

CÁC HỆ THỐNG THÔNG MINH LAI VÀ CÁC AGENT THÔNG MINH*

Học Viện Công Nghệ Bưu Chính Viễn Thông

This work is produced by Thư viện Học liệu Mở Việt Nam and licensed under the
Creative Commons Attribution License †

Tóm tắt nội dung

Mục đích của chúng ta trong cuốn sách này là thiết kế các agent sao cho chúng làm việc tốt trong môi trường của chúng. Trước tiên, chúng ta sẽ làm rõ ràng hơn một chút về cái ý “việc tốt” vừa nói trên. Sau đó, chúng ta sẽ nói về các cách khác nhau thiết kế các agent hoàn thiện (đó chính là câu trả lời cho dấu ?). Chúng ta thảo luận một vài nguyên lý chung được sử dụng thiết kế các agent trình bày trong cuốn sách này. Quan trọng nhất đó là nguyên lý để thiết các agent phải hiểu biết các sự việc. Cuối cùng, chúng ta chỉ ra sự gắn liền giữa một agent và một môi trường như thế nào, và chỉ ra một vài loại môi trường.

1 CÁC HỆ THỐNG THÔNG MINH LAI

Mỗi một hệ thống thông minh có những ưu điểm và hạn chế riêng của nó. Chúng ta có thể nhìn nhận các hệ thống kết hợp nhiều phương pháp trí tuệ hiện đại theo các nhóm như sau

1.1 Hệ thống Nơ ron -Mô

Các hệ thống nơ ron mà là sự kết hợp của hai phương pháp mờ và nơ ron. Trong mạng nơ ron: một số lớp được chọn làm chức năng của hàm thuộc. Chức năng đó kết hợp khả năng phân lớp đối tượng của hai công cụ. Sản phẩm được xay dựng có thể là phần cứng và phần mềm

Việc kết hợp (hay lai) này xuất phát từ ưu điểm của mạng nơ ron là thông minh hoá trên các phần tử, còn logic mờ làm nhiệm vụ khái quát, suy diễn.

1.2 Hệ thống Nơ ron - Giải thuật di truyền

Như ta biết mạng nơ ron có khả năng cao trong xử lý song song, phi tuyến và có khả năng giải nhiều bài toán tối ưu cục bộ. Để giải các bài toán tối ưu toàn cục, người ta thường dùng giải thuật di truyền để tối ưu toàn cục sau đó dùng mạng nơ ron nhân tạo để tối ưu mịn (cục bộ). Giải pháp này mang lại hiệu quả cao, hoàn thiện hơn. Tuy nhiên, giá phải trả là tăng độ phức tạp

*Version 1.1: Dec 21, 2010 9:43 pm GMT+7

†<http://creativecommons.org/licenses/by/3.0/>

1.3 Các hệ thống lai khác

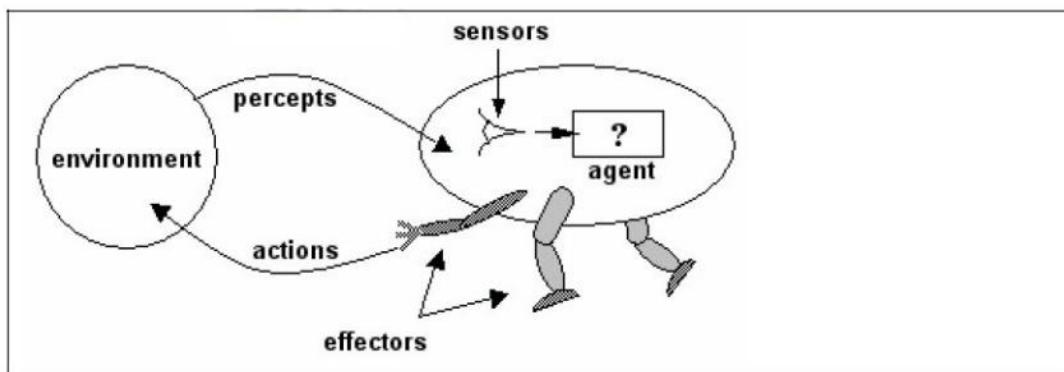
- Một số phương pháp khác là kết hợp cả mạng nơ ron giải thuật di truyền và logic mờ. Về mặt khoa học, vấn đề đó mang tính khách quan. Trong thực tế độ phức tạp ngày càng cao.
- Kết hợp phương pháp thống kê hiện đại (như phương pháp Bayes trong suy diễn mờ)

2 CÁC AGENT THÔNG MINH

Phần này thảo luận “agent thông minh” là gì, nó có quan hệ với môi trường như thế nào, nó được đánh giá như thế nào, và chúng ta có thể xây dựng nó như thế nào?

2.1 Giới thiệu

Một “agent” là bất kỳ vật gì đó mà nó có thể nhận biết (perceiving) môi trường (environment) quanh nó thông qua các cảm biến (sensor) và tác động lại môi trường thông qua các bộ phận phản ứng lại kích thích (effector). Một cơ thể người có đôi mắt, đôi tai và các bộ phận khác là các cảm biến; đôi bàn tay, đôi chân và các phần cơ thể khác là các cơ quan phản ứng lại kích thích. Một robot camera và tia hồng ngoại nhìn thấy được là các sensor; các motor khác nhau là các các bộ phận phản ứng lại kích thích. Một phần mềm đã mã hoá các xâu bit xem như là các đối tượng tri giác và các hành động của nó.



Hình 1: Các agent tương tác với môi trường thông qua sensor và các effector

Mục đích của chúng ta trong cuốn sách này là thiết kế các agent sao cho chúng làm việc tốt trong môi trường của chúng. Trước tiên, chúng ta sẽ làm rõ ràng hơn một chút về cái ý “việc tốt” vừa nói trên. Sau đó, chúng ta sẽ nói về các cách khác nhau thiết kế các agent hoàn thiện (đó chính là câu trả lời cho dấu ? trong hình 1). Chúng ta thảo luận một vài nguyên lý chung được sử dụng thiết kế các agent trình bày trong cuốn sách này. Quan trọng nhất đó là nguyên lý để thiết các agent phải hiểu biết các sự việc. Cuối cùng, chúng ta chỉ ra sự gắn liền giữa một agent và một môi trường như thế nào, và chỉ ra một vài loại môi trường.

2.2 Hoạt động của các Agent

Một agent có lý trí (rational agent) là agent làm các việc theo lẽ phải. Hiển nhiên, điều đó tốt hơn là làm việc sai trái, vậy điều đó có ý nghĩa là gì. Như là một sự đánh giá không được chính xác lắm, chúng ta nói rằng hành động đúng là lý do agent thành công hơn. Điều đó dẫn đến vấn đề chúng ta phải đánh giá sự thành công của agent như thế nào (how) và khi nào (when).

Tiêu chuẩn đánh giá sự thực hiện (Performance Measure)

Chúng ta sử dụng thuật ngữ tiêu chuẩn đánh giá sự thực hiện cho tiêu chuẩn how – tiêu chuẩn xác định xem như thế nào là một agent là thành công. Hiển nhiên không có một tiêu chuẩn đánh giá cố định nào thích hợp cho tất cả các agent. Chúng ta có thể hỏi agent theo quan điểm chủ quan nó hạnh phúc như thế nào với sự thực hiện của chính nó, nhưng một vài agent không thể trả lời, và một vài agent khác lừa dối chúng. (Con người ai cũng biết câu chuyện “con cáo và chùm nho xanh” - các con cáo nói rằng chúng không thực sự muốn điều đó sau khi họ không thành công trong việc dành lấy nó). Vì vậy, chúng ta đòi hỏi một tiêu chuẩn đánh giá sự thực hiện khách quan áp đặt bởi một số chuyên gia. Nói cách khác, chúng ta như là người quan sát thiết lập một chuẩn để cho nó thành công trong một môi trường và sử dụng nó để đánh giá sự thực hiện của các agent.

