# I. Artificial Intelligence

## 1. Introduction - Giới thiệu

Chương này định nghĩa AI và thiết lập nền tảng văn hóa mà nó đã phát triển. Một số điểm quan trọng như sau:

* Quan điểm được thông qua là trí thông minh chủ yếu xoay quanh hành động hợp lý, trong đó mục tiêu của tác nhân thông minh là thực hiện hành động tốt nhất có thể trong bất kỳ tình huống nào. Trọng tâm của nghiên cứu là phát triển các tác nhân thể hiện dạng trí thông minh này.

### Foundations of AI - Nền tảng của AI

* **Triết học:** Các nhà triết học từ năm 400 trước Công nguyên đã tưởng tượng ra mô hình trí tuệ nhân tạo bằng cách khám phá các quan điểm rằng tâm trí hoạt động tương tự như máy móc, hoạt động dựa trên kiến thức được lưu trữ bằng ngôn ngữ bên trong và nhận thức đó có thể được sử dụng để đưa ra quyết định về các hành động cần thực hiện.
* **Toán học:** Các nhà toán học đã cung cấp các công cụ để xử lý các phát biểu có tính logic chắc chắn cũng như các phát biểu mang tính xác suất, không chắc chắn. Họ cũng đặt nền tảng cho việc hiểu các tính toán và suy luận về các thuật toán.
* **Kinh tế học:** Các nhà kinh tế học đã chính thức hóa vấn đề về việc đưa ra quyết định nhằm tối đa hóa kết quả mong đợi cho người ra quyết định.
* **Khoa học thần kinh:** Các nhà khoa học thần kinh đã khám phá ra một số sự thật về cách thức hoạt động của bộ não cũng như những điểm giống và khác nhau của nó với máy tính.
* **Tâm lý học:** Các nhà tâm lý học đã áp dụng ý tưởng rằng con người và động vật có thể được coi là những cỗ máy xử lý thông tin. Các nhà ngôn ngữ học cho thấy việc sử dụng ngôn ngữ phù hợp với mô hình này.
* **Kỹ thuật máy tính:** Các kỹ sư máy tính đã cung cấp những cỗ máy ngày càng mạnh mẽ để việc tạo ra các ứng dụng AI trở nên khả thi.
* **Lý thuyết điều khiển và điều khiển học:** Lý thuyết điều khiển liên quan đến việc thiết kế các thiết bị có thể hoạt động tối ưu dựa trên phản hồi từ môi trường. Mặc dù các công cụ toán học của lý thuyết điều khiển và AI khá khác nhau nhưng các lĩnh vực này đang tiến gần nhau hơn.

### History of AI & State of the Art - Lịch sử AI & Công nghệ tiên tiến

* Lịch sử của AI đã có những chu kỳ thành công, sự lạc quan không đúng chỗ và dẫn đến sự mất hứng thú và cắt giảm nguồn tài trợ. Cũng đã có những chu kỳ giới thiệu các phương pháp tiếp cận sáng tạo mới và cải tiến một cách có hệ thống những phương pháp tốt nhất.
* AI đã phát triển nhanh hơn trong thập kỷ qua nhờ việc sử dụng nhiều hơn các phương pháp khoa học trong việc thử nghiệm và so sánh các phương pháp tiếp cận.
* Những tiến bộ gần đây trong việc tìm hiểu cơ sở lý thuyết của trí thông minh đã đi đôi với những cải tiến về khả năng của các hệ thống thực. Các lĩnh vực con của AI đã trở nên tích hợp hơn và AI đã tìm thấy điểm chung với các ngành khác.

## 2. Intelligent Agents - Tác nhân tri thức

Chương này giống như một chuyến tham quan ngắn gọn về AI, thứ mà chúng ta coi là khoa học về thiết kế tác nhân. Những điểm chính cần nhớ lại như sau:

### Agents and environments - Tác nhân và môi trường

* Tác nhân là thứ có nhận thức và hành động trong một môi trường. Hàm tác nhân chỉ định hành động được thực hiện bởi tác nhân để đáp ứng với bất kỳ chuỗi nhận thức nào.

### The concept of Rationality - Khái niệm về tính hợp lý

* Thước đo hiệu suất đánh giá hành vi của tác nhân trong một môi trường. Một tác nhân hợp lý hoạt động để tối đa hóa giá trị mong đợi của thước đo hiệu suất dựa trên chuỗi nhận thức mà nó gặp phải.

### The nature of environments - Bản chất của môi trường

* Đặc tả môi trường nhiệm vụ bao gồm thước đo hiệu suất, môi trường bên ngoài, bộ truyền động và cảm biến (PEAS). Khi thiết kế một tác nhân, bước đầu tiên phải luôn là xác định môi trường tác vụ một cách đầy đủ nhất có thể.
* Môi trường nhiệm vụ khác nhau theo một số khía cạnh quan trọng. Chúng có thể được quan sát đầy đủ hoặc một phần, một tác nhân hoặc đa tác nhân, xác định hoặc ngẫu nhiên, từng giai đoạn hoặc tuần tự, tĩnh hoặc động, rời rạc hoặc liên tục, và được biết hoặc chưa biết.

