Giới thiệu SWI-Prolog

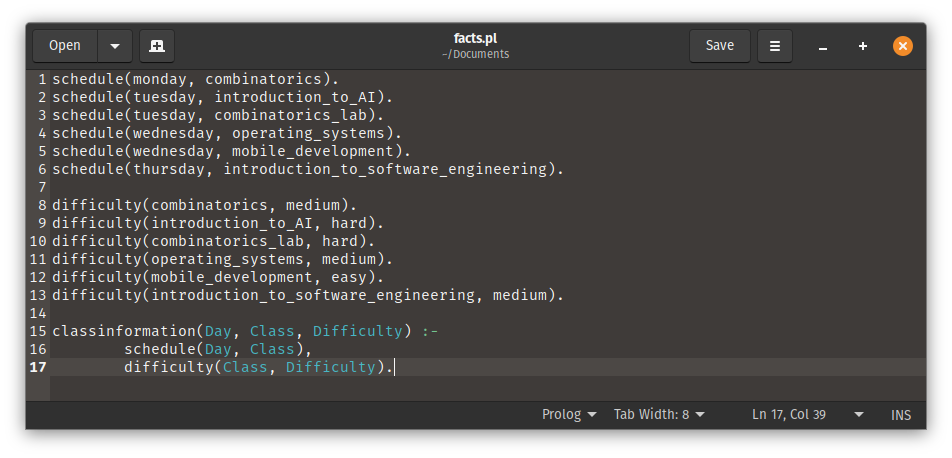
Hiện nay, có rất nhiều môi trường để lập trình Prolog như B-Prolog, GNU Prolog, Poplog Prolog, Quintus Prolog, ... Trong đó, SWI-Prolog cung cấp một môi trường đa dạng cho ngôn ngữ lập trình Prolog. Tên gọi SWI được đặt theo “Sociaal-Wetenschappelijke Informatica” (Tiếng Anh: “Social Science Informatics”) là tên trước đây của nhóm Human-Computer Studies ở Đại học Amsterdam nơi mà tác giả chính là Jan Wielemaker đã làm việc. SWI-Prolog ra đời năm 1987 và vẫn đang được phát triển tới phiên bản 8.3.13 (ngày 16 tháng 11 năm 2020). SWI-Prolog được sử dụng rộng rãi trong nghiên cứu, giáo dục, và cả trong các ứng dụng thương mại. SWI-Prolog sử dụng Giấy phép FreeBSD (BSD đơn giản).

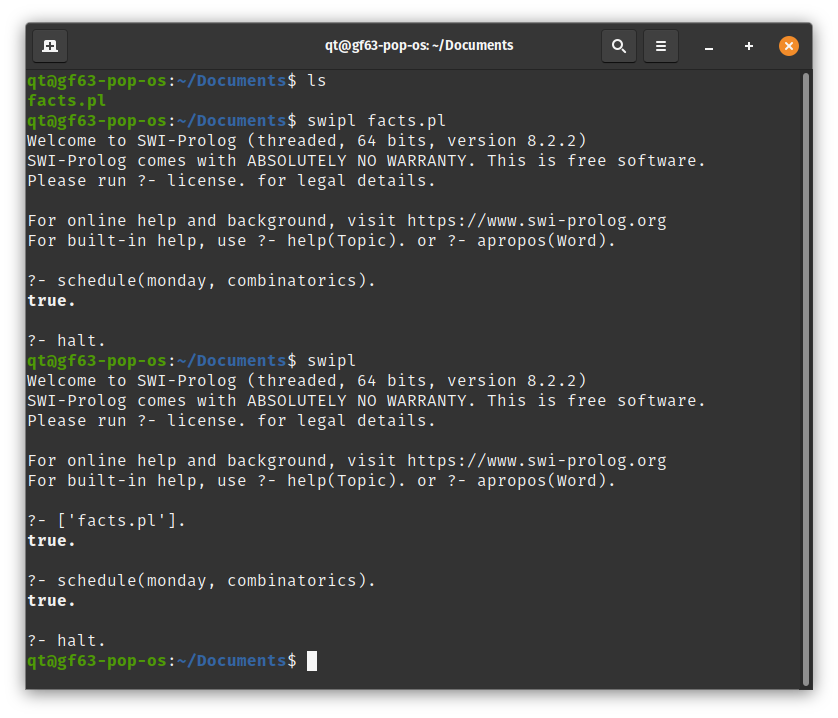
SWI-Prolog có thể chạy được trên các hệ điều hành Windows, macOS, và các nền tảng Unix/Linux. SWI-Prolog hỗ trợ giao diện đồ họa bằng XPCE, có khả năng thực thi độc lập, và sử dụng nhưng ngôn ngữ nhúng bằng các interface cho các ngôn ngữ khác như C++, Java, C#, Python, …