Cho một ví dụ, xem xét trường hợp một agent được hỗ trợ để hút bụi bẩn sàn nhà. Một tiêu chuẩn đánh giá hiệu suất đáng tin cậy có thể là số lượng bụi bẩn làm sạch trong một ca làm việc 8 giờ. Một đánh giá hiệu suất phức tạp hơn có thể là số lượng điện tiêu thụ và mức độ tiếng ồn phát ra. Đánh giá hiệu suất thứ ba có thể đưa ra điểm số cao hơn cho một agent không chỉ làm sạch sàn nhà một cách yên tĩnh và hiệu quả mà còn nhận ra thời gian nghỉ cuối tuần.¹

Tiêu chuẩn when để đánh giá sự thực hiện cũng quan trọng. Nếu chúng ta đánh giá xem có bao nhiêu bụi bẩn agent làm sạch trong giờ đầu tiên trong ngày, chúng ta có thể thỏa mãn với các agent khởi động nhanh (thậm chí chúng làm ít hoặc không làm sau thời điểm đó), và không hài lòng với những agent làm việc hợp lý. Vì vậy, chúng ta muốn đánh giá sự thực hiện trên toàn thời gian hoạt động, đó là một ca làm việc 8 giờ hoặc thời gian tồn tại của agent.

Sự thông thái (Omniscience)

Chúng ta cần lưu ý phân biệt giữa sự hợp lý (rationality) và sự thông thái (omniscience). Một agent thông thái biết kết quả của các hành vi của nó ngay hiện thời và có thể hành động sao cho phù hợp, nhưng thông thái không có trong thực tế. Xem xét ví dụ: Một ngày tôi đang đi dọc theo đường Champs Elysées và tôi nhìn thấy một người bạn cũ bên kia đường. Không có xe cộ nào xung quanh và tôi không bận việc khác, vì vậy theo lẽ thường, tôi bắt đầu băng qua đường. Trong lúc đó, ở độ cao 33.000 feet một cánh cửa khoang hàng hoá rơi xuống từ một máy bay dân dụng vừa bay qua², và trước khi tôi đến được bên kia đường tôi đã ngã sõng soài. Tôi băng qua đường là không hợp lý? Điều không muốn xảy ra là cáo phó của tôi có thể viết “Thằng ngốc cố gắng băng qua đường”. Đúng hơn, điều đó chỉ ra rằng, sự hợp lý được quan tâm với sự thành công mong muốn mang lại cái được nhận biết. Băng qua đường là hợp lý bởi vì đa phần những lần băng qua là thành công, và không có cách nào tôi có thể lường trước việc cánh cửa rơi. Lưu ý rằng một agent khác được trang bị radar phát hiện nhằm tránh các cánh cửa rơi xuống hoặc một khung thép đủ nặng sẽ “thành công” hơn, nhưng nó có thể không còn hợp lý.

Nói cách khác, chúng ta không thể đổ lỗi cho một agent vì không tính đến một số điều xảy ra mà nó không nhận thức được, hoặc vì không đưa ra một hành vi (ví dụ như tránh cánh cửa hành hoá) mà nó không có khả năng đưa ra. Nhưng việc giảm bớt yêu cầu của sự hoàn hảo không phải là vấn đề của việc hợp lý hoá agent. Nhận xét rằng, nếu chúng ta xác định một agent thông minh phải luôn luôn làm những cái thực sự chính xác, thì sẽ không thể thiết kế một agent đáp ứng đầy đủ yêu cầu đó - trừ khi chúng ta lợi dụng được sự làm của các tinh thể hình cầu.

Tóm lại, tính có lý trí đã được đưa ra dựa trên 4 nội dung sau: Tiêu chuẩn đánh giá sự thực hiện định rõ mức độ thành công.

Những gì mà agent nhận biết được từ xa (theo nghĩa thời gian). Chúng ta gọi tiền sử giác quan một cách đầy đủ này là “chuỗi kết quả tri giác” (percept sequence).

Những gì agent hiểu biết về môi trường.

Những hành động mà agent có thể thực hiện.

Điều đó dẫn tới một định nghĩa về một agent có lý trí lý tưởng: Với một “chuỗi kết quả tri giác” có thể, một agent có lý trí lý tưởng phải đưa ra hành động nào đó được mong đợi đạt tới cực đại theo tiêu chuẩn đánh giá sự thực hiện dựa trên dấu hiệu nhận biết cơ bản quy định bởi chuỗi kết quả tri giác và sự hiểu biết có sẵn nào đó mà agent có.

Chúng ta cần xem xét cẩn thận định nghĩa này. Thoạt nhìn, nó có thể xuất hiện khả năng một agent tự cho phép mình một số hoạt động rõ ràng kém thông minh. Ví dụ, nếu một agent không nhìn cả hai lối đi

trước ngã tư một con đường nào nhiệt, thì chuỗi kết quả tri giác của nó sẽ không chỉ cho nó biết rằng có một xe tải lớn tiến đến gần với tốc độ cao. Sự xác định đó đường như chỉ ra rằng nó có thể được chấp nhận băng qua đường. Trên thực tế, sự giải thích đó là sai trên cả hai phương diện. Thứ nhất, nó có thể không có lý trí khi băng qua đường: ngã tư đang nhìn là quá lớn. Thứ hai, một agent có lý trí lý tưởng sẽ có lựa chọn hành vi “nhìn” trước khi bước từng bước trên đường, bởi vì hành vi “nhìn” trợ giúp lớn nhất cho sự thực hiện mong muốn. Các hành động đang làm theo trình tự để thu được thông tin hữu ích là một phần quan trọng của “sự hợp lý” và được xem xét sâu hơn trong chương sau.

Khái niệm về một agent có ý nghĩa như là một công cụ để phân tích các hệ thống, và là sự mô tả các đặc tính không tuyệt đối để phân chia thế giới thành các agent và không phải agent. Xem xét một cái đồng hồ. Nó có thể được nghĩ chỉ là một đối tượng vô tri vô giác, hoặc nó có thể được xem như một agent đơn giản. Như một agent vì đa số đồng hồ luôn luôn thực hiện các hành động đúng: sự chuyển động của các kim (hoặc hiển thị các con số đối với đồng hồ điện tử) theo đúng quy tắc cấu tạo nêu. Các đồng hồ là loại agent thoái hoá trong đó chuỗi kết quả tri giác của chúng là rõ ràng; không có sự kiện nào xảy ra bên ngoài ảnh hưởng đến các hành động của đồng hồ.

May mắn, điều đó là không hoàn toàn đúng. Nếu cái đồng hồ và chủ nhân của nó đi trên con tàu từ California đến Australia, một thực tế là đồng hồ đã tự quay ngược trở lại 6 giờ. Chúng ta không làm đảo lộn các đồng hồ của chúng ta do chúng ta không làm cái việc đó; chúng ta thấy rõ ràng chúng đang hoạt động hợp lý, nguyên nhân là do chúng ta có cảm giác các thiết bị của các đồng hồ đã chạy nhanh³.

Phép ánh xạ lý tưởng từ các chuỗi kết quả tri giác thành các hành vi

Đôi khi chúng thấy rằng cách hành động của một agent chỉ dựa trên chuỗi kết quả tri giác để xác định thời điểm, trong khi đó chúng ta có thể mô tả agent riêng biệt nào đó bằng cách xây dựng bảng các hành vi nó thi hành đáp lại chuỗi kết quả tri giác có thể. (Với nhiều agent, bảng đó là một danh sách rất dài - trong thực tế sẽ là vô hạn, trừ khi chúng ta đặt một giới hạn chiều dài của chuỗi kết quả tri giác mà chúng ta muốn xem xét). Một danh sách như vậy được gọi là một phép ánh xạ từ chuỗi kết quả tri giác thành các hành vi. Về nguyên tắc, chúng ta có thể tìm ra một phép ánh xạ phù hợp để mô tả một agent bằng việc thử tất cả các khả năng chấp nhận được của chuỗi kết quả tri giác và ghi lại hành vi mà agent đáp lại. (Nếu agent sử dụng một vài phép ngẫu nhiên trong việc tính toán, thì chúng ta sẽ thử một vài chuỗi kết quả tri giác, mỗi chuỗi dăm ba lần để đưa ra hiểu biết đúng về cách hành động theo mức thông thường của agent). Và nếu sử dụng các phép ánh xạ mô tả các agent, thì các phép ánh xạ lý tưởng được sử dụng mô tả các agent lý tưởng. Theo đó, hành vi một agent phải đáp lại chuỗi kết quả tri giác nào đó sẽ cho ta một phác thảo về một agent lý tưởng.