### The structure of agents - Cơ cấu tác nhân

* Chương trình tác nhân đưa chức năng tác nhân vào hoạt động. Có nhiều thiết kế chương trình tác nhân cơ bản khác nhau, phản ánh thông tin rõ ràng được sử dụng trong việc ra quyết định. Những thiết kế này khác nhau về hiệu quả, sự nhỏ gọn và tính linh hoạt. Việc thiết kế chương trình tác nhân phù hợp phụ thuộc vào bản chất của môi trường.
* Các tác nhân phản xạ đơn giản phản ứng trực tiếp với các nhận thức, trong khi các tác nhân phản xạ dựa trên mô hình duy trì trạng thái bên trong để theo dõi các khía cạnh của thế giới không hiển nhiên trong nhận thức hiện tại. Các tác nhân dựa trên mục tiêu hành động để đạt được mục tiêu của mình và các tác nhân dựa trên tiện ích nhằm mục đích tối đa hóa “hạnh phúc” mong đợi của chính nó.
* Tất cả các tác nhân có thể cải thiện hiệu suất thông qua học tập.

# II. Problem-solving

## 3. Solving problems by searching - Giải quyết vấn đề bằng cách tìm kiếm

Chương này đã giới thiệu các phương pháp mà một tác nhân có thể sử dụng để chọn các hành động trong các môi trường xác định, có thể quan sát được, tĩnh và hoàn toàn được biết đến. Trong những trường hợp như vậy, tác nhân có thể xây dựng chuỗi hành động để đạt được mục tiêu của mình; quá trình này được gọi là tìm kiếm.

### Problem-Solving Agents - Tác nhân giải quyết vấn đề

* Trước khi một tác nhân có thể bắt đầu tìm kiếm giải pháp, mục tiêu phải được xác định và vấn đề được xác định rõ ràng phải được hình thành.Trước khi một tác nhân có thể bắt đầu tìm kiếm giải pháp, mục tiêu phải được xác định và vấn đề được xác định rõ ràng phải được hình thành.

### Well-defined problems and solutions - Các vấn đề và giải pháp được xác định rõ ràng

* Một bài toán gồm có 5 phần:
  1. Trạng thái ban đầu
  2. Một tập hợp các hành động
  3. Mô hình chuyển đổi mô tả kết quả của những hành động đó
  4. Chức năng kiểm tra mục tiêu
  5. Hàm chi phí đường dẫn.
* Môi trường của bài toán được biểu diễn bằng không gian trạng thái.
* Đường đi trong không gian trạng thái từ trạng thái ban đầu đến trạng thái đích là một nghiệm.

### Searching for solutions - Tìm kiếm giải pháp

* Các thuật toán tìm kiếm coi các trạng thái và hành động là nguyên tử: chúng không xem xét bất kỳ cấu trúc bên trong nào mà chúng có thể sở hữu.
* Thuật toán TREE-SEARCH tổng quát xem xét tất cả các đường dẫn có thể để tìm ra giải pháp, trong khi thuật toán GRAPH-SEARCH tránh xem xét các đường dẫn dư thừa.
* Các thuật toán tìm kiếm được đánh giá dựa trên tính đầy đủ, tối ưu, độ phức tạp về thời gian và độ phức tạp về không gian. Độ phức tạp phụ thuộc vào b, hệ số phân nhánh trong không gian trạng thái và d, độ sâu của nghiệm nông nhất.
* Các phương pháp tìm kiếm không chính xác chỉ có quyền truy cập vào định nghĩa vấn đề. Các thuật toán cơ bản như sau:
  + **Tìm kiếm theo chiều rộng** sẽ mở rộng các nút nông nhất trước tiên; nó hoàn chỉnh, tối ưu cho chi phí bước đơn vị, nhưng có độ phức tạp về không gian theo cấp số nhân.
  + **Tìm kiếm với chi phí đồng nhất** mở rộng nút có chi phí đường đi thấp nhất, g(n) và tối ưu cho chi phí bước chung.
  + **Tìm kiếm theo chiều sâu** sẽ mở rộng nút chưa được mở rộng sâu nhất trước tiên. Nó không hoàn chỉnh cũng như không tối ưu nhưng có độ phức tạp về không gian tuyến tính. Tìm kiếm giới hạn độ sâu thêm giới hạn độ sâu.
  + **Tìm kiếm sâu hơn** lặp lại gọi tìm kiếm theo chiều sâu với giới hạn độ sâu tăng dần cho đến khi tìm thấy mục tiêu. Nó hoàn chỉnh, tối ưu cho chi phí bước đơn vị, có độ phức tạp về thời gian tương đương với tìm kiếm theo chiều rộng và có độ phức tạp về không gian tuyến tính.
  + **Tìm kiếm hai chiều** có thể làm giảm đáng kể độ phức tạp về thời gian, nhưng nó không phải lúc nào cũng có thể áp dụng được và có thể cần quá nhiều không gian.
* Các phương pháp tìm kiếm có hiểu biết có thể có quyền truy cập vào hàm heuristic h(n) ước tính chi phí của giải pháp từ n.
  + **Thuật toán tìm kiếm đầu tiên tốt nhất** chung sẽ chọn một nút để mở rộng theo hàm đánh giá.
  + **Greedy best-first search** mở rộng các nút với h(n) tối thiểu. Nó không tối ưu nhưng thường hiệu quả.
  + **Tìm kiếm A∗** mở rộng các nút có giá trị tối thiểu f(n) = g(n) + h(n). A∗ là đầy đủ và tối ưu, miễn là h(n) có thể chấp nhận được (đối với TÌM KIẾM CÂY) hoặc nhất quán (đối với TÌM KIẾM HÌNH THỨC). Độ phức tạp về không gian của A∗ vẫn còn rất hạn chế.
  + **RBFS (tìm kiếm đệ quy tốt nhất đầu tiên) và SMA∗ (A∗ giới hạn bộ nhớ đơn giản)** là các thuật toán tìm kiếm tối ưu, mạnh mẽ sử dụng lượng bộ nhớ hạn chế; nếu có đủ thời gian, nó có thể giải quyết các vấn đề mà A∗ không thể giải quyết được vì hết bộ nhớ.
* Hiệu suất của các thuật toán tìm kiếm heuristic phụ thuộc vào chất lượng của hàm heuristic. Đôi khi người ta có thể xây dựng các phương pháp phỏng đoán tốt bằng cách nới lỏng định nghĩa vấn đề, bằng cách lưu trữ chi phí giải pháp được tính toán trước cho các bài toán con trong cơ sở dữ liệu mẫu hoặc bằng cách học hỏi kinh nghiệm với lớp vấn đề.