Sử dụng SWI-Prolog

Mỗi chương trình Prolog là một cơ sở tri thức (knowledge base, KB) bao gồm các mệnh đề. Chương trình được lưu trong tập tin văn bản đuôi của tên tập tin phổ biết là “.pl”. Để sử dụng tập tin chứa cơ sở tri thức, ta khởi chạy SWI-Prolog bằng lệnh swipl <địa chỉ tập tin> hoặc khởi chạy SWI-Prolog bằng lệnh swipl và gõ lệnh ['<địa chỉ tập tin>']. Mỗi lệnh trong SWI-Prolog phải được kết thúc bằng một dấu chấm. Để kết thúc SWI-Prolog, ta sử dụng lệnh halt.

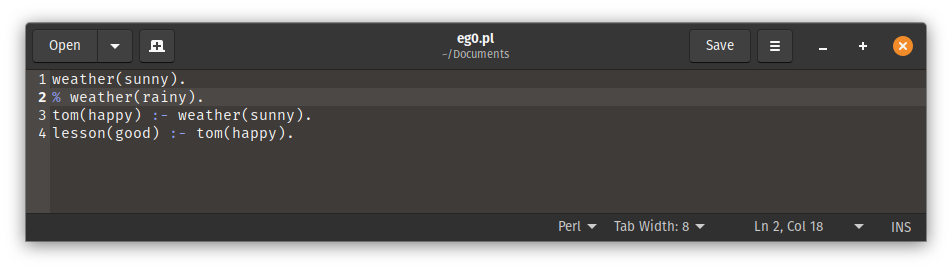
Ví dụ trong tập tin facts.pl có nội dung như sau:

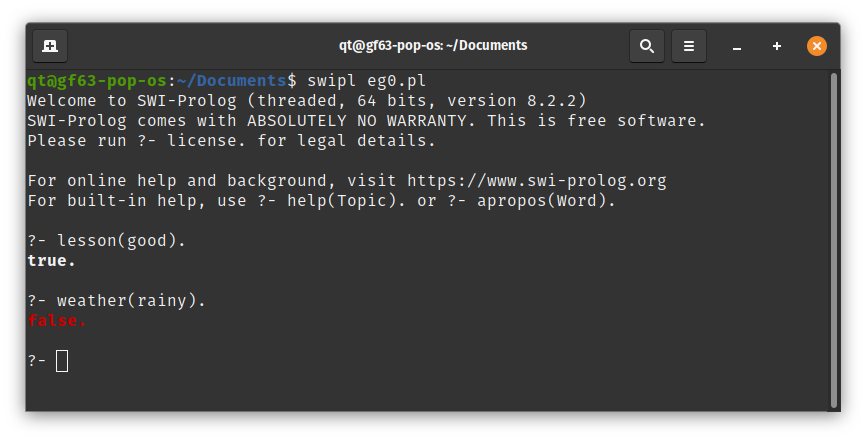
Ta sử dụng tập tin facts.pl bằng 2 cách đã trình bày ở trên.