Dĩ nhiên, điều đó không có nghĩa là chúng ta phải tạo ra một bảng rõ ràng cho mọi chuỗi kết quả tri giác có khả năng xảy ra. Điều đó cho phép xác định một đặc tả phép ánh xạ mà không phải liệt kê đầy đủ nó. Một agent đơn giản được cho là rất hay: Hàm căn bậc hai gần đúng của máy tính. Chuỗi kết quả tri giác cho agent này là một chuỗi các sự kiện bấm phím biểu diễn các con số trên bàn phím, hành vi là hiển thị con số trên màn hình hiển thị. Phép ánh xạ lý tưởng: kết quả tri giác là một số thực x, hành vi đúng là hiển thị một số thực z sao cho $z^2 \approx x$, lấy chính xác đến 15 chữ số. Sự đặc tả đó của phép ánh xạ không mang lại cho người thiết kế cấu trúc thực của bảng các căn bậc hai gần đúng. Và cũng không mang lại cho hàm căn bậc hai gần đúng sử dụng bảng để có cách hành động chính xác: Hình 2 chỉ ra một phần của phép ánh xạ lý tưởng và một chương trình đơn giản tính toán phép ánh xạ sử dụng phương pháp tính gần đúng Newton.

Ví dụ căn bậc hai gần đúng ở trên minh họa quan hệ giữa phép ánh xạ lý tưởng và một thiết kế agent lý tưởng cho rất nhiều nhiệm vụ là hữu hạn. Trong khi bảng có kích thước rất lớn, thì agent lại là một chương trình hay súc tích. Điều đó cho thấy có thể thiết kế các agent súc tích thực hiện phép ánh xạ lý tưởng vào các tình huống tổng quát hơn rất nhiều: các agent mà có thể giải quyết vô số các nhiệm vụ khác nhau trong vô số các môi trường khác nhau. Trước khi chúng ta thảo luận xem điều đó được thực hiện như thế nào, chúng ta cần xem xét thêm một yêu cầu nữa mà một agent thông minh cần thỏa mãn.

Kết quả tri giác x	Hành vi z
1.0	1.000000000000000
1.1	1.048808848170152
1.2	1.095445115010332
1.3	1.140175425099138
1.4	1.183215956619923
1.5	1.224744871391589
1.6	1.264911064067352
1.7	1.303840481040530
1.8	1.341640786499847
1.9	1.378404875209022

```

function SQRT(x)
Begin
    z ← 1.0 /* initial guess*/
    repeat until |z2 - x| < 10-15
        z ← z - (z2 - x)/(2z)
    end
    return z

```

Hình 2: Một phần của phép ánh xạ lý tưởng của bài toán căn bậc hai gần đúng (lấy xấp xỉ đến 15 chữ số), và một chương trình tương ứng để thực hiện phép ánh xạ lý tưởng.

Sự tự trị (Autonomy)

Có một điều nữa cần quan tâm trong việc định nghĩa một agent có lý trí lý tưởng: ở phần “Sự hiểu biết có sẵn - (built-in knowledge)”. Nếu các hành vi của agent được bố trí hoàn toàn trong “sự hiểu biết có sẵn”, thí dụ rằng nó không cần chú ý tới các kết quả giác của nó, thì chúng ta nói rằng agent thiếu sự tự trị. Cho ví dụ, nếu nhà sản xuất đồng hồ có thể thấy trước rằng chủ nhân của chiếc đồng hồ sẽ đến Australia vào một ngày nào đó, thì một cơ chế có thể được cài đặt sẵn để điều chỉnh các kim một cách tự động 6 giờ cho đúng. Dĩ nhiên điều đó có thể thực hiện thành công, nhưng sự thông minh dường như là việc của người thiết kế đồng hồ hơn là của chính chiếc đồng hồ.

Cách hành động của agent có thể được bố trí vào trong cả “kinh nghiệm của chính nó” và “sự hiểu biết có sẵn” trong khi xây dựng cấu trúc của agent, agent đó hoạt động trong một môi trường cụ thể. Một hệ thống là “tự trị - autonomous” tới một mức độ nhất định đó là hệ thống mà cách hành động được quyết định bởi chính kinh nghiệm của nó. Mặc dù nó có thể là quá khó khăn để đạt được sự tự trị hoàn toàn: khi agent có ít hoặc không có kinh nghiệm, nó phải hành động ngẫu nhiên trừ khi người thiết kế đưa ra một vài sự giúp đỡ. Vì vậy, khi mà sự tiến hoá cung cấp cho động vật với đầy đủ các phản xạ tự nhiên thì chúng có thể tồn tại đủ để tự học tập, nó có lý để cung cấp một agent thông minh nhân tạo với một vài sự hiểu biết ban đầu giống như năng lực để học tập.

Sự tự trị không chỉ phù hợp với trực giác của chúng ta mà nó còn là một ví dụ của thực tiễn cơ sở khoa học. Một agent hoạt động trên cơ sở của các giả định sẽ chỉ hoạt động thành công khi hiểu rõ các giả định, và như vậy thiếu đi sự linh hoạt. Giả sử, cho ví dụ con bọ hung. Sau khi tìm ra tổ và đẻ trứng, nó đem về các viên phân từ một đồng phân gần đó để bịt lối vào; nếu viên phân bị rơi trên đường đi, con bọ hung sẽ tiếp tục làm và bịt lối vào tổ với những viên phân khác theo kịch bản, mà không bao giờ để ý đến những viên phân nó đã đánh rơi. Sự tiến hoá đã gắn vào giả định của con bọ hung, và khi nó bị vi phạm, kết quả của hành vi không thành công (vì đã không nhặt lại các viên phân bị rơi). Một agent thông minh tự trị thực sự phải có khả năng hành động thành công trong các môi trường rộng lớn khác nhau đã cho đủ thời gian

thích nghi.

2.3 Cấu trúc của các agent thông minh

Cho đến giờ chúng ta đã nói đến các agent qua sự mô tả cách hành xử của chúng - hành vi được thực hiện sau khi đã đưa ra chuỗi kết quả tri giác. Bây giờ chúng ta sẽ cố gắng chờ đợi và thảo luận xem bên trong nó làm việc như thế nào. Công việc của AI (Artificial Intelligence - Trí tuệ nhân tạo) là thiết kế agent chương trình: một chức năng thực hiện agent ánh xạ từ các kết quả tri giác tới hành vi. Chúng ta giả sử chương trình này sẽ chạy trên vài loại thiết bị tính toán - gọi là kiểu kiến trúc. Hiển nhiên, chương trình chúng ta lựa chọn là chương trình mà kiểu kiến trúc chấp nhận và chạy. Kiểu kiến trúc có thể là một máy tính đơn giản, hoặc nó có thể bao gồm thiết bị cho mục đích đặc biệt dành cho các nhiệm vụ nào đó, như camera ghi lại các hình ảnh hoặc bộ lọc âm thanh đầu vào. Nó cũng có thể bao gồm phần mềm đưa ra sự phân biệt trình độ giữa máy tính và agent chương trình, vì vậy chúng ta có thể lập chương trình ở mức cao hơn. Nói chung, kiểu kiến trúc thu các kết quả tri giác từ các sensor sẵn sàng cho chương trình, chạy chương trình, và cho hoạt động của chương trình lựa chọn các bộ phận phản ứng lại kích thích được sinh ra. Mỗi quan hệ giữa các agent, các kiểu kiến trúc, và các chương trình có thể tóm tắt như sau:

agent = kiểu kiến trúc + chương trình

(agent = architecture + program)

Phản lớn trong cuốn sách này nói về việc thiết kế các chương trình agent.