**Nhóm 1: chương 7 (logical Agent) và chương 8 (First-order logic)**

**chương 7 (logical Agent)**

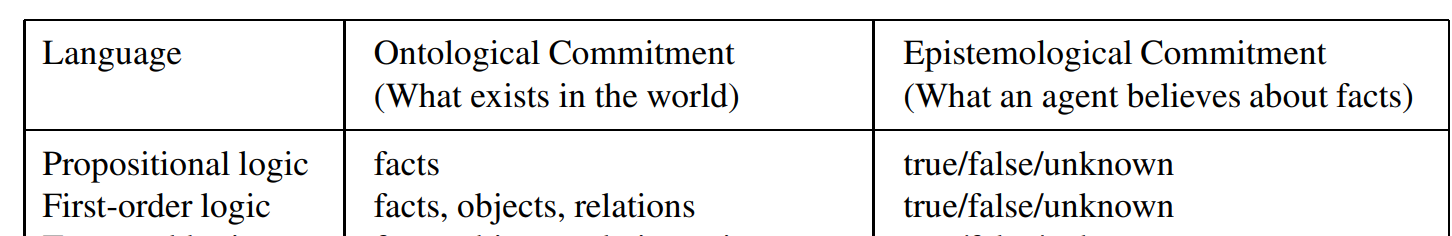
We have introduced knowledge-based agents and have shown how to define a logic with which such agents can reason about the world. The main points are as follows:

* Intelligent agents need knowledge about the world in order to reach good decisions.
* knowledge is contained in agents in the form of **sentences** in a **knowledge representation** language that are stored in a **knowledge base**.
* **A knowledge-based agent** is composed of a knowledge base and an inference mechanism. It operates by storing sentences about the world in its knowledge base, using the inference mechanism to infer new sentences, and using these sentences to decide what action to take.
* **A representation language** is defined by its **syntax**, which specifies the structure of sentences, and its **semantics**, which defines the **truth** of each sentence in each **possible world** or **model**.
* The relationship of **entailment** between sentences is crucial to our understanding of reasoning. A sentence α entails another sentence β if β is true in all worlds where α is true. Equivalent definitions include the **validity** of the sentence α ⇒ β and the **unsatisfiability** of the sentence α ∧ ¬β.
* Inference is the process of deriving new sentences from old ones. **Sound** inference algorithms derive only sentences that are entailed; **complete** algorithms derive all sentences that are entailed.
* **Propositional logic** is a simple language consisting of **proposition symbols** and **logical connectives**. It can handle propositions that are known true, known false, or completely unknown.
* The set of possible models, given a fixed propositional vocabulary, is finite, so entailment can be checked by enumerating models. Efficient **model-checking** inference algorithms for propositional logic include backtracking and local search methods and can often solve large problems quickly.
  + **Inference rules** are patterns of sound inference that can be used to find proofs. **The resolution rule** yields a complete inference algorithm for knowledge bases that are expressed in **conjunctive normal form**. **Forward chaining** and **backward chaining** are very natural reasoning algorithms for knowledge bases in **Horn form**
  + **Local search** methods such as WALKSAT can be used to find solutions. Such algorithms are sound but not complete.
  + Logical **state estimation** involves maintaining a logical sentence that describes the set of possible states consistent with the observation history. Each update step requires inference using the transition model of the environment, which is built from **successor-state axioms** that specify how each **fluent** changes.
  + Decisions within a logical agent can be made by SAT solving: finding possible models specifying future action sequences that reach the goal. This approach works only for fully observable or sensorless environments
  + Propositional logic does not scale to environments of unbounded size because it lacks the expressive power to deal concisely with time, space, and universal patterns of relationships among objects.