Các ví dụ minh họa

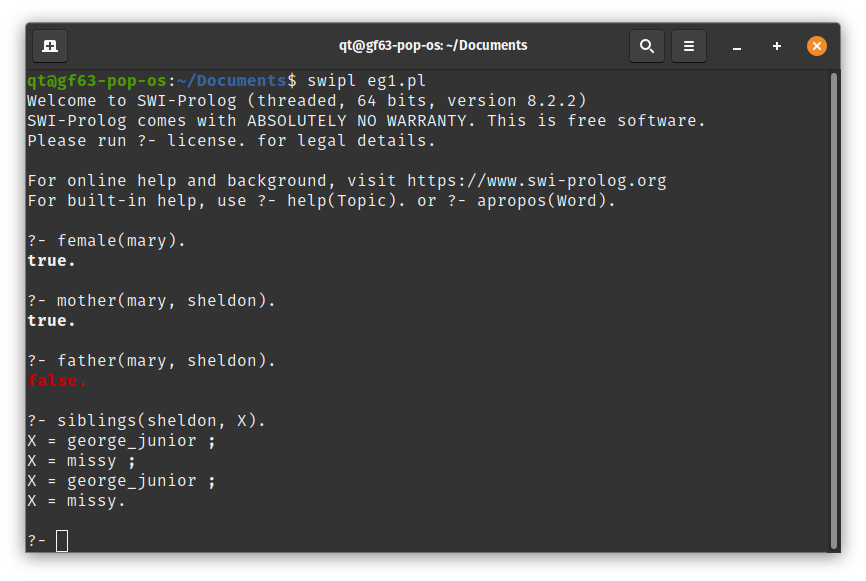
Ví dụ 1

Cho nội dung cơ sở tri thức như hình. Ta sẽ thử sử dụng cơ sở tri thức này trong SWI-Prolog.

Ta thấy dòng thứ 2 trong KB bắt đầu bằng “%”, đây là dòng comment nên sẽ không được đọc. Kết quả lesson(good) bằng true thỏa mã tam đoạn luận rằng nếu trời nắng thì tom vui vẻ, nếu tom vui vẻ thì bài học sẽ tốt.

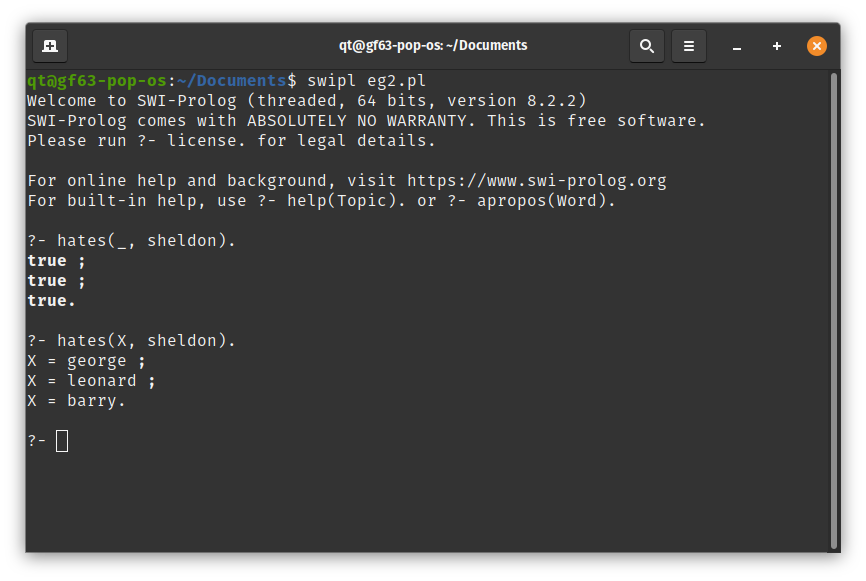
Ví dụ 2

Cho nội dung cơ sở tri thức như hình.