Có thể là ngạc nhiên với một vài độc giả vì chúng ta đã bao gồm trong danh sách các kiểu agent một vài chương trình đường như hoạt động trong môi trường nhân tạo hoàn toàn được định nghĩa bởi bàn phím cho đầu vào và các ký tự đầu ra trên màn hình. “Không nghĩ ngờ gì” có thể nói “đó không phải là môi trường thực, vậy nó là gì?”. Trong thực tế, không có sự khác biệt giữa môi trường “thực” và môi trường “nhân tạo”, nhưng mối quan hệ giữa cách hành xử của agent, chuỗi kết quả tri giác được sinh ra bởi môi trường, và mục tiêu cần đạt được của agent được hỗ trợ lại phức tạp. Một vài môi trường “thực” là hết sức đơn giản. Cho ví dụ, robot được thiết kế để kiểm tra những phần mà chúng đi qua trên một băng truyền có thể dùng một số giả định đơn giản: các phần của băng truyền luôn luôn được chiếu sáng, các phần đó cùng thuộc một loại, và robot chỉ có hai hành vi - “chấp thuận” nếu phần băng truyền được chiếu sáng hoặc “dánh dấu” nếu phần băng truyền không được chiếu sáng. Trái lại, các agent phần mềm (hoặc các robot phần mềm) lại phong phú, phạm vi ứng dụng rất lớn. Hãy hình dung một hệ thống được thiết kế để mô phỏng lái máy bay 747. Sự mô phỏng có rất nhiều chi tiết, môi trường phức tạp, và tác nhân phần mềm phải chọn lựa từ rất nhiều hành động đa dạng trong thời gian thực. Hoặc hình dung một hệ thống được thiết kế để duyệt các nguồn tin tức trực tuyến và hiển thị các mục thú vị cho khách hàng. Để làm việc tốt, nó sẽ cần đến một vài ngôn ngữ tự nhiên có khả năng xử lý, cần phải biết được cái mà khách hàng cho là thú vị, và nó cần phải năng động trong việc thay đổi kế hoạch ví dụ khi việc kết nối vào một nguồn tin tức bị gián đoạn hoặc khi có một cái gì đó mới hơn trên đường truyền.

Kiểu Agent	Các tri giác	Các hành vi	Các mục tiêu	Môi trường
Hệ thống chuẩn đoán bệnh	các triệu chứng, các phát hiện, các câu trả lời của bệnh nhân	Các câu hỏi, các trắc nghiệm, cách điều trị	Sức khoẻ người bệnh, chi phí thấp nhất	Bệnh nhân, bệnh viện
Hệ thống phân tích hình ảnh vè tinh	Cường độ, màu sắc khác nhau của các điểm ảnh	Xuất ra một sự phân loại quanh cảnh nó quan sát	phân loại chính xác	Các hình ảnh từ vè tinh
Part-picking robot	cường độ khác nhau của các điểm ảnh	nhặt lên các bộ phận (của một loại máy móc) và đưa vào các cái thùng	Vị trí chính xác của các bộ phận trong các thùng	Băng tải với các bộ phận máy móc
Điều khiển nhà máy tinh chế dầu hoặc đường (Refinery controller)	Nhiệt độ, áp suất	Mở, đóng van; điều chỉnh nhiệt độ	sự tinh khiết, sản lượng và sự an toàn cao nhất	Nhà máy tinh chế
Chương trình dạy tiếng Anh	các từ loại	đưa ra các bài tập, các gợi ý, chỉ ra chỗ đúng	Kết quả kiểm tra sinh viên cao	Lớp sinh viên

Hình 3: Một số ví dụ về một số kiểu agent

Một số môi trường là không rõ ràng giữa “thực” và “nhân tạo”. Trong môi trường ALIVE, agent phần mềm được đưa đến các kết quả tri giác là hình ảnh camera kỹ thuật số của một phòng nơi có một người đi dạo trong đó. Agent này xử lý hình ảnh camera và lựa chọn một hành động. Môi trường cũng hiển thị hình ảnh camera trên một màn hình lớn, và thêm vào đó hình ảnh mà agent phần mềm biểu diễn thông qua đồ họa máy tính. Hình ảnh đó có thể là một con chó hoạt hình, nó được lập trình để chuyển động hướng đến một người (trừ trường hợp ông ta đuổi con chó đi) và vẫy chiếc chân trước hoặc nhảy lên một cách mừng rỡ khi người đó người đó ra hiệu.

Môi trường nhân tạo nổi tiếng nhất đó là Turing Test, trong đó chỉ ra rằng các agent “thực” và “nhân tạo” có quan hệ bình đẳng, nhưng môi trường lại có sự thách thức khá lớn là nó rất khó cho agent phần mềm làm việc như một con người.

Các agent chương trình

Trong toàn bộ cuốn sách này, chúng ta sẽ xây dựng các agent thông minh. Tất cả được xây dựng giống như bộ xương, chúng nhận các kết quả tri giác từ một môi trường và sinh ra các hành vi. Các phiên bản ban đầu của các agent chương trình có dạng rất đơn giản. Chúng sẽ sử dụng một vài cấu trúc dữ liệu nội tại mà nó được cập nhật khi các kết quả tri giác được đưa đến. Các cấu trúc dữ liệu đó được đem lại bởi

sự quyết định của agent - các thủ tục để sinh ra một sự lựa chọn hành vi, sau đó hành vi được chuyển qua kiểu kiến trúc để thực hiện.

Có hai điều cần chú ý về chương trình khung xương (chương trình vỏ) này. Thứ nhất, cho dù chúng ta định nghĩa agent như là một chức năng ánh xạ từ các chuỗi kết quả tri giác đến các hành vi, thì agent chương trình cũng chỉ nhận một kết quả tri giác đơn lẻ như là đầu vào. Nó để cho agent xây dựng dần chuỗi kết quả tri giác trong bộ nhớ, khi có yêu cầu. Trong một vài môi trường, nó có thể thực hiện hết sức thành công mà không có sự lưu trữ chuỗi kết quả tri giác, và trong những lĩnh vực phức tạp nó không thể lưu trữ chuỗi đầy đủ. Thứ hai, mục tiêu và thước đo hiệu suất không phải là nhiệm vụ của chương trình khung xương. Bởi vì, thước đo hiệu suất được đưa ra để đánh giá cách hoạt động của agent, và nó thường có thể đạt được hiệu suất cao mặc dù không biết rõ ràng về thước hiệu suất.

```
function SKELETON-AGENT(percept) return action
  static: memory, the agent's memory of the world

  memory  $\leftarrow$  UPDATE-MEMORY(memory,percept)
  action  $\leftarrow$  CHOOSE-BEST-ACTION(memory)
  memory  $\leftarrow$  UPDATE-MEMORY(memory,action)
  return action
```

Hình 4: Một agent khung

Tại sao không tìm thấy các câu trả lời?

Hãy bắt đầu với cách đơn giản nhất có thể, chúng ta viết một agent chương trình - bảng tra cứu (hình 5). Nó hoạt động dựa trên việc lưu trữ trên bộ nhớ toàn bộ chuỗi kết quả tri giác, và sử dụng nó để đưa vào bảng, bao gồm hành vi thích hợp cho tất cả các chuỗi kết quả tri giác có thể.

```
function TABLE-DRIVEN-AGENT(percept) return action
  static: percept, a sequence, initially empty
    table, a table, index by percept sequence, initially fully specified
    append percept to the end of percept
    action  $\leftarrow$  LOOKUP(percept,table)
  return action
```

Hình 5: Một agent dựa trên lý thuyết bảng tra cứu.

Nó cung cấp tài liệu để xem xét tại sao những đề nghị sau bị thất bại:

1. Bảng cần cho một điều gì đó, đơn giản như một agent chỉ có thể chơi cờ với khoảng 35100 nước đi.
2. Nó sẽ cần một khoảng thời gian không nhiều lầm để người thiết kế xây dựng bảng.
3. Agent này không tự giải quyết tất cả, bởi vì kết quả tính toán cho các hành vi hay nhất đều được xây dựng sẵn. Cho nên nếu môi trường thay đổi một cách đột ngột, thì agent này sẽ bị thất bại.
4. Thậm chí nếu chúng ta trao cho agent một cơ chế tự học tốt nhất, để nó có thể có khả năng tự giải quyết, thì nó vẫn cần phải học mãi mãi.

Mặc cho tất cả các điều đó, TABLE-DRIVEN-AGENT vẫn làm cái điều chúng ta muốn: nó thực hiện đầy đủ phép ánh xạ agent theo yêu cầu. Do đó không chắc chắn để khẳng định: “Nó không thông minh”.

Một ví dụ

Phần này giúp xem xét một môi trường riêng, với mục đích sự thảo luận của chúng ta trở nên cụ thể hơn. Nguyên nhân chính của sự hiểu biết của nó, và nguyên nhân nó yêu cầu một phạm vi rộng những kỹ năng. Chúng ta sẽ xem xét công việc thiết kế tự động hóa lái taxi. Cần lưu ý rằng hệ thống hiện thời có phần vượt ra ngoài những phạm vi khả năng công nghệ cho phép, mặc dù hầu hết những phần hợp thành là sẵn có theo một vài mẫu. Nhiệm vụ lái xe là cực kỳ phức tạp, tổ hợp các tình huống mới có thể xảy ra là không có giới hạn.

Dầu tiên chúng ta phải nghĩ ra các đối tượng tri giác, các hành vi, các mục tiêu và môi trường cho taxi (xem hình 6).

Kiểu Agent	Đối tượng tri giác (Percepts)	Hành vi (Actions)	Mục tiêu (Goals)	Môi trường Environment
Lái taxi	Các camera, công tơ mét, hệ thống GPS, microphone	định hướng, tăng giảm tốc, phanh, thông báo với hành khách	An toàn, nhanh, đúng luật, hành trình thoải mái, kinh tế nhất	các con đường,

Hình 6: Kiểu agent “Lái xe Taxi”

Taxi cần sẽ biết nó ở đâu, trên con đường nào và đang chạy nhanh chậm ra sao. Những thông tin này có thể thu được từ các percept. Đó là một hoặc nhiều camera (Controllable TV camera), công tơ mét đo tốc độ (Speedometer), và đồng hồ đo đoạn đường đi (Odometer). Để điều khiển xe cộ một cách chính xác, nhất là trên đoạn đường cong, cần phải có một bộ điều khiển tăng giảm tốc (accelerometer). Cũng cần biết trạng thái máy móc của xe cộ, vì vậy cần phải có một hệ thống các cảm biến cơ học và điện tử (engine and electrical sensor). Có thể có các thiết bị không có giá trị với người điều khiển thông thường như: hệ thống vệ tinh định vị toàn cầu GPS, để cung cấp thông tin chính xác vị trí trên bản đồ điện tử, hoặc cảm biến hồng ngoại (infrared sensor) hoặc cảm biến âm (sonar sensor) để phát hiện khoảng cách với các ô tô và các chướng ngại vật khác. Cuối cùng, cần có một microphone hoặc một bàn phím (keyboard) để cho các hành khách nói với nhau nói đến của họ.

Các hành vi có thể đối với agent “Lái taxi” sẽ ít nhiều tương tự như đối với một con người điều khiển taxi: điều khiển động cơ thông qua bàn đạp gas, điều khiển hướng, điều khiển phanh. Ngoài ra, sẽ cần đưa ra “đầu ra” trên màn hình hoặc thiết bị âm thanh để giao tiếp với hành khách; và một vài cách có thể để truyền thông với các xe cộ khác.

Khả năng phạm vi thực hiện (performance measure) mà chúng ta muốn hệ thống lái xe tự động đạt được là gì? Các năng lực mong muốn bao gồm: đến đích chính xác, sự tiêu thụ năng lượng là ít nhất, hao mòn ít nhất, ít hỏng hóc nhất, thời gian hành trình ngắn nhất hoặc chi phí ít nhất hoặc cả thời gian và chi phí

ít nhất, ít vi phạm luật giao thông nhất, làm ảnh hưởng đến các lái xe khác là ít nhất, an toàn nhất, hành khách thoải mái nhất, thu lợi cao nhất. Rõ ràng, một vài mục tiêu đó là mâu thuẫn, vì vậy sẽ cần phải có sự thỏa hiệp.

Cuối cùng, với dự án thực tế, chúng ta phải quyết định môi trường lái xe taxi sẽ đối diện. Nó phải hoạt động trên các con đường địa phương, hoặc là trên các xa lộ? Đó là Nam California, nơi ít khi có tuyết rơi, hoặc Alaska nơi tuyết thường xuyên rơi? Luôn luôn lái xe bên phải hoặc có thể chúng ta muốn linh hoạt để có thể di chuyển trong trường hợp taxi hoạt động ở nước Anh hoặc Nhật Bản. Rõ ràng, môi trường bị hạn chế hơn, vẫn đề thiết kế dễ dàng hơn.

Bây giờ, chúng ta phải quyết định như thế nào để xây dựng chương trình thực để thực hiện phép ánh xạ từ các đối tượng tri giác đến các hành vi. Chúng ta thấy rằng các khía cạnh điều khiển khác nhau sẽ đề xuất các kiểu agent chương trình khác nhau. Có 4 kiểu agent được chỉ ra:

- Các agent phản xạ đơn giản - simple reflex agents
- Các agent theo dõi (các vật có sự kiện liên quan)
- Các agent mục tiêu cơ bản - Goal-based agents
- Các agent “lợi ích” cơ bản - Utility-based agents

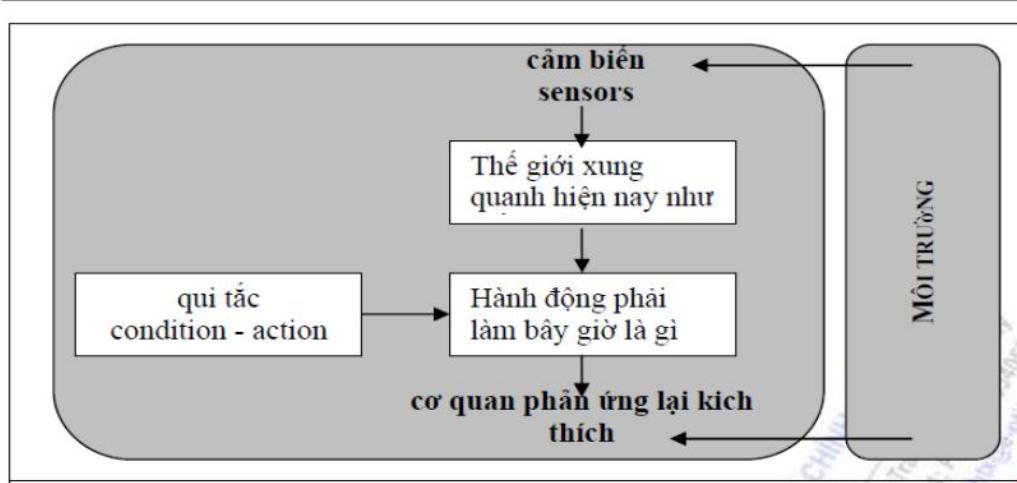
Simple reflex agents

Lựa chọn xây dựng “một bảng tra cứu hiện” là một vấn đề khó. Đầu vào thị giác từ một camera đơn đưa vào với tốc độ 50 Mb/giây (25 frame/giây, 1000×1000 pixel với 8 bit màu và 8 bit cường độ). Vì vậy, bảng tra cứu cho một giờ sẽ là $2^{60} \times 60 \times 50M$ trạng thái.

Tuy nhiên, chúng ta có thể tóm tắt từng phần của bảng bằng cách ghi nhớ các sự kết hợp vào thường xuyên xuất hiện một cách chắc chắn. Cho ví dụ, nếu chiếc xe phía trước phanh lại, và đèn phanh của nó sáng lên, thì người lái xe phải chú ý và bắt đầu phanh. Nói cách khác, một vài sự xử lý được thực hiện trước đầu vào thị giác sẽ thiết lập điều kiện mà chúng ta gọi là “ô tô phía trước đang phanh - The car in front is braking”, thì gây ra một vài sự thiết lập kết nối tới agent chương trình để hành động “bắt đầu phanh - initiate braking”. Chúng ta gọi là một sự kết nối một qui tắc **condition - action7** (phản xạ có điều kiện), viết như sau:

```
if car-in-front-is-braking then intiate-bracking
```

Con người cũng có nhiều kết nối, một vài trong số đó học được từ sự phản ứng lại - phản xạ có điều kiện (ví như việc lái xe) và một vài trong số đó là phản xạ tự nhiên (ví như việc chớp mắt khi một vật gì đó bay vào mắt). Theo cách này, trong cuốn sách, chúng ta sẽ thấy một số phương pháp khác nhau, sự kết nối có thể tự học và tự thực thi.

**Hình 7:** Sơ đồ biểu diễn simple reflex agent

Hình 8 đưa ra cấu trúc của một “simple reflex agent” dưới dạng giản đồ, chỉ ra qui tắc condition - action cho phép agent thực hiện kết nối từ tri giác đến hành động như thế nào. Chúng ta sử dụng hình chữ nhật để biểu thị trạng thái hiện tại bên trong tiến trình giải quyết của agent, hình oval biểu diễn thông tin cơ sở được sử dụng trong tiến trình. Agent chương trình cũng rất đơn giản, được chỉ ra ở hình 5.20. Chức năng INTERPRET-INPUT sinh ra một sự diễn tả trứu tượng của trạng thái hiện tại từ sự tri giác, RULE-MATCH trả về qui tắc đầu tiên trong tập các qui tắc mà nó phù hợp với sự mô tả trạng thái được đưa đến. mặc dù các agent có thể được thực hiện rất hiệu quả , nhưng phạm vi ứng dụng lại rất hẹp.

```

function SIMPLE-REFLEX-AGENT(percept) return action
  static: rules, a set of condition-action rules
  state  $\leftarrow$  INTERPRET-INPUT(percept)
  rule  $\leftarrow$  RULE-MATCH(state,rules)
  action  $\leftarrow$  RULE-ACTION[rule]
  return action

```

Hình 8: Kiểu agent “Lái xe”

Agent theo dõi (Agents that keep track of the world)

Simple reflex agent mô tả ở phần trước sẽ chỉ làm việc nếu sự giải quyết chính xác có thể được dựa trên tri giác hiện thời. Nếu chiếc ô tô phía trước là một mô hình đã xuất hiện trước đây không lâu, và bây giờ

có ánh sáng đèn phanh được mang lại từ trung tâm kết quả ở nước Mỹ, thì từ một hình ảnh đơn sê có thể nó rằng nó đang phanh. Không may, các mô hình cũ có sự khác biệt về hình dáng đèn hậu, đèn phanh, đèn xi nhan, và không phải bao giờ cũng có khả năng trả lời rằng chiếc ô tô phía trước đang phanh. Vì vậy, dù là việc phanh đơn giản, thiết bị lái của chúng ta vẫn phải duy trì một vài phần nào đó tình trạng bên trong theo trình tự để lựa chọn hành động. Ở đây tình trạng bên trong không quá lớn - nó chỉ cần frame trước đó từ camera để dò tìm khi hai đèn đỏ ở rìa xe cộ cùng sáng hoặc tắt đồng thời.

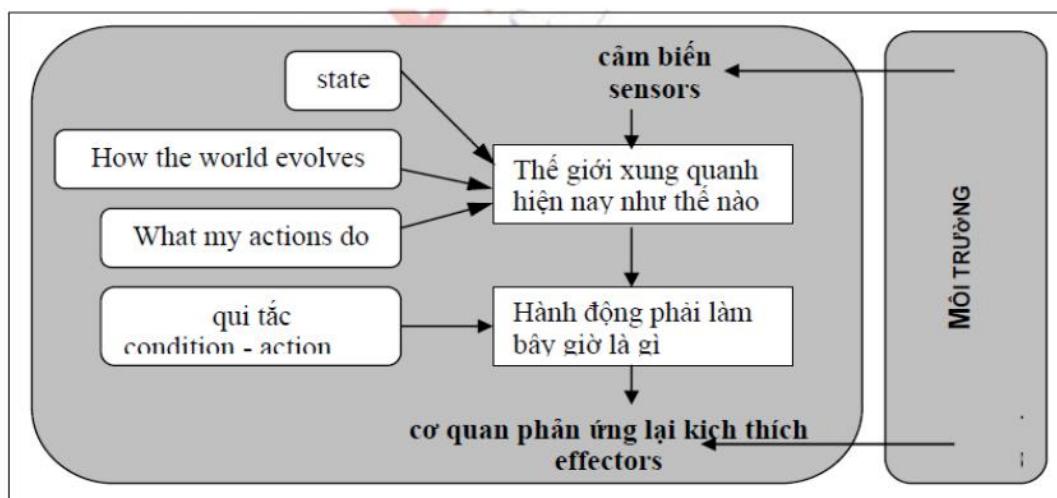
Hãy xem rõ ràng hơn trường hợp sau: thỉnh thoảng, người lái xe nhìn vào gương chiếu hậu để kiểm soát các xe cộ phía sau. Khi lái xe đang không quan sát gương, không quan sát thấy xe cộ ở làn đường bên cạnh (tức là không thể phân biệt được các trạng thái, ở đó chúng xuất hiện và không xuất hiện); vì vậy để quyết định chuyển làn xe một cách khéo léo thận trọng, người lái xe cần phải biết dù là chúng có ở đó hay không.

Vẫn đề này sinh, bởi vì các sensor không cung cấp một cách đầy đủ trạng thái. Trong các trường hợp đó, agent có thể duy trì một vài thông tin trạng thái bên trong để nhận biết được các trạng thái phát sinh giống như giác quan đem lại, nhưng dù sao cũng khác nhau một cách đáng kể. Ở đây, “khác nhau một cách đáng kể” có ý nghĩa rằng các hành động khác nhau thích hợp cho hai trạng thái.

Việc cập nhật thông tin trạng thái bên trong này thực hiện bằng cách yêu cầu hai loại kiến thức được mã hoá trong agent chương trình. Thứ nhất, chúng cần một vài thông tin “thế giới xung quanh tiến triển không phụ thuộc vào agent như thế nào - how the world evolves” - cho ví dụ, một chiếc ô tô đang vượt qua nhin chung sẽ là tiến dần đến phía sau hơn là ngay tức thì. Thứ hai, chúng ta cần một vài thông tin “Các hành động của agent phản ứng lại thế giới xung quanh như thế nào - What my actions do” - cho ví dụ, khi agent rẽ sang đường nhỏ bên phải, có một cái rãnh cắt ngang trước mặt nó sẽ tiến về phía trước hoặc lùi lại về phía sau theo con đường lớn nơi có con đường nhỏ khác.

Hình 8 đưa ra cấu trúc của agent phản xạ, cho biết tri giác hiện tại được kết hợp với trạng thái bên trong cũ để sinh ra sự mô tả cập nhật của trạng thái hiện thời. Agent chương trình được chỉ ra ở hình 9. Một nhiệm vụ thú vị là chức năng UPDATE-STATE, nó đáp ứng cho việc tạo ra sự mô tả trạng thái bên trong mới. Giống như công việc phiên dịch tri giác mới dưới ánh sáng của kiến thức có sẵn về trạng thái, nó sử dụng thông tin “how the world evolves” để theo dõi các phần không thấy được của thế giới xung quanh, và cũng phải biết các hành động của agent làm gì với trạng thái của thế giới.

Agent mục tiêu (Goal-based agent)



Hình 9: Agent chương trình

Cần biết rằng trạng thái hiện thời của môi trường luôn luôn không đủ để quyết định làm việc việc gì. Cho ví dụ, tại nơi gặp nhau của các con đường, taxi có thể rẽ trái, rẽ phải, hoặc đi thẳng. Quyết định đúng đắn trên cơ sở “taxi đang cố gắng đi đến đâu”. Nói cách khác, giống như sự mô tả trạng thái hiện thời, agent cũng cần một số thông tin về mục tiêu, nó diễn tả trạng thái đang mong muốn - ví dụ, đó là điểm cần đến của hành khách. Agent chương trình có thể kết hợp điều này với thông tin về kết quả của các hành động có thể xảy ra (giống như thông tin được sử dụng để cập nhật trạng thái trong của agent phản xạ) để lựa chọn các hành động đạt được mục tiêu. Đôi khi, điều đó sẽ là đơn giản, khi sự thoả man mục tiêu là kết quả trực tiếp từ một hành động đơn giản; đôi khi, nó đòi hỏi phải khéo léo hơn, khi agent xem như các chuỗi xoắn dài và luẩn quẩn để tìm cách đạt được mục tiêu. Chú ý rằng quyết định đưa ra của loại agent này về cơ bản khác với các quy tắc condition-action được mô tả phần trước, bởi vì nó cần phải xem xét trong tương lai cả hai vấn đề “Điều gì sẽ xảy ra nếu tôi làm gì đó? - What will happen if I do such-and-such?” và “Điều đó sẽ mang lại cho tôi hạnh phúc? - Will that make me happy”. Trong thiết kế agent phản xạ, thông tin này không được sử dụng một cách rõ ràng, bởi vì người thiết kế đã tính trước hành động chính xác cho các trường hợp khác nhau. Agent phản xạ phanh xe khi nó nhìn thấy đèn phanh của chiếc ô tô phía trước sáng nó sẽ đi chậm lại. Theo cách diễn tiến thông thường của thế giới xung quanh, hành động chỉ đạt được mục tiêu là không dung các xe khác khi phanh xe. Mặc dù agent mục tiêu xem ra kém hiệu quả hơn, nó khó thuyết phục. Nhưng nếu trời bắt đầu mưa, agent có thể cập nhật hiểu biết của nó về việc phanh xe sẽ hoạt động hiệu quả như thế nào, điều đó sẽ là lý do để tự động đưa ra tất cả các cách hành động có liên quan để sửa đổi sao cho phù hợp với điều kiện mới. Mặt khác, với agent phản xạ, chúng ta sẽ phải viết lại một số lượng lớn các qui tắc condition-action. Dĩ nhiên, agent mục tiêu linh hoạt hơn nhiều đối với việc đi đến các đích khác nhau. Một cách đơn giản để xác định một cái đích mới, chúng ta có thể đưa ra agent mục tiêu để theo kịp với cách hành động mới. Các qui tắc của agent phản xạ lái xe khi rẽ và khi đi thẳng sẽ chỉ làm việc với một đích đơn lẻ, và các qui tắc đó sẽ phải thay đổi tất cả khi cần di đến các nơi mới.

2.4 Môi trường (Environments)

Đầu tiên, chúng ta sẽ mô tả các kiểu khác nhau của môi trường và chúng ảnh hưởng đến việc thiết kế các agent như thế nào. Sau đó, chúng ta sẽ mô tả các chương trình môi trường mà nó được sử dụng như là “nơi kiểm tra” các agent chương trình.

Các thuộc tính của môi trường (Properties of Environments)

Môi trường có những đặc trưng riêng. Các đặc trưng chính được chỉ ra như sau:

Có thể được và Không thể được (Accessible vs. Inaccessible)

Nếu cơ cấu giác quan của agent đem đến cho nó một trạng thái đầy đủ của môi trường thì chúng ta nói rằng môi trường là có thể đối với agent. Một môi trường là thực sự có thể nếu các sensor phát hiện ra tất cả các khía cạnh liên quan đến sự lựa chọn hành động. Một môi trường có thể cần thuận tiện bởi vì agent mong muốn không phải duy trì một số trạng thái bên trong để theo dõi thế giới xung quanh.

Tiền định và không tiền định (Deterministic vs. Nondeterministic)

Nếu trạng thái tiếp theo của môi trường là hoàn toàn xác định được thông qua trạng thái hiện thời và hành động mà agent đã lựa chọn, thì chúng ta nói môi trường là tiền định. Theo nguyên tắc, agent mong muốn không phải lo nghĩ về một môi trường có thể không chắc chắn tiền định. Tuy nhiên, nếu một môi trường là không thể thì nó có thể được xem như không tiền định. Điều đó đặc biệt đúng, nếu môi trường phức tạp gây ra sự khó khăn cho việc theo dõi tất cả các mặt không thể của nó. Vì vậy, môi trường tiền định hoặc môi trường không tiền định đối với điểm nhìn của agent thường được coi là tốt nhất.

Phân đoạn và không phân đoạn (Episodic vs. Nonepisodic)

Trong môi trường phân đoạn, kinh nghiệm của agent bị phân ra thành từng đoạn. Trong mỗi đoạn gồm có sự nhận biết của agent và hành khi có được nhận biết đó. Đặc tính của hành động phụ thuộc vào chính đoạn đó, bởi vì các đoạn tiếp theo không phụ vào các hành động trong đoạn trước đó. Môi trường phân đoạn là rất đơn giản vì agent không muốn nghĩ xa.

Tĩnh và Động (Static vs. Dynamic)

Nếu môi trường có thể thay đổi trong khi agent đang cân nhắc, thì chúng ta nói rằng môi trường là động

đối với agent, ngược lại môi trường là tĩnh. Môi trường tĩnh rất dễ giao tiếp bởi vì agent mong muốn không phải theo dõi thế giới xung quanh trong khi đang quyết định lựa chọn hành động và cũng mong muốn không phải lo nghĩ thời gian đã trôi qua. Nếu môi trường không thay đổi theo thời gian nhưng sự thực hiện của agent thành công, thì chúng ta nói rằng môi trường phần nào là động.

Rời rạc và Liên tục (Discrete vs. Continuous)

Nếu có một số lượng giới hạn nhất định các tri giác và hành động rõ ràng thì ta nói rằng môi trường là rời rạc. Môi trường “Chơi cờ” là rời rạc - có một số lượng cố định “nước đi có khả năng” trong mỗi lần di. Môi trường “Lái taxi” là liên tục - tốc độ, vị trí của taxi và các xe cộ khác nhận giá trị trong một khoảng giá trị liên tục8.

Chúng ta sẽ xem xét xem các kiểu môi trường khác nhau yêu cầu các agent chương trình khác nhau như thế nào để xử lý chúng hiệu quả. Nó sẽ được đưa ra như bạn mong đợi, đó là trường hợp khó nhất: môi trường không thể, môi trường không phân đoạn, môi trường động và môi trường liên tục. Nó cũng có thể được đưa ra trong nhiều hoàn cảnh thực tế là rất phức tạp, đến mức phải thảo luận xem trên thực tế chúng có thực sự tiền định hay chúng được xem như không tiền định.

Môi trường	Có thể	Tiền định	Phân đoạn	Tinh	Rời rạc
Chess with a clock	Yes	Yes	No	Semi	Yes
Chess without a clock	Yes	Yes	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Medical diagnosis system	No	No	No	No	No
Image-analysis system	Yes	Yes	Yes	Semi	No
Part-picking robot	No	No	Yes	No	No
Refinery controller	No	No	No	No	No
Interactive English tutor	No	No	No	No	Yes

Bảng trên liệt kê các thuộc tính của một số họ môi trường. Chú ý rằng các câu trả lời có thể thay đổi phụ thuộc vào việc bạn dựa trên các môi trường và các agent như thế nào. Ví dụ, Poker là tiền định nếu agent có thể theo dõi trạng thái của các quân bài trong cỗ bài, nhưng nó là không tiền định nếu nó không thể. Cũng như vậy, nhiều môi trường là phân đoạn ở mức hơn cho từng hành động riêng lẻ của agent. Ví dụ, một vòng thi đấu cờ gồm có một loạt các trận thi đấu, mỗi trận thi đấu là một phân đoạn, bởi vì tất cả các nước đi của một trận đấu trong toàn bộ sự thực hiện của agent không bị ảnh hưởng bởi các nước đi của trận đấu tiếp theo. Mặt khác, các nước đi trong phạm vi một trận đấu đơn lẻ dĩ nhiên là ảnh hưởng lẫn nhau, đến mức agent cần phải nghĩ trước vài ba nước đi.

Các chương trình môi trường (Environment programs)

Chương trình môi trường trong hình 10 minh họa mối quan hệ cơ bản giữa các agent và các môi trường. Chúng ta sẽ thấy nó thuận tiện cho nhiều ví dụ và nhiều bài tập sử dụng môi trường mô phỏng sinh ra cấu trúc chương trình đó. Việc mô phỏng một hoặc nhiều agent được thực hiện giống như là đưa vào và sắp xếp lặp đi lặp lại để sinh ra trong mỗi agent các tri giác đúng và nhận lại hành động. Việc mô phỏng sau đó cập nhật môi trường trên cơ sở các hành động, và các tiến trình động khác có thể trong môi trường mà các tiến trình đó không được quyết định bởi agent (ví dụ trời mưa). Vì vậy, môi trường được xác định bởi trạng thái ban đầu và hàm cập nhật. Một vấn đề hiển nhiên, một agent làm việc trong môi trường mô phỏng cũng phải làm việc giống như trong môi trường thực mà nó mô phỏng đó là: các loại tri giác, các hành động.

Thủ tục RUN-ENVIRONMENT rèn luyện các agent trong một môi trường một cách đúng đắn. Đối với một vài loại agent, ví dụ như các agent tham gia vào cuộc đối thoại bằng ngôn ngữ tự nhiên, nó có thể dễ dàng theo dõi hành vi của chúng. Hàm RUN-EVAL-ENVIRONMENT áp đặt một phạm vi thực hiện đối với mỗi agent và trả về danh sách các điểm số kết quả. Các biến score theo dõi điểm của mỗi agent.

Nói chung, phạm vi thực hiện có thể được quyết định bởi toàn bộ chuỗi trạng thái sinh ra trong quá trình chương trình hoạt động.

```
procedure RUN-ENVIRONMENT(state, UPDATE-Fn, agents, termination)
inputs: state, the initial state off the environment
          Update-Fn, function on modify the environment
          agents, a set of agents
          termination, a predicate to test when we are done
repeat
    for each agent in agents do
        Percept[agent]  $\leftarrow$  Get-Percept(agent,state)
    end
    for each agent in agents do
        Action[agent]  $\leftarrow$  Program[agent](Percept[agent])
    end
    state  $\leftarrow$  Update-Fn(actions,agents,state)
until termination(state)
```

Hình 10: Chương trình mô phỏng agent cơ bản. Nó sinh ra cho mỗi agent các tri giác của chúng, đưa ra một hành động từ mỗi agent, và sau đó cập nhật môi trường

```

function Run-Eval-Environment(state, Update-Fn, agents, termination,
                           Performance-Fn) returns scores
  local variable: scores, a vector thw same size as agents, all 0
  repeat
    for each agent in agents do
      Percept[agent]  $\leftarrow$  Get-Percept(agent,state)
    end
    for each agent in agents do
      Action[agent]  $\leftarrow$  Program[agent](Percept[agent])
    end
    state  $\leftarrow$  Update-Fn(actions,agents,state)
    scores  $\leftarrow$  Performance-Fn(scores,agents,state)
  until termination(state)
  return scores

```

Hình 11: Một chương trình mô phỏng môi trường mà nó theo dõi phạm vi thực hiện cho mỗi agent.

TÓM TẮT

Chương này chúng ta đã giới thiệu tổng quan một số vấn đề của Trí tuệ nhân tạo, chúng ta đã hình dung được việc thiết kế agent. Những điểm chủ yếu đó là:

Agent là một cái gì đó mà nó có thể nhận biết và tác động lại môi trường. Chúng ta phân biệt agent với kiểu kiến trúc và agent chương trình.

Agent lý tưởng luôn luôn thực hiện hành động được mong đợi đạt được phạm vi thực hiện cực đại, đem lại chuỗi kết quả tri giác nó thấy.

Agent là tự trị để phạm vi mà các hành động của nó lựa chọn được quyết định bởi chính kinh nghiệm của nó, một phần nào đó sự hiểu biết của môi trường được xây dựng bởi người thiết kế.

Chương trình agent ánh xạ từ một tri giác đến một hành động, khi cập nhật một trạng thái trong.

Có nhiều chương trình agent cơ bản khác nhau được thiết kế, được quyết định bởi loại thông tin và việc sử dụng trong quá trình ra quyết định. Các thiết kế dẫn đến sự khác nhau về hiệu quả, tính súc tích và tính linh động. Các thiết kế chương trình agent thích hợp quyết định bởi các tri giác, các hành động, các mục tiêu và môi trường.

Các agent phản xạ (reflex agents) phản ứng lại ngay tức thì các tri giác, các agent mục tiêu cơ bản (goal-based agents) hành động với mục đích là chúng sẽ giành được được các mục tiêu của chúng, và các agent lợi ích (utility-based agent) cố gắng để đạt được sự “hạnh phúc” lớn nhất.

Quá trình ra quyết định bằng việc sử dụng sự hiểu biết là trọng tâm của Trí tuệ nhân tạo và thiết kế thành công agent. Điều này có nghĩa là việc biểu diễn sự hiểu biết là rất quan trọng.

Một vài môi trường đòi hỏi khắt khe hơn rất nhiều. Các môi trường là “không thể được”, “không tiền định”, “không phân đoạn”, “động” và “liên tục” là những thách thức lớn.

3 CÂU HỎI VÀ BÀI TẬP

Bài tập 1

Trình bày mô hình: mạng nơ ron sinh vật và mạng nơ ron nhân tạo. Nêu rõ các thành phần cơ bản của chúng.

Bài tập 2

Trình bày thuật toán lan truyền ngược

Bài tập 3

Ứng dụng mạng nơ ron trong xử lý tín hiệu. Trình bày ví dụ chứng tỏ mang BAM có khả năng nhớ và khả năng tách lõi.

Bài tập 4

Trình bày các hàm thuộc (hàm thành viên) thường sử dụng trong logic mờ ?

Bài tập 5

Trình bày các phép logic thường sử dụng trong logic mờ ?

Bài tập 6

Trình bày các toán tử sau đây trong giải thuật di truyền?

a) toán tử sinh sản

b) toán tử ghép chéo

c) toán tử đột biến

Bài tập 7

Trình bày cấu trúc của Agent thông minh

GIẢI THÍCH TỪ VÀ THUẬT NGỮ VIẾT TẮT

AI : Artificial Intelligent: Trí tuệ nhân tạo

ANN : Artificial Neural Network

ADALINE (Adalines) : ADaptive LINear Element mạng do Widrow nêu năm 1960

ART : Adaptive Resonance Networks

(knowledgeBase: viết tắt tiếng Anh: KB, viết tắt tiếng Việt: CSTT)

BP : Back Propagation Network

CPU : Center Processing Unit

GPS : General Problem Solver

EP : Evolutionary Programming: Lập trình tiến hóa

HMM : Hidden Markov Model

FIR : Finite Impulsse Response : Bộ lọc đáp ứng xung hữu hạn

GA : Genetic Algorithm: Giải thuật di truyền,

PC : Personal Computer: Máy tính cá nhân.

CSDL .Cơ sở dữ liệu

NLP: Natural Language Processing

IR : Information Retrieval

NP Noun Phrase : danh từ

VP :Verb Phrase : động từ