**chương 8 (First-order logic)**

This chapter has introduced **first-order logic**, a representation language that is far more powerful than propositional logic. The important points are as follows:

* Knowledge representation languages should be declarative, compositional, expressive, context independent, and unambiguous.
* Logics differ in their **ontological commitments** and **epistemological commitments**. While propositional logic commits only to the existence of facts, first-order logic commits to the existence of objects and relations and thereby gains expressive power
* The syntax of first-order logic builds on that of propositional logic. It adds terms to represent objects, and has universal and existential quantifiers to construct assertions about all or some of the possible values of the quantified variables.
* A **possible world, or model**, for first-order logic includes a set of objects and an **interpretation** that maps constant symbols to objects, predicate symbols to relations among objects, and function symbols to functions on objects.
* An atomic sentence is true just when the relation named by the predicate holds between the objects named by the terms. **Extended interpretations**, which map quantifier variables to objects in the model, define the truth of quantified sentences.
* Developing a knowledge base in first-order logic requires a careful process of analyzing the domain, choosing a vocabulary, and encoding the axioms required to support the desired inferences

****

[First-order logic là một hệ thống hình thức được sử dụng trong toán học, triết học, ngôn ngữ học và khoa học máy tính1](https://en.wikipedia.org/wiki/First-order_logic). [First-order logic sử dụng các biến lượng tử trên các đối tượng phi logic và cho phép sử dụng các câu chứa biến, để thay vì các mệnh đề như “Socrates là một người đàn ông”, ta có thể có các biểu thức dạng “tồn tại x sao cho x là Socrates và x là một người đàn ông”, trong đó “tồn tại” là một biến lượng tử, trong khi x là một biến 1](https://en.wikipedia.org/wiki/First-order_logic). Điều này phân biệt nó với logic mệnh đề, không sử dụng biến lượng tử hoặc quan hệ; trong ý nghĩa này, logic mệnh đề là nền tảng của first-order logic.

[**Ontological commitments** là các giả định về sự tồn tại của các thực thể, đối tượng hoặc hiện tượng trong thế giới 1](https://link.springer.com/chapter/10.1007/978-94-010-1186-0_5). [Ví dụ, một nhà triết học có thể có ontological commitments về sự tồn tại của các đối tượng trừu tượng như tư duy, ý niệm hoặc cảm xúc](https://link.springer.com/chapter/10.1007/978-94-010-1186-0_5" \t "_blank)[1](https://link.springer.com/chapter/10.1007/978-94-010-1186-0_5).

[**Epistemological commitments** là các giả định về kiến thức và cách tiếp cận của một hệ thống logic 2](https://www.frontiersin.org/articles/10.3389/frwa.2022.1038322/full). [Ví dụ, một hệ thống logic có thể giả định rằng kiến thức được xác định bởi các quy tắc suy luận, trong khi một hệ thống khác có thể giả định rằng kiến thức được xác định bởi các quan sát thực tế](https://www.frontiersin.org/articles/10.3389/frwa.2022.1038322/full" \t "_blank)[2](https://www.frontiersin.org/articles/10.3389/frwa.2022.1038322/full).

Khác nhau giữa hai khái niệm này là ontological commitments liên quan đến sự tồn tại của các thực thể, đối tượng hoặc hiện tượng trong thế giới, trong khi epistemological commitments liên quan đến kiến thức và cách tiếp cận của một hệ thống logic.

**Nhóm 2: chương 9 (Inference in first order logic) và chương 10 (Classical Planning)**

**chương 9 (Inference in first order logic)**

We have presented an analysis of logical inference in first-order logic and a number of algorithms for doing it.

* A first approach uses inference rules (**universal instantiation** and **existential instantiation**) to **propositionalize** the inference problem. Typically, this approach is slow, unless the domain is small.
* The use of **unification** to identify appropriate substitutions for variables eliminates the instantiation step in first-order proofs, making the process more efficient in many cases.
* A lifted version of **Modus Ponens** uses unification to provide a natural and powerful inference rule, **generalized Modus Ponens.** The **forward-chaining** and **backward-chaining** algorithms apply this rule to sets of definite clauses.
* Generalized Modus Ponens is complete for definite clauses, although the entailment problem is **semidecidable**. For **Datalog** knowledge bases consisting of function-free definite clauses, entailment is decidable.
* Forward chaining is used in **deductive databases**, where it can be combined with relational database operations. It is also used in **production systems**, which perform efficient updates with very large rule sets. Forward chaining is complete for Datalog and runs in polynomial time.
* Backward chaining is used in logic programming systems, which employ sophisticated compiler technology to provide very fast inference. Backward chaining suffers from redundant inferences and infinite loops; these can be alleviated by **memoization**.
* Prolog, unlike first-order logic, uses a closed world with the unique names assumption and negation as failure. These make Prolog a more practical programming language, but bring it further from pure logic.
* The generalized **resolution** inference rule provides a complete proof system for firstorder logic, using knowledge bases in conjunctive normal form.
* Several strategies exist for reducing the search space of a resolution system without compromising completeness. One of the most important issues is dealing with equality; we showed how **demodulation** and **paramodulation** can be used.
* Efficient resolution-based theorem provers have been used to prove interesting mathematical theorems and to verify and synthesize software and hardware.