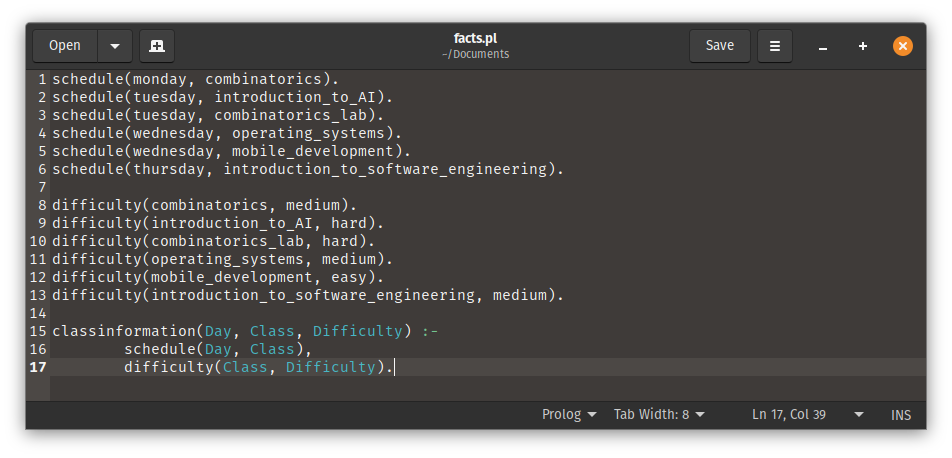
Ở câu lệnh đầu tiên, ta chỉ kiểm tra sự thật (fact) trong KB. Ở câu lệnh thứ 2 và 3, ta kiểm tra mary có phải là mẹ hay bố của sheldon hay không. Theo như luật đã định nghĩa trong KB, thì điều này là đúng. Ở câu lệnh cuối cùng, ta cần tìm các giá trị của X sao cho X là anh chị em của sheldon. Theo như luật đã định nghĩa thì X là anh chị em với Y nếu X khác Y (cấu trúc dif có sẵn trong SWI-Prolog) và tồn tại P sao cho P là bố mẹ của X và P là bố mẹ của Y. Ta tìm được 4 giá trị với 2 giá trị lặp 2 lần do có george\_senior và mary cùng là P thỏa mãn cho giá trị tìm được là george\_junior và missy. Sau mỗi dòng tìm kiếm trên SWI-Prolog, ta nhấn phím ‘;’ để tiếp tục tìm kiếm hoặc ‘.’ để kết thúc tìm kiếm.

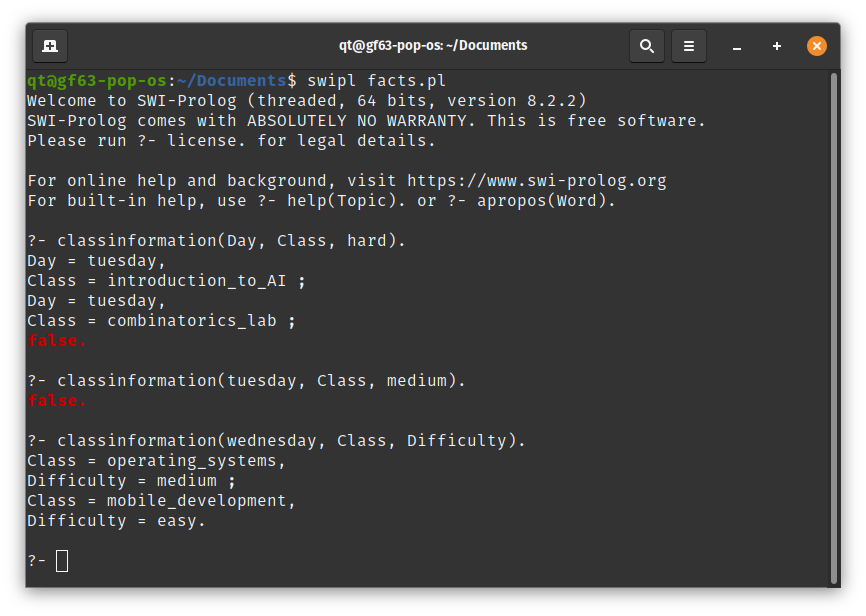
Ví dụ 3

Cho KB như hình.

Ở đây, ta sử dụng biến ẩn danh (‘\_’) để tìm những người ghét sheldon nhưng không hiện những người này ra. Ta thấy tìm được 3 giá trị true tức là có 3 người ghét sheldon. Sau đó, ta sử dụng lại biến bình thường để kiểm tra lại thì thấy tìm được đúng 3 người là george, leonard, và barry.

Ví dụ 4

Sử dụng lại KB facts.pl đã sử dụng ở trên.

Luật classinformation(Day, Class, Difficulty) nếu tồn tại schedule(Day, Class) và difficulty(Class, Difficulty). Ở ví dụ này, ta thử tìm kiếm những ngày và lớp học nào có độ khó hard. Ta tìm được 2 lớp ở ngày thứ ba thỏa điều kiện này. Ta thử tìm những lớp ngày thứ ba có độ khó medium thì kết quả trả về là false tức là không tìm được.

Ví dụ 5

Giải phép toán TWO + TWO = FOUR. Để có thể giải bài này, ta xây dựng KB như sau.

Ở đây, phép “=:=” là phép so sánh bằng giá trị 2 biểu thức. Khi chạy trên SWI-Prolog, ta được một số giá trị như sau:

