lấy ra một thủ tục, người ta cần dự kiến các thay đổi trong toàn bộ thủ tục khác sử dụng đến thủ tục muốn thêm vào hay lấy ra này.Ngược lại, về nguyên tắc ,việc soạn thảo kết hợp cho phép tạo ra một luật mà không để ý đến sự hiện diện của các luật khác .Với mỗi luật, dù là của ai , một khi được đua vào trong cơ sở tri thức, thì chỉ cần để ý đến các biểu thức điều kiện để xác định nếu luật đó là áp dụng được va do vậy, có thể gọi tới nó hay không.Người ta cũng xem rằng các sự kiện được đưa vào như là hậu quả của một luật có thể giúp để gọi đến các luật khác nhờ các bộ lọc của chúng.

Như vậy, phương pháp soạn thảo kết hợp cho phép bổ sung và loại bỏ dễ dàng các luật mà không cần xem xét hậu quả của việc bổ sung và loại bỏ đó.Phương pháp soạn thảo kết hợp có vị trí quan trọng trong các hệ thống dựa trên luật của các hệ chuyên gia.Đó là các hệ thông suy diễn định hướng bởi các bộ lọc (PDISPatterm-Directed Inference System).

1. **Kỹ thuật suy diễn tiến trong hệ chuyên gia**

Có nhiều phương pháp tổng quát để suy luận trong các chiến lược giải quyết các vấn đề của hệ chuyên gia.Những phương pháp hay gặp là suy diễn tiến (forward chaining ),suy diễn lùi (backward chaining) và phối hợp hai phương pháp này (mixed chaining).Những phương pháp khác là phân tích phương tiện (means-end analysis ),rút gọn vấn đề (problem reduction),quay lui (backtracking),kiểm tra lập kế hoạch (plan-geenerate-test),lập kế hoạch phân cấp( heirachical planning)…

Sau đây sẽ giới thiệu về phương pháp suy luận theo suy diễn tiến:

Suy diễn tiến (forward charning) là lập luận từ các sự kiện ,sự việc để rút ra các kết luận.

Ví dụ: Nếu thấy trời mưa trước khi ra khỏi nhà (sự kiện ) thì phải lấy áo mưa (kết luận).

Trong phương pháp này, người sử dụng cung cấp các sự kiện cho hệ chuyên gia để hệ thống (máy suy diễn )tìm cách rút ra các kết luận có thể .Kết luận được xem là những thuộc tính có thể được gán giá trị .Trong số những kết luận này,có thể có những kết luận làm người sử dụng quan tâm,một số khác không nói lên điều gì,một số khác có thể vắng mặt.

Các sự kiện thường có dạng :

Atthibute=value

Lần lượt các sự kiện trong cơ sở tri thức được chọn và hệ thống xem xét tất cả các luât mà các sự kiện này xuất hiện như là tiên đề.Theo nguyên tắc lập luận trên , hệ thống sẽ lấy ra những luật thỏa mãn .Sau khi gán giá trị cho các thuộc tính thuộc kết luận tương ứng , người ta nói rằng các sự kiện đã được thỏa mãn. Các thuộc tính được gán giá trị sẽ là một phần của kết quả chuyên gia.Sau khi mọi sự kiện đã được xem xét , kết quả được xuất ra cho người sử dụng.

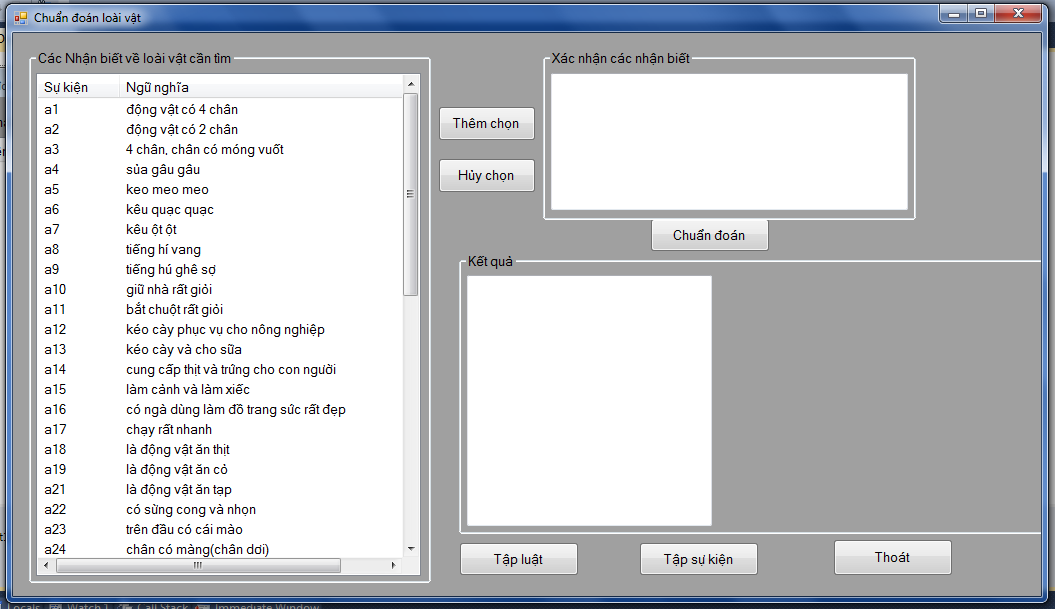
1. **Xây dựng hệ chuyên gia chuẩn đoán loài vật.**
2. **Giới thiệu về hệ chuyên gia chuẩn đoán loài vật.**

Hệ chuyên gia chuẩn đoán loài vật trợ giúp người xác định được loài vật thông qua đăc tính của chúng.Hệ thống sẽ dựa trên đặc tính của các loài động vật như sinh đẻ,cách thức hoạt động,giá tri của chúng với con người, vv…

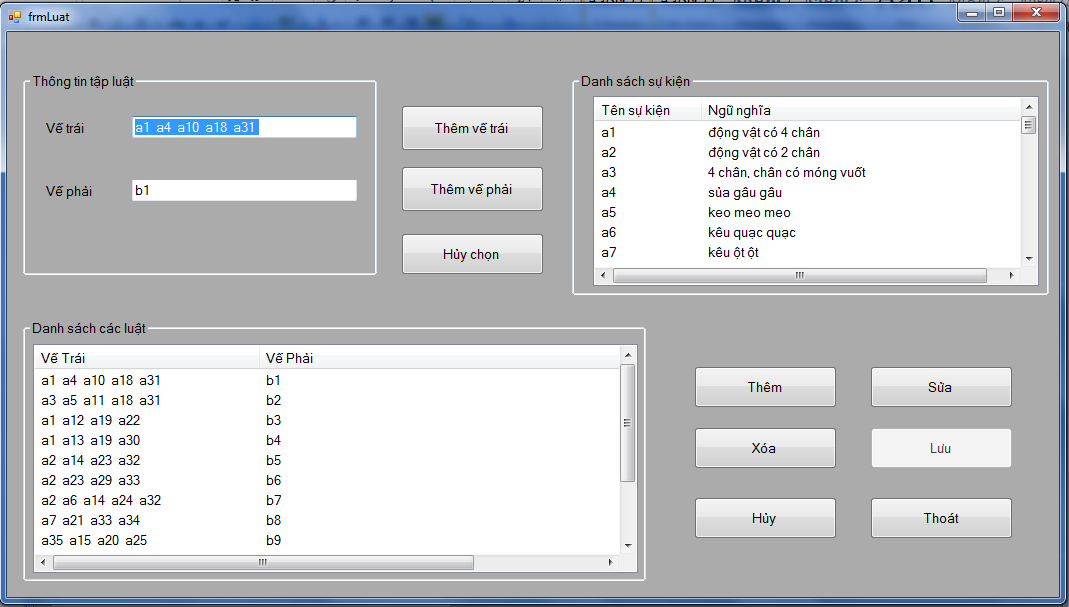
Trong chương trinh chỉ sử dụng cơ sở dữ liệu nhỏ nên cơ sở tri thức được lưu ra 2 file dulieu.dat và dulieuluat.dat.chương trình được cài đặt theo thuật toán suy diễn tiến.

1. **Giao diện chương trình.**

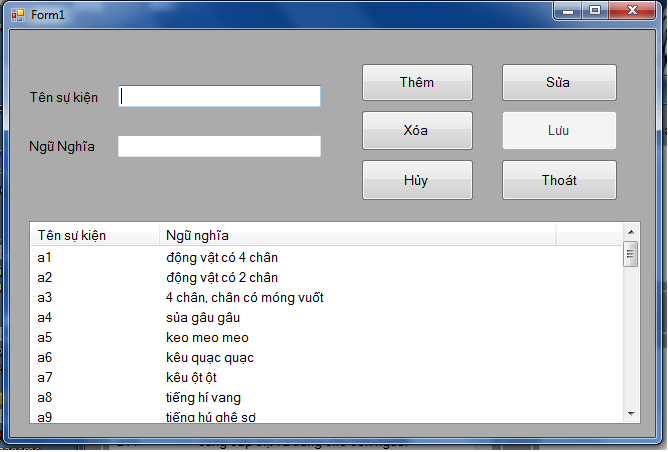
Giao diện chinh của chương trình.



Giao diện form luật



Giao diện form sự kiện.



1. **Code mã chương trình**.
2. **form suy diễn**.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.IO;

using System.Windows.Forms;

using System.Collections;

namespace HCG\_nhom\_5

{

public partial class frmSuydien : Form

{

public frmSuydien()

{

InitializeComponent();

}

string factsbase;//Gia thiet

// string ketluan; //ket luan

int soluat = 0;

int sosk = 0;

//string dtkt;

string[] tapkl;

string[] tapskt;

string[] tapsk;

// string giaithich;

string huyluat = "";//chua cac luat bi huy trong suy dien tien

string vet = ""; //Chứa các luật trong quá trình sd, dùng để inkq

ArrayList rulesbase = new ArrayList();//khai báo biến để chứa tập luật

ArrayList facts = new ArrayList();//Khai báo biến để chứa tập sự kiện

private void loadtapkl()

{

StreamReader f = new StreamReader("dulieuluat.dat");//mở file

int i = 0;

tapkl = new string[soluat];

while (f.Peek() != -1)//trong khi chưa kết thúc file

{

string s = f.ReadLine();//đọc 1 dòng từ file

string[] ss = s.Split('|');

// dt = new Luat(ss[0], ss[1]);

tapkl[i] = ss[1];

i++;

}

f.Close();

}

private void loadsk()

{

StreamReader f = new StreamReader("dulieu.dat");//mở file

int i = 0;

tapsk = new string[sosk];

while (f.Peek() != -1)//trong khi chưa kết thúc file

{

string s = f.ReadLine();//đọc 1 dòng từ file

string[] ss = s.Split('|');

// dt = new Luat(ss[0], ss[1]);

tapsk[i] = ss[1];

i++;

}

f.Close();

}

private void loadskTP()

{

StreamReader f = new StreamReader("dulieu.dat");//mở file

int i = 0;

tapsk = new string[sosk];

tapskt = new string[sosk];

while (f.Peek() != -1)//trong khi chưa kết thúc file

{

string s = f.ReadLine();//đọc 1 dòng từ file

string[] ss = s.Split('|');

// dt = new Luat(ss[0], ss[1]);

tapsk[i] = ss[1];

tapskt[i] = ss[0];

i++;

}

f.Close();

}

private int FileCount()

{

StreamReader f = new StreamReader("dulieuluat.dat");//mở file

int i = 0;

while (f.Peek() != -1)//trong khi chưa kết thúc file

{

string s = f.ReadLine();//đọc 1 dòng từ file

i++;

}

f.Close();

return i;

}

private int FileCountsk()

{

StreamReader f = new StreamReader("dulieu.dat");//mở file

int i = 0;

while (f.Peek() != -1)//trong khi chưa kết thúc file

{

string s = f.ReadLine();//đọc 1 dòng từ file

i++;

}

f.Close();

return i;

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

factsbase = txtLoi.Text;

suydientien();

}

private void listView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void frmSuydien\_Load(object sender, EventArgs e)

{

imageList1 = new ImageList();

imageList1.ImageSize = new Size(255, 255);

soluat = FileCount();

sosk = FileCountsk();

loadtapkl();

//mở file để đọc tập luật vào biến rulesbase

StreamReader f = new StreamReader("dulieu.dat");//mở file

Sukien sk;

while (f.Peek() != -1)//trong khi chưa kết thúc file

{

string s = f.ReadLine();//đọc 1 dòng từ file

string[] ss = s.Split('|');//tách vế trái (ss[0]) và vế phải (ss[1])

sk = new Sukien(ss[0], ss[1]);

facts.Add(sk);

//khởi tạo đối tượng Luật dt

// ss[0] = ss[0] + " ->"; //nối -> vào cuối vế trái

ListViewItem lv = new ListViewItem(ss);//khởi tạo dt listviewitem

listSukien.Items.Add(lv);// thêm 1 dòng vào listview1

//rulesbase.Add(dt);//thêm 1 luật vào rulesbase

}

f.Close();//đóng file

//mở file để đọc tập sự kiện vào biến facts

f = new StreamReader("dulieuluat.dat");//mở file

Luat dt;//khai báo dt là đối tượng của lớp Luat

while (f.Peek() != -1)

{

string s = f.ReadLine();

string[] ss = s.Split('|');

dt = new Luat(ss[0], ss[1]);

rulesbase.Add(dt);

// sk = new SuKien(ss[0], ss[1]);

// facts.Add(sk);

}

f.Close();

}

public void TimLuat(ArrayList tapluat, string tg, ref ArrayList sat)

{

for (int i = 0; i < tapluat.Count; i++)

{

//khai báo dt là đối tượng Luật và lấy luật thứ i về gán cho dt

Luat dt = (Luat)tapluat[i];

string[] ss = dt.Left.Split(' ');//tách vế trái thành các sự kiện

int j = 0;

while (!huyluat.Contains(i.ToString()) && //luật chưa bị xóa

!sat.Contains(i) && //tập sat chưa chứa luật i

j < ss.Length && // chưa duyệt hết sự kiện ở vế trái

tg.IndexOf(ss[j]) != -1)//trung gian có chứa sự kiện ss[j]

j++;

if (j == ss.Length) sat.Add(i);//vế trái thuộc tg nên thêm luật i vào sat

}

}

public struct CauHinh

{

public string sk;

public int cs;

}

private void suydientien()

{

ArrayList sat = new ArrayList();//khai báo tập sat để chứa các luật

loadskTP();

//gán biến tg bằng giả thiết factsbase

foreach (string a in tapkl)

{

string tg = factsbase;

vet = ""; //gán vết bằng rỗng

txtKetqua.Text = "";

huyluat = "";//khởi tạo hủy luật bằng rỗng

TimLuat(rulesbase, tg, ref sat);//tìm luật có vt thuộc tg đưa vào sat

while (sat.Count > 0)//trong khi tập sat khác rỗng

{

int cs = (int)sat[0];//lấy trong tập sat 1 chỉ số luật và gán cho cs

Luat r = (Luat)rulesbase[cs];//lấy luật ứng với cs và gán cho r

sat.RemoveAt(0);//xóa luật ở vị trí đầu tiên trong sat

tg = tg + " " + r.Right;//thêm vế phải của luật r vào tg

huyluat = huyluat + " " + cs.ToString();//hủy luật r

vet = vet + cs.ToString() + " ";//thêm cs luật r vào vết

if (tg.Contains(a)) //nếu kết luận thuộc tg thì inkq

{

txtKetqua.Text = "Thanh cong\r\n";

vet = vet.Trim();

string[] ss = vet.Split(' ');//tách vet thành các cs luật

for (int i = 0; i < ss.Length; i++)

{

int vt = int.Parse(ss[i]);//chuyển chuỗi ss[i] thành số

Luat l = (Luat)rulesbase[vt];//lấy luật tại vt về gán cho l

string s = string.Format("{0}->{1}\r\n", l.Left, l.Right);//định dạng để in luật

txtKetqua.Text = txtKetqua.Text + s;//thêm s vào textbox3

}

foreach (string tempkl in tapskt)

{

if (factsbase.Contains(tempkl))

{

int ii = Array.IndexOf(tapskt, tempkl);

string kl = tapsk[ii];

txtKetqua.Text = txtKetqua.Text + tempkl + "->" + kl + "\r\n";

}

}

txtKetqua.Text = txtKetqua.Text + "--------------------\r\n";

int iii = Array.IndexOf(tapskt, a);

string kl1 = tapsk[iii];

txtKetqua.Text = txtKetqua.Text + "Chẩn đoán: " + kl1 + "\r\n";

string images="images/"+kl1+".jpg";

pictureBox1.Image = Image.FromFile(images);

return; //kết thúc suy diễn

}

TimLuat(rulesbase, tg, ref sat);//tìm luật lại

}//while

txtKetqua.Text = " Không trùng trong tập luật !";

}

}

private void button2\_Click(object sender, EventArgs e)

{

}

private void listSukien\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void button2\_Click\_1(object sender, EventArgs e)

{

frmsukien fr = new frmsukien();

fr.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

frmLuat fr1 = new frmLuat();

fr1.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

for (int i = 0; i < listSukien.SelectedIndices.Count; i++)

txtLoi.Text = txtLoi.Text + " " + listSukien.Items[listSukien.SelectedIndices[i]].Text + " ";

}

private void button6\_Click(object sender, EventArgs e)

{

txtLoi.Text = "";

}

}

}

1. **form luật.**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace HCG\_nhom\_5

{

public partial class frmLuat : Form

{

int vt;

bool cothem;

public frmLuat()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button9\_Click(object sender, EventArgs e)

{

this.Close();

}

public void list\_text(int vt)

{

txtVT.Text = listLuat.Items[vt].SubItems[0].Text;

txtVP.Text = listLuat.Items[vt].SubItems[1].Text;

}

public void text\_list(int vt)

{

listLuat.Items[vt].SubItems[0].Text = txtVT.Text;

listLuat.Items[vt].SubItems[1].Text = txtVP.Text;

}

private void frmLuat\_Load(object sender, EventArgs e)

{

StreamReader f = new StreamReader("dulieuluat.dat");

while (f.Peek() != -1)

{

vt = vt + 1;

Luat dt = new Luat();

string s = f.ReadLine();

string[] ss = s.Split('|');

ListViewItem lv = new ListViewItem(ss);

listLuat.Items.Add(lv);

}

f.Close();

f = new StreamReader("dulieu.dat");//mo file

while (f.Peek() != -1)//trong khi chua den cuoi file

{

vt = vt + 1;

//SuKien dt = new SuKien();

string s = f.ReadLine();//doc 1 dong tu file roi gan cho s

string[] ss = s.Split('|');//cat s thanh 2 phan:ten va NN

//khai bao va khoi tao doi tuong listviewitem

ListViewItem lv = new ListViewItem(ss);

listSukien.Items.Add(lv);

}

f.Close();

if (vt != -1)

{

txtVT.Text = listLuat.Items[0].SubItems[0].Text;

txtVP.Text = listLuat.Items[0].SubItems[1].Text;

vt = 0;

}

}

private void btThem\_Click(object sender, EventArgs e)

{

txtVT.Text = "";

txtVP.Text = "";

txtVT.Focus();

cothem = true;

btLuu.Enabled = true;

btThem.Enabled = false;

}

public void TextRong()

{

txtVT.Text = "";

txtVP.Text = "";

}

public void LuuFile()

{

StreamWriter f = new StreamWriter("dulieuluat.dat", false);

for (int i = 0; i <= listLuat.Items.Count - 1;i++ )

{

string s = string.Format("{0}|{1}", listLuat.Items[i].SubItems[0].Text, listLuat.Items[i].SubItems[1].Text);

f.WriteLine(s);

}

MessageBox.Show(" Lưu thành công ! ", "ok", MessageBoxButtons.OK, MessageBoxIcon.Information);

f.Close();

}

private void btLuu\_Click(object sender, EventArgs e)

{

if (cothem)

{

string[] s = new string[2];

s[0] = txtVT.Text;

s[1] = txtVP.Text;

ListViewItem lv = new ListViewItem(s);

listLuat.Items.Add(lv);

}

else

{

text\_list(vt);

}

LuuFile();

btThem.Enabled = true;

btLuu.Enabled = false;

}

private void btHuy\_Click(object sender, EventArgs e)

{

if (vt != -1)

list\_text(vt);

else

TextRong();

btThem.Enabled = true;

btLuu.Enabled = false;

btSua.Enabled = true;

}

private void btXoa\_Click(object sender, EventArgs e)

{

listLuat.Items.RemoveAt(vt);

if (listLuat.Items.Count == 0)

{

TextRong();

vt = -1;

}

else

{

if (vt != 0)

vt = vt - 1;

list\_text(vt);

}

StreamWriter f = new StreamWriter("dulieuluat.dat", false);

for (int i = 0; i <= listLuat.Items.Count - 1; i++)

{

string s = string.Format("{0}|{1}", listLuat.Items[i].SubItems[0].Text, listLuat.Items[i].SubItems[1].Text);

f.WriteLine(s);

}

// MessageBox.Show(" Lưu thành công ! ", "ok", MessageBoxButtons.OK, MessageBoxIcon.Information);

f.Close();

}

private void btSua\_Click(object sender, EventArgs e)

{

txtVT.Focus();

cothem = false;

btLuu.Enabled = true;

btSua.Enabled = false;

}

private void listLuat\_ItemSelectionChanged(object sender, ListViewItemSelectionChangedEventArgs e)

{

txtVT.Text = e.Item.SubItems[0].Text;

txtVP.Text = e.Item.SubItems[1].Text;

vt = e.ItemIndex;

}

private void btThemVT\_Click(object sender, EventArgs e)

{

for (int i = 0; i < listSukien.SelectedIndices.Count; i++)

txtVT.Text = txtVT.Text + " " + listSukien.Items[listSukien.SelectedIndices[i]].Text + " ";

}

private void btThemVP\_Click(object sender, EventArgs e)

{

txtVP.Text = listSukien.Items[listSukien.SelectedIndices[0]].Text;

}

private void btHuyChon\_Click(object sender, EventArgs e)

{

listSukien.CheckBoxes = false;

txtVP.Text = "";

txtVT.Text = "";

btThem.Enabled = true;

btLuu.Enabled = false;

}

private void listSukien\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

}

1. **form sự kiện**.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.IO;

using System.Windows.Forms;

namespace HCG\_nhom\_5

{

public partial class frmsukien : Form

{

public frmsukien()

{

InitializeComponent();

}

int vt;

bool cothem;

private void button6\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

txtTenSK.Text = "";

txtNgunghia.Text = "";

txtTenSK.Focus();

btThem.Enabled = false;

btLuu.Enabled = true;

cothem = true;

}

public void list\_text(int vt)

{

txtTenSK.Text = listView1.Items[vt].SubItems[0].Text;

txtNgunghia.Text = listView1.Items[vt].SubItems[1].Text;

}

public void text\_list(int vt)

{

listView1.Items[vt].SubItems[0].Text = txtTenSK.Text;

listView1.Items[vt].SubItems[1].Text = txtNgunghia.Text;

}

private void btLuu\_Click(object sender, EventArgs e)

{

if (cothem)

{

string[] s = new string[2];

s[0] = txtTenSK.Text;

s[1] = txtNgunghia.Text;

ListViewItem lv = new ListViewItem(s);

listView1.Items.Add(lv);

}

else

{

text\_list(vt);

}

LuuFile();

btSua.Enabled = true;

btLuu.Enabled = false;

btThem.Enabled = true;

}

private void btSua\_Click(object sender, EventArgs e)

{

txtTenSK.Focus();

cothem = false;

btLuu.Enabled = true;

btSua.Enabled = false;

btThem.Enabled = false;

}

frmSuydien f = new frmSuydien();

private void Form1\_Load(object sender, EventArgs e)

{

StreamReader f = new StreamReader("dulieu.dat");

while (f.Peek() != -1)

{

vt = vt + 1;

string s = f.ReadLine();

string[] ss = s.Split('|');

ListViewItem lv = new ListViewItem(ss);

listView1.Items.Add(lv);

}

f.Close();

if (vt != -1)

{

txtTenSK.Text = listView1.Items[0].SubItems[0].Text;

txtNgunghia.Text = listView1.Items[0].SubItems[1].Text;

vt = 0;

}

}

private void LuuFile()

{

StreamWriter f = new StreamWriter("dulieu.dat", false, Encoding.Unicode);

for (int i = 0; i <= listView1.Items.Count - 1; i++)

{

string s = string.Format("{0}|{1}",

listView1.Items[i].SubItems[0].Text,

listView1.Items[i].SubItems[1].Text);

f.WriteLine(s);

}

MessageBox.Show(" Lưu thành công ! ", "ok", MessageBoxButtons.OK, MessageBoxIcon.Information);

f.Close();

}

private void btLuuFile\_Click(object sender, EventArgs e)

{

}

public void text\_rong()

{

txtTenSK.Text = "";

txtNgunghia.Text = "";

}

private void btXoa\_Click(object sender, EventArgs e)

{

listView1.Items.RemoveAt(vt);

if (listView1.Items.Count == 0)

{

text\_rong();

vt = -1;

}

else

{

if (vt != 0)

vt = vt - 1;

list\_text(vt);

}

}

private void btHuy\_Click(object sender, EventArgs e)

{

if (vt != 1)

list\_text(vt);

else

text\_rong();

}

private void listView1\_ItemSelectionChanged(object sender, ListViewItemSelectionChangedEventArgs e)

{

txtTenSK.Text = e.Item.SubItems[0].Text;

txtNgunghia.Text = e.Item.SubItems[1].Text;

vt = e.ItemIndex;

}

private void btLuuFile\_Click\_1(object sender, EventArgs e)

{

}

}

}

1. Class luat

using System;

using System.Collections.Generic;

using System.Text;

namespace HCG\_nhom\_5

{

class Luat

{

string left, right;

public Luat()

{ left = right = ""; }

public Luat(string vt, string vp)

{

left = vt;

right = vp;

}

public string Left

{

get

{

return left;

}

set

{

left = value;

}

}

public string Right

{

get

{

return right;

}

set

{

right = value;

}

}

}

}

1. Class su kien

using System;

using System.Collections.Generic;

using System.Text;

using System.IO;

namespace HCG\_nhom\_5

{

class Sukien

{

string ten, ngunghia;

public Sukien()

{

ten = ngunghia = "";

}

public Sukien(string ten, string ngunghia)

{

this.ten = ten;

this.ngunghia = ngunghia;

}

public string Ten

{

get

{

return ten;

}

set

{

ten = value;

}

}

public string Ngunghia

{

get

{

return ngunghia;

}

set

{

ngunghia = value;

}

}

}

}

**Tài liệu tham khảo**

<http://www.wattpad.com/8953802-%C4%91%E1%BA%B7c-tr%C6%B0ng-v%C3%A0-%C6%B0u-%C4%91i%E1%BB%83m-c%E1%BB%A7a-h%E1%BB%87-chuy%C3%AAn-gia-suy-di%E1%BB%85n#.UhdjzdLTquI>

<http://luanvan.net.vn/luan-van/bai-tap-lon-he-chuyen-gia-28267/>

<http://123doc.vn/document/16792-thuat-toan-giai-bai-toan-suy-dien-tien.htm>