[**Universal instantiation** (UI) và **existential instantiation** (EI) là hai quy tắc suy luận trong **first-order logic** (FOL)](https://carnap.io/book/12)[1](https://carnap.io/book/12).

**Universal instantiation** (UI) còn được gọi là **universal elimination**. [Nó cho phép suy ra bất kỳ trường hợp cụ thể nào của một tổng quát về một đối tượng](https://carnap.io/book/12" \t "_blank)[1](https://carnap.io/book/12). [Ví dụ, nếu chúng ta có một tổng quát như “Tất cả các con vật đều có thể di chuyển”, chúng ta có thể sử dụng UI để suy ra các trường hợp cụ thể như “Chó có thể di chuyển” hoặc “Mèo có thể di chuyển”](https://carnap.io/book/12" \t "_blank)[1](https://carnap.io/book/12).

**Existential instantiation** (EI) còn được gọi là **existential elimination**. [Nó cho phép suy ra một trường hợp cụ thể từ một câu khẳng định chứa lượng từ tồn tại](https://bing.com/search?q=universal+instantiation+and+existential+instantiation" \t "_blank)[2](https://bing.com/search?q=universal+instantiation+and+existential+instantiation). [Ví dụ, nếu chúng ta có một câu khẳng định như “Có một con vật có thể bay”, chúng ta có thể sử dụng EI để suy ra một trường hợp cụ thể như “Chim có thể bay”](https://bing.com/search?q=universal+instantiation+and+existential+instantiation" \t "_blank)[2](https://bing.com/search?q=universal+instantiation+and+existential+instantiation).

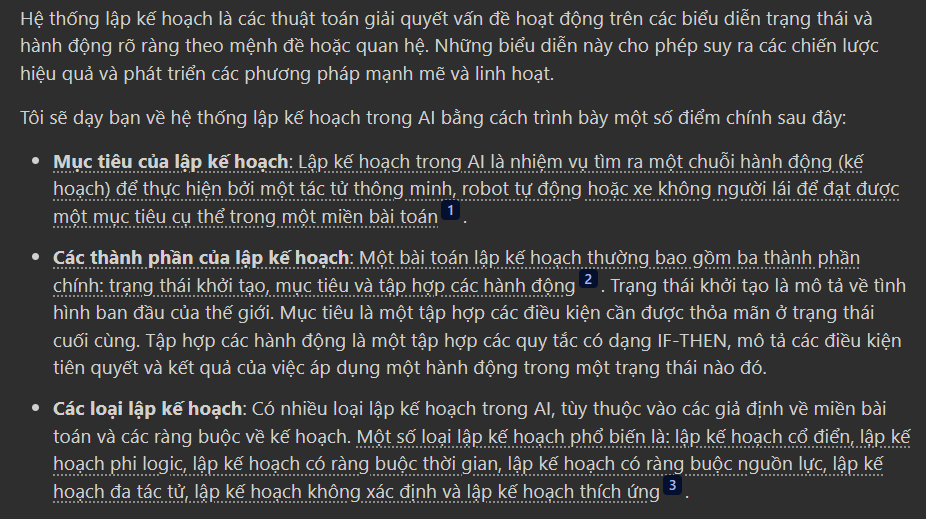
[**Generalized Modus Ponens** (GMP) là một phiên bản nâng cao của quy tắc suy luận Modus Ponens trong **first-order logic** (FOL) 1](https://engineeringinterviewquestions.com/mcqs-on-unification-and-lifting-and-answers/). [GMP sử dụng **unification** để cung cấp một quy tắc suy luận tự nhiên và mạnh mẽ hơn 1](https://engineeringinterviewquestions.com/mcqs-on-unification-and-lifting-and-answers/).

[Unification là quá trình giải quyết các phương trình chứa các biểu thức ký hiệu trong FOL 2](https://bing.com/search?q=lifted+version+of+Modus+Ponens+unification). Nếu cả hai bên của phương trình đều là hằng số, kết quả sẽ được giải quyết nếu chúng giống nhau. Nếu một bên là biến, chúng ta thêm thay thế của biến này bằng bên kia. [Nếu cả hai bên đều là hàm, chúng giống nhau nếu đó là cùng một hàm (cùng tên) và cùng số lượng tham số, sau đó giải quyết đệ quy các tham số](https://bing.com/search?q=lifted+version+of+Modus+Ponens+unification" \t "_blank)[2](https://bing.com/search?q=lifted+version+of+Modus+Ponens+unification).

**chương 10 (Classical Planning)**

In this chapter, we defined the problem of planning in deterministic, fully observable, static environments. We described the PDDL representation for planning problems and several algorithmic approaches for solving them. The points to remember:

* Planning systems are problem-solving algorithms that operate on explicit propositional or relational representations of states and actions. These representations make possible the derivation of effective heuristics and the development of powerful and flexible algorithms for solving problems.
* **PDDL, the Planning Domain Definition Language**, describes the initial and goal states as conjunctions of literals, and actions in terms of their preconditions and effects.
  + can operate in the forward direction (**progression**) or the backward direction (**regression**). Effective heuristics can be derived by subgoal independence assumptions and by various relaxations of the planning problem.
* A **planning graph** can be constructed incrementally, starting from the initial state. Each layer contains a superset of all the literals or actions that could occur at that time step and encodes mutual exclusion (mutex) relations among literals or actions that cannot cooccur. Planning graphs yield useful heuristics for state-space and partial-order planners and can be used directly in the GRAPHPLAN algorithm.
* Other approaches include first-order deduction over situation calculus axioms; encoding a planning problem as a Boolean satisfiability problem or as a constraint satisfaction problem; and explicitly searching through the space of partially ordered plans.
* Each of the major approaches to planning has its adherents, and there is as yet no consensus on which is best. Competition and cross-fertilization among the approaches have resulted in significant gains in efficiency for planning systems.

****

**Cấu Trúc Cơ Bản của PDDL:**

#### **1. Domain Definition:**

* Trong PDDL, "domain" là nơi mà bạn định nghĩa các yếu tố chung của bài toán lập kế hoạch.
* Nó bao gồm các phần như tên domain, các khai báo về các toán tử (hành động), các khai báo về các đối tượng, v.v.

#### **2. Objects:**

* Định nghĩa các đối tượng có thể xuất hiện trong bài toán.
* Ví dụ: Nếu bạn đang giải một bài toán vận chuyển, các đối tượng có thể là các thành phố, đồ vật, v.v.

#### **3. Predicates:**

* Mô tả các trạng thái có thể xuất hiện.
* Ví dụ: Trạng thái "On(x, y)" có thể mô tả đối tượng x đặt trên đối tượng y.

#### **4. Actions:**

* Định nghĩa các hành động có thể thực hiện trong bài toán.
* Mỗi hành động có thể có điều kiện tiên quyết (preconditions) và hiệu ứng (effects).

#### **5. Initial State:**

* Mô tả trạng thái ban đầu của bài toán.

#### **6. Goal State:**

* Mô tả trạng thái mà bạn muốn đạt được.

**Nhóm 3: chương 11 (Planning and acting in the real world ) và chương 12 (knowledge representation)**

**chương 11 (Planning and acting in the real world )**

This chapter has addressed some of the complications of planning and acting in the real world. The main points:

* **Many actions consume resources, such as money, gas, or raw materials. It is convenient to treat these resources as numeric measures in a pool rather than try to reason about, say, each individual coin and bill in the world. Actions can generate and consume resources, and it is usually cheap and effective to check partial plans for satisfaction of resource constraints before attempting further refinements.**
* is one of the most important resources. It can be handled by specialized scheduling algorithms, or scheduling can be integrated with planning.
* **Hierarchical task network (**HTN) planning allows the agent to take advice from the domain designer in the form of **high-level actions** (HLAs) that can be implemented in various ways by lower-level action sequences. The effects of HLAs can be defined with **angelic semantics**, allowing provably correct high-level plans to be derived without consideration of lower-level implementations. HTN methods can create the very large plans required by many real-world applications.
* Standard planning algorithms assume complete and correct information and deterministic, fully observable environments. Many domains violate this assumption.
* **Contingent plans** allow the agent to sense the world during execution to decide what branch of the plan to follow. In some cases, sensorless or conformant planning can be used to construct a plan that works without the need for perception. Both conformant and contingent plans can be constructed by search in the space of belief states. Efficient representation or computation of belief states is a key problem
* An **online planning agent** uses execution monitoring and splices in repairs as needed to recover from unexpected situations, which can be due to nondeterministic actions, exogenous events, or incorrect models of the environment
* **Multiagent** planning is necessary when there are other agents in the environment with which to cooperate or compete. Joint plans can be constructed, but must be augmented with some form of coordination if two agents are to agree on which joint plan to execute.
* This chapter extends classic planning to cover nondeterministic environments (where outcomes of actions are uncertain), but it is not the last word on planning. Chapter 17 describes techniques for stochastic environments (in which outcomes of actions have probabilities associated with them): Markov decision processes, partially observable Markov decision processes, and game theory. In Chapter 21 we show that reinforcement learning allows an agent to learn how to behave from past successes and failures.

Trong lập kế hoạch Hierarchical Task Network (HTN), "**contingent plans**" là một khái niệm liên quan đến việc xác định các bước thực hiện dựa trên điều kiện cụ thể hoặc tình huống mà hệ thống có thể gặp phải trong quá trình thực hiện một nhiệm vụ. Điều này làm cho kế hoạch trở nên linh hoạt và có thể thích ứng được với những thay đổi trong môi trường hoặc tình huống.

Một kế hoạch có thể được chia thành các phần có thể thay thế, được gọi là "contingent plans", để xử lý các điều kiện đặc biệt. Khi một điều kiện nào đó xuất hiện, hệ thống có thể chuyển từ một phần của kế hoạch sang một phần khác mà xử lý điều kiện đó. Điều này giúp đảm bảo rằng hệ thống có khả năng thích ứng với môi trường thực tế và có thể hoàn thành nhiệm vụ một cách hiệu quả trong các tình huống biến đổi.

Ví dụ, nếu bạn có một kế hoạch để đi từ điểm A đến B, một phần của kế hoạch có thể là đi bộ, và một phần khác có thể là lái xe. Nếu trời mưa, bạn có thể chọn phần của kế hoạch là lái xe thay vì đi bộ. Điều này là một ví dụ về cách một kế hoạch có thể có các phần có thể thay thế tùy thuộc vào điều kiện môi trường.

**Hierarchical Task Network (HTN) Planning là gì?**

**HTN Planning là một phương pháp trong trí tuệ nhân tạo mà người ta sử dụng để lập kế hoạch cho các hệ thống tự động thực hiện nhiệm vụ. Điều đặc biệt là nó cho phép người thiết kế (hoặc người lập kế hoạch) cung cấp các hành động ở mức độ cao, gọi là High-Level Actions (HLAs). Những hành động này có thể được thực hiện theo nhiều cách bởi các chuỗi hành động ở mức độ thấp hơn.**

**Vì sao HTN Planning quan trọng?**

* **Linh hoạt: HTN Planning mang lại tính linh hoạt, giúp người lập kế hoạch chỉ cần xác định những gì cần thực hiện ở mức độ cao, không cần quan tâm đến chi tiết cài đặt ở mức thấp.**
* **Hiệu suất: Hành động cấp cao có thể được thực hiện theo nhiều cách khác nhau, giúp tối ưu hóa sự linh hoạt trong quá trình lập kế hoạch.**
* **Độ chính xác: HTN Planning có thể xác định hiệu ứng của các hành động cấp cao một cách chính xác, giúp đảm bảo tính đúng đắn của kế hoạch.**

**Cách HTN Planning hoạt động:**

* **Hành động Ở Mức Độ Cao (HLAs): Người lập kế hoạch xác định các hành động ở mức độ cao, ví dụ như "di chuyển đến đích".**
* **Phân rã Cấp Độ Cao Thành Cấp Độ Thấp (Decomposition): Các HLA được phân rã thành chuỗi hành động ở mức độ thấp hơn, chẳng hạn như "bước chân trái, bước chân phải" để thực hiện "di chuyển đến đích".**
* **Ràng buộc và Quy tắc Thực hiện: Xác định các ràng buộc giữa các hành động và quy tắc thực hiện chúng.**
* **Lập kế hoạch: Hệ thống lập kế hoạch tự động tạo ra chuỗi hành động ở mức độ thấp hơn từ các HLA, đưa ra một kế hoạch để đạt được mục tiêu.**

**Ví dụ:**

**Giả sử bạn muốn robot nhà bạn làm nhiệm vụ "lấy một cốc nước từ bàn và đưa cho bạn". Trong HTN Planning, bạn chỉ cần nói robot cần thực hiện các hành động cao cấp như "điều khiển cánh tay", "lấy cốc từ bàn" và "đưa cốc cho bạn". Còn lại, robot tự quyết định cách thực hiện chi tiết từng bước.**

**"An online planning agent."**

**1. Định Nghĩa:**

* **Agent: Một tác nhân tự động có khả năng thực hiện quyết định để đạt được mục tiêu.**
* **Lập Kế Hoạch Trực Tuyến (Online Planning): Đây là quá trình mà agent đưa ra quyết định ngay lập tức, không chờ đến khi có đủ thông tin. Nó phải liên tục điều chỉnh kế hoạch của mình dựa trên thông tin mới và tình hình thay đổi.**

**2. Hoạt Động Cơ Bản:**

* **Quyết Định Nhanh Chóng: Agent phải đưa ra quyết định ngay lập tức mà không cần đến sự đầy đủ thông tin. Điều này thường xảy ra trong môi trường động và thay đổi.**
* **Tối Ưu Hóa Kế Hoạch: Agent phải liên tục tối ưu hóa kế hoạch của mình để đảm bảo nó phản ánh môi trường hiện tại.**

**3. Ví Dụ Áp Dụng:**

* **Robot Tự Lái Xe: Nếu bạn nghĩ về một hệ thống lái xe tự động, nó phải liên tục đưa ra quyết định về việc di chuyển, tránh vật cản, và thậm chí quyết định đường đi dựa trên tình hình giao thông thực tế.**
* **Game AI: Trong trò chơi video, các nhân vật thông minh cần có khả năng đưa ra quyết định nhanh chóng để phản ứng với hành động của người chơi.**

**4. Thách Thức và Chiến Lược:**

* **Thách Thức Tính Thời: Agent phải xử lý thông tin một cách nhanh chóng, đôi khi chỉ trong khoảng thời gian ngắn.**
* **Chiến Lược Điều Chỉnh: Agent cần có chiến lược linh hoạt để điều chỉnh kế hoạch dựa trên thông tin mới và tình hình thay đổi.**

