Chúng tôi sẽ xây dựng các agent thông minh xuyên suốt cuốn sách.Tất cả chúng sẽ có cùng một bộ khung, cụ thể là, chấp nhận các đối tượng nhận thức từ một môi trường và tạo ra hành động. Các phiên bản đầu của các chương trình agent sẽ có một hình thức rất đơn giản (hình 2.4).Mỗi agent sẽ sử dụng một số cấu trúc dữ liệu nội bộ sẽ được cập nhật như kết quả tri giác mới. Những cấu trúc dữ liệu được hoạt động của agent làm thủ tục ra quyết định để tạo ra một sự lựa chọn hành động, mà sau đó thông qua kiến trúc để được thực thi.

Có hai điều cần lưu ý về chương trình khung này.Đầu tiên, mặc dù chúng ta định nghĩa việc mapping các agent như là một chức năng từ các chuỗi percept để hành động, các chương trình agent chỉ nhận được một percept duy nhất là đầu vào của nó. Đó là vào các agent để xây dựng trình tự percept trong bộ nhớ, nếu mà chúng muốn. Trong một số môi trường, có thể là khá thành công mà không cần lưu trữ  
trình tự percept, và trong các lĩnh vực phức tạp, có thể lưu trữ các chuỗi số hoàn chỉnh.

Hình 2.4

Thứ hai, biện pháp hoặc việc thực hiện mục tiêu không phải là một phần của chương trình khung.Điều này là do các biện pháp hiệu quả được áp dụng bên ngoài để đánh giá hành vi của các agent, và nó thường có thể đạt được hiệu suất cao mà không cần có kiến thức rõ ràng của các biện pháp thực hiện (xem ví dụ: the square-root agent).

Tại sao không tìm kiếm các câu trả lời?  
Chúng ta hãy bắt với cách đơn giản nhất chúng ta có thể nghĩ ra để viết chương trình một bảng tra cứu agent. Hình 2.5 cho thấy chương trình agent. Nó hoạt động bằng cách giữ trong bộ nhớ toàn bộ chuỗi percept của , và sử dụng nó để đánh chỉ số vào bảng, trong đó có các hành động thích hợp cho tất cả các chuỗi percept .

Đây là hướng dẫn để xem xét lý tại sao đề xuất này doomed đến thất bại:

1.Bảng cần thiết cho một cái gì đó đơn giản như một tác nhân mà chỉ có thể chơi cờ sẽ có khoảng 35.100 mục.  
2. Nó sẽ mất khá nhiều thời gian cho các nhà thiết kế để xây dựng các bảng.  
3. agent không có quyền tự chủ ở tất cả, bởi vì việc tính toán hành động tốt nhất là hoàn toàn được xây dựng bên trong. Vì vậy, nếu môi trường thay đổi theo một cách bất ngờ, agent sẽ bị mất.

Hình 2.5

4. Ngay cả khi chúng tôi đã cho các agent một cơ chế học tập tốt để nó có thể có một mức độ tự chủ, nó sẽ mất mãi mãi để tìm hiểu giá trị quyền cho tất cả các mục trong bảng.

Một ví dụ  
Tại thời điểm này, sẽ rất hữu ích để xem xét một môi trường cụ thể, do đó, cuộc thảo luận của chúng ta có thể trở nên cụ thể hơn. Chủ yếu là vì sự quen thuộc của nó, và bởi vì nó liên quan đến một loạt các kỹ năng, chúng ta sẽ nhìn vào công việc thiết kế một lái xe taxi tự động. Chúng ta nên chỉ ra, trước khi người đọc báo động, rằng một hệ thống như hiện nay nằm ngoài khả năng của công nghệ hiện có, mặc dù hầu hết các thành phần có sẵn trong một số form.6 Nhiệm vụ lái xe đầy đủ là cực kỳ mở không có hạn chế các kết hợp mới lạ của tình huống có thể phát sinh (đó cũng là một lý do tại sao chúng tôi đã chọn nó như là một trọng tâm cho cuộc thảo luận).  
Đầu tiên chúng ta phải suy nghĩ về percepts, hành động, mục tiêu và môi trường cho xe taxi. Chúng được tóm tắt trong hình 2.6 và thảo luận lần lượt

Hình 2.6Taxi sẽ cần phải biết nó ở đâu, những gì khác trên đường, và nó nhanh như thế nào nó sẽ.Những thông tin này có thể thu được từ percepts được cung cấp bởi một hoặc nhiều camera TV kiểm soát, đồng hồ công tơ mét và đồng hồ đo cây số.Để kiểm soát xe đúng cách, đặc biệt trên các đường cong, nó cần phải có một gia tốc, nó cũng sẽ cần biết trạng thái cơ học của chiếc xe, do đó, nó sẽ cần những mảng thông thường của động cơ và cảm biến hệ thống điện. Nó có thể có công cụ mà không có sẵn trình điều khiển: hệ thống định vị vệ tinh toàn cầu (GPS) để cung cấp cho nó chính xác vị tin thông tin liên quan đến một bản đồ điện tử, hoặc các cảm biến hồng ngoại hoặc sonar để phát hiện khoảng cách với các xe khác và trở ngại. Cuối tin, nó sẽ cần một micro hoặc bàn phím cho hành khách để cho nó nói điểm đến của họ nó. Những hành động có sẵn cho một lái xe taxi sẽ có nhiều hơn hoặc ít hơn so với con người lái: kiểm soát động cơ thông tin bàn đạp ga và kiểm soát lái và phanh.Ngoài ra, sẽ cần đầu ra màn hình hoặc một bộ tổng hợp giọng nói để nói lại cho hành khách, và có lẽ một số cách để giao tiếp với các loại xe khác.

Những biện pháp tự động nào chúng tôi mong muốn đưa rachất lượng mong muốn nhận được đến đích chính xác, giảm thiểu tiêu thụ nhiên liệu và hao mòn, giảm thiểu thời gian chuyến đi và / hoặc chi phí, giảm thiểu các hành vi vi phạm luật giao thông và rối loạn các trình điều khiển khác, tối đa hóa an toàn và tiện nghi cho hành khách, tối đa hóa lợi nhuận. Rõ ràng, một số mục tiêu bị xung đột, do đó sẽ liên quan đến sự thoả hiệp.

Cuối cùng, là một dự án thực tế, chúng ta sẽ cần phải quyết định loại taxi tự lái xe sẽ phải đối mặt với những gì ở môi trường. Cần hoạt động trên đường địa phương, hoặc trên đường cao tốc?Nó sẽ được ở miền Nam California hoặc ở Alaska Nó luôn luôn được lái xe bên phải, hoặc chúng tôi có thể muốn nó được linh hoạt, đủ để lái xe bên trái trong trường hợp chúng ta muốn hoạt động taxi ở Anh hay Nhật Bản?Rõ ràng, hạn chế môi trường, vấn đề thiết kế sẽ dễ dàng hơn.

Bây giờ chúng ta phải quyết định làm thế nào để xây dựng một chương trình thực tế để thực hiện việc mapping từ các percepts hành động.Chúng ta sẽ thấy rằng các khía cạnh khác nhau của lái xe đề xuất các loại khác nhau của chương trình agent . Chúng tôi sẽ xem xét bốn loại của chương trình agent :

Đơn giản phản ứng các agent  
theo dõi agent  
Các agent dựa trên mục tiêu

Các agent dựa trên tiện ích

Đơn giản phản ứng agents

Các tùy chọn xây dựng một bảng tra hình rõ hình là các câu hỏi.Các đầu vào hình hình từ một máy ảnh duy hình có tỷ lệ 50 MB mỗi giây (25 khung hình mỗi giây, 1000 1000 pixel với 8 bit của màu sắc và 8 bit thông tin cường độ). Vì vậy, các bảng tra cứu cho một  
tiếng sau sẽ là 2 60x60 x50M mục.

Tuy nhiên, chúng ta có thể tóm tắt một phần của bảng bằng cách ghi nhận nhất định phổ biến đầu vào / đầu ra các hiệp hội.Ví dụ, nếu xe hơi ở phanh trước và đèn phanh trên, sau đó người lái xe cần phải nhận thấy điều này và bắt đầu phanh.Nói cách khác, một số xử lý được thực hiện vào đầu vào trực quan để thiết lập điều kiện chúng ta gọi là "chiếc xe phía trước phanh", sau đó kích hoạt một số kết nối được thiết lập trong chương trình agent hành động "bắt đầu phanh".Chúng tôi kêu gọi một kết nối một quy tắc điều kiện hành động bằng văn bản như:

**if** *car-in-front-is-braking* **then** *initiate-braking*

Con người cũng có nhiều kết nối như vậy, một số trong đó được học phản ứng (như lái xe) và một số trong số đó là phản xạ bẩm sinh (như chớp mắt khi một cái gì đó gần mắt).trong tiến trình của cuốn sách, chúng ta sẽ thấy một số cách khác nhau mà kết nối này có thể được học và thực hiện.

Hình 2.7 cho cấu trúc của một agent phản xạ đơn giản ở dạng sơ đồ, hiển thị các quy tắc điều kiện hành động cho phép các agent để làm cho các kết nối từ percept hành động.(Đừng lo lắng nếu điều này có vẻ bình thường, nó sẽ thú vị hơn ngay). Chúng tôi sử dụng hình chữ nhật để biểu thị

Hình 2.7

Hình 2.8

Hình 2,8.Chức năng giải thích-INPUT tạo ra một trừu tượng mô tả trạng thái hiện tại từ các percept, và chức năng Rule Match trở về quy tắc đầu tiên trong các bộ quy tắc phù hợp với mô tả trạng thái đã cho. Mặc dù đại lý có thể được thực hiện rất hiệu quả  
(xem Chương 10), phạm vi áp dụng của chúng là rất hẹp, như chúng ta sẽ thấy.

Theo dõi Agent

agent phản ứng đơn giản mô tả trước đây sẽ chỉ làm việc nếu quyết định chính xác có thể được thực hiện trên cơ sở của percept hiện tại.Nếu chiếc xe ở phía trước là một mô hình gần đây, và có đèn phanh được gắn theo yêu cầu trực thuộc Trung ương Hoa Kỳ, thì nó sẽ có thể cho biết nếu nó được phanh từ một hình ảnh duy nhất. Thật không may, các mô hình cũ có cấu hình khác nhau của đèn hậu, đèn phanh, và tín hiệu đèn lần lượt, và nó không phải là luôn luôn có thể để nói nếu chiếc xe đang phanh.Vì vậy, ngay cả đối với các quy tắc phanh đơn giản, người lái xe của chúng tôi sẽ có để duy trì một số loại trạng thái bên trong để lựa chọn một hành động. Ở đây, trạng thái bên trong không phải là quá rộng rãi nó chỉ cần khung trước đó từ máy ảnh để phát hiện khi hai đèn đỏ ở các cạnh của chiếc xe đi vào hoặc tắt cùng một lúc. Hãy theo theo trường hợp rõ ràng hơn sau: theo thời gian, người lái xe nhìn vào gương chiếu hậu để kiểm tra về các địa điểm xe gần theo.Khi người lái xe không nhìn vào gương, xe trong làn đường tiếp theo vô hình (ví dụ, các trạng thái, trong đó chúng xuất hiện và vắng mặt không thể phân biệt), nhưng để quyết định một cơ động thay đổi làn đường, người lái xe cần phảibiết có hay không họ đang có.

Các vấn đề được minh họa bằng ví dụ này phát sinh bởi vì các bộ cảm biến không cung cấp quyền truy cập đến trạng thái hoàn toàn của thế giới.Trong những trường hợp như vậy, các đại lý có thể cần phải duy trì một số thông tin trạng thái nội bộ để phân biệt giữa các quốc gia trên thế giới tạo ra cùng một đầu vào cảm nhận, nhưng dù sao cũng là đáng kể khác nhau.Ở đây, "sự khác nhau đáng kể" có nghĩa là hành động khác nhau phù hợp trong hai tiểu bang.

Cập nhật thông tin trạng thái nội bộ như thời gian trôi qua yêu cầu hai loại kiến ​​thức được mã hóa trong chương trình agent. Trước triển, chúng ta cần một số thông tin về việc làm thế nào trên thế giới phát triển độc lập của các agent ví dụ, một chiếc xe vượt qua nói chung sẽ gần phía sau hơn là một giây phút trước đây. Thứ hai, chúng ta cần một số thông tin về cách của agent hành động của ảnh hưởng đến các thế giới-ví dụ, rằng khi agent thay đổi làn đường bên phải, có là một khoảng cách (ít nhất là tạm thời) trong các làn đường ở trước, hoặc là sau khi lái xetrong năm phút về hướng bắc trên đường cao tốc thường là khoảng năm dặm về phía bắc của một trong những nơi cách đây năm phút.

Hình 2.9 cho cấu trúc của các agent phản ứng, chỉ ra cách percept hiện nay được kết hợp với tình trạng nội bộ cũ để tạo ra các mô tả cập nhật của tình trạng hiện tại.Chương trình agent được thể hiện trong hình 2.10. Các phần thú vị là chức năng UPDATE –STATE, thứ mà l chịu trách nhiệm cho việc tạo ra các mô tả trạng thái nội bộ mới Cũng như giải thích các percept mới trong ánh sáng của kiến thức hiện tại về trạng thái, sử dụng thông tin như thế nào trên thế giới phát triển để theo dõi các phần vô hình của thế giới, và cũng phải biết về những hành động của agent làm cho trạng tháitrên thế giới. Ví dụ chi tiết trình bày trong các chương 7 và 17.

Các agent dựa trên mục tiêu

Hiểu biết về hiện trạng của môi trường không phải lúc nào cũng đủ để quyết định làm gì. Ví dụ, tại một ngã ba đường, taxi có thể rẽ trái, phải, hoặc đi thẳng vào.Quyết định quyền phụ thuộc vào nơi mà các taxi đang cố gắng để đến. Nói cách khác, cũng như mô tả trạng thái hiện tạiagent cần một số loại thông các các mục tiêu, trong đó mô tả các tình các cũng rất cần thiết, ví dụ, tại điểm đến của hành khách.Chương trình agent các có thể kết hợp với các thông tin về kết quả của các hành động có thể (các thông tin tương tự như đã được sử dụng để cập nhật trạng thái nội bộ tại các agent phản ứng) để lựa chọn hành động đạt được mục tiêu.

Hình 2.9

Hình 2.10

Chú ý rằng quyết định của loại hình này về cơ bản là khác nhau từ các quy định điều kiện hành động được mô tả trước đó, trong đó nó liên quan đến cả việc xem xét tương lai. "Điều gì sẽ xảy ra nếu tôi làm như vậy và như vậy?" Và "đó sẽ làm cho tôi hạnh phúc?" Trong thiết vì agent phản ứng, thông tin này không được sử dụng rõ ràng, bởi vì các nhà thiết kế đã precomputed hành động chính xác đối với các trường hợp khác nhau. agent phản ứng phanh khi nó thấy đèn phanh. Một agent dựa trên mục tiêu, về nguyên tắc, có thể viện lý rằng nếu chiếc xe phía trước có đèn phanh của nó, nó sẽ làm chậm.Từ cách thế giới thường tiến hóa, các hành động duy nhất mà sẽ đạt được mục tiêu không đụng vào xe khác là để phanh. Mặc dù các agent dựa trên mục tiêu xuất hiện kém hiệu quả, linh hoạt hơn rất nhiều. Nếu trời bắt đầu mưa, các agent có thể cập nhật kiến ​​thức của hệ thống phanh sẽ hoạt động hiệu quả, điều này sẽ tự động gây ra tất cả các hành vi có liên quan phải được thay đổi cho phù hợp với điều kiện mới.Đối với các agent phản ứng, mặt khác, chúng tôi sẽ phải viết lại một số lượng lớn các quy định điều kiện hành động. Tất nhiên, các agent dựa trên mục tiêu cũng linh hoạt hơn đối với việc đạt được điểm đến khác nhau.Đơn giản chỉ cần bằng cách xác định một điểm đến mới, chúng ta có thể nhận được các agent dựa trên mục tiêu để đến với một hành vi mới.Các agent phản ứng của quy tắc khi lần lượt, và khi đi thẳng sẽ chỉ làm việc cho một điểm đến duy nhất, tất cả đều phải được thay thế để đi một nơi nào đó mới.

Hình 2.11 cho thấy cấu trúc của các agent dựa trên mục tiêu.Chương 13 có chứa các chương trình agent chi tiết cho các agents dựa trên mục tiêu.

Hình 2.11

Các agent dựa trên tiện ích

một mình Mục tiêu không thực sự đủ để tạo ra hành vi có chất lượng cao. Ví dụ, có rất nhiều cảnh hành động sẽ nhận được taxi đến đích của nó, từ đó đạt được mục tiêu, nhưng một số là nhanh hơn, an toàn hơn, đáng tin cậy hơn, rẻ hơn so với những người khác. Mục tiêu chỉ cần cung cấp một sự phân biệt giữa "hạnh phúc" và "không hài lòng" , trong khi đó một thước đo hiệu suất tổng quát hơn nên cho phép một so sánh của các quốc gia khác nhau trên thế giới (hoặc trình tự của các quốc gia) theo họ sẽ làm thế nào nếu họ có thể được hạnh phúc. Bởi vì "hạnh phúc" không phải là thuật ngữ khoa học, các thuật ngữ thông thường là để nói rằng nếu một trạng thái thế giới được ưa thích khác, sau đó nó có tiện ích cao cho các agent. Do đó, tiện ích là một chức năng mà map các trạng thái vào một số thực, trong đó mô tả mức độ liên quan của hạnh phúc. Một đặc điểm kỹ thuật đầy đủ các chức năng tiện ích cho phép các quyết định hợp lý trong hai loại trường hợp mục tiêu gặp khó khăn.Đầu tiên, khi có những mục tiêu mâu thuẫn nhau, chỉ có một số trong đó có thể được đạt được (ví dụ, tốc độ và an toàn), chức năng tiện ích xác định thích hợp trade-off. Thứ hai, khi có một số mục tiêu mà các agent có thể nhằm mục đích, không ai trong số đó có thể đạt được một cách chắc chắn, tiện ích cung cấp một cách thức mà khả năng thành công có thể được cân nhắc lên chống lại tầm quan trọng của các mục tiêu.

Trong chương 16, chúng ta thấy rằng bất kỳ agent hợp lý có thể được mô tả như có một chức năng tiện ích.Một agent sở hữu một chức năng tiện ích rõ ràng do đó có thể đưa ra các quyết định hợp lý, nhưng có thể có để so sánh các tiện ích đạt được của các khóa học khác nhau của các hành động. Mục tiêu, mặc dù cruder, cho phép các agent để chọn một hành động ngay lập tức nếu nó đáp ứng các mục tiêu.Trong một số trường hợp, hơn nữa, một chức năng hữu ích có thể được dịch sang một tập hợp các mục tiêu, như vậy mà các quyết định được thực hiện bởi một agent dựa trên mục tiêu bằng cách sử dụng những mục tiêu giống hệt với những người được thực hiện bởi các agent dựa trên tiện ích. Tiện ích agent dựa trên cấu trúc tổng thể xuất hiện trong hình 2.12.Tiện ích dựa trên thực tế các chương trình agent xuất hiện ở Chương 5, nơi mà chúng tôi kiểm tra các chương trình chơi game mà phải làm cho sự phân biệt tốt giữa các vị trí ban khác nhau, và trong Chương 17, nơi mà chúng ta giải quyết những vấn đề chung của thiết kế ra quyết định agents.

Hình 2.12