**5. Công Nghệ Liên Quan:**

* **Học Máy: Các phương pháp học máy có thể được tích hợp để cải thiện khả năng quyết định của agent dựa trên dữ liệu và kinh nghiệm trước đó.**
* **Thuật Toán Lập Kế Hoạch: Sử dụng các thuật toán lập kế hoạch hiệu quả để đảm bảo rằng quá trình đưa ra quyết định là nhanh chóng và đáng tin cậy.**

**Multiagent Planning là gì?**

**Multiagent Planning (lập kế hoạch đa tác nhân) nghiên cứu cách nhiều tác nhân (như robot, hệ thống tự động, hoặc chương trình máy tính) có thể làm việc cùng nhau để đạt được các mục tiêu chung trong một môi trường chia sẻ.**

**Các Đặc Điểm Chính:**

* **Tác Nhân (Agent): Mỗi tác nhân là một đối tượng tự động có khả năng đưa ra quyết định và thực hiện hành động để đạt được mục tiêu.**
* **Tương Tác: Các tác nhân có thể tương tác với nhau, giao tiếp, và cần phải làm việc cùng nhau để đạt được mục tiêu chung.**
* **Mục Tiêu Cộng Đồng (Joint Goals): Có thể có mục tiêu riêng cho từng tác nhân, nhưng cũng có những mục tiêu chung mà tất cả đều cố gắng đạt được.**
* **Ràng Buộc Tương Tác (Interaction Constraints): Các quy luật và ràng buộc xác định cách mà các tác nhân có thể tương tác và làm việc cùng nhau.**
* **Lập Kế Hoạch Tương Tác (Interactive Planning): Là quá trình xây dựng kế hoạch trong khi còn đang tương tác với môi trường và các tác nhân khác.**

**Quy Trình Multiagent Planning:**

* **Mục Tiêu Xác Định (Goal Specification): Xác định những gì hệ thống đa tác nhân cần đạt được. Điều này có thể là một mục tiêu chung cho cả hệ thống hoặc các mục tiêu riêng lẻ cho từng tác nhân.**
* **Phân Tích Môi Trường (Environment Analysis): Đánh giá môi trường và các tác nhân trong đó để xác định tác động của chúng đối với nhau.**
* **Lập Kế Hoạch (Planning): Xây dựng kế hoạch cho từng tác nhân hoặc kế hoạch chung mà tất cả cần tuân theo để đạt được mục tiêu.**
* **Tương Tác và Thực Hiện Kế Hoạch (Interaction and Execution): Các tác nhân thực hiện kế hoạch của mình trong môi trường, tương tác với nhau và cập nhật kế hoạch nếu cần thiết.**
* **Đánh Giá và Tối Ưu Hóa (Evaluation and Optimization): Đánh giá hiệu suất của kế hoạch và điều chỉnh nếu cần thiết để tối ưu hóa đạt được mục tiêu.**

**Ứng Dụng:**

* **Robotic Swarm: Trong các đội robot (swarm), mỗi robot có thể là một tác nhân, và chúng cần phối hợp để di chuyển, thu thập dữ liệu, hoặc thậm chí là xây dựng cấu trúc**
* **Giao Thông Tự Động: Trong môi trường giao thông tự động, các xe có thể làm việc cùng nhau để giảm kẹt xe, cải thiện an toàn, và tối ưu hóa luồng giao thông.**

**Thách Thức:**

* **Tương Tác Phức Tạp: Điều chỉnh hành vi và kế hoạch của từng tác nhân để đảm bảo tương tác mượt mà và hiệu quả là một thách thức.**
* **Chia Sẻ Thông Tin: Cần phải có các cơ chế hiệu quả để chia sẻ thông tin giữa các tác nhân để họ có thể hiểu và phản ứng đúng đắn.**

**Tổng Kết:**

**Multiagent Planning là một lĩnh vực phức tạp nhưng quan trọng trong trí tuệ nhân tạo, mở rộng áp dụng từ các ứng dụng trong robotica đến quản lý tài nguyên và giao thông. Để hiểu rõ hơn, bạn có thể tìm hiểu về các thuật toán cụ thể, công cụ và nghiên cứu mới nhất trong lĩnh vực này.**

**chương 12 (knowledge representation)**

By delving into the details of how one represents a variety of knowledge, we hope we have given the reader a sense of how real knowledge bases are constructed and a feeling for the interesting philosophical issues that arise. The major points are as follows:

* Large-scale knowledge representation requires a general-purpose ontology to organize and tie together the various specific domains of knowledge.
* A general-purpose ontology needs to cover a wide variety of knowledge and should be capable, in principle, of handling any domain.
* Building a large, general-purpose ontology is a significant challenge that has yet to be fully realized, although current frameworks seem to be quite robust
* We presented an **upper ontology** based on categories and the event calculus. We covered categories, subcategories, parts, structured objects, measurements, substances, events, time and space, change, and beliefs.
* Natural kinds cannot be defined completely in logic, but properties of natural kinds can be represented.
* Actions, events, and time can be represented either in situation calculus or in more expressive representations such as event calculus. Such representations enable an agent to construct plans by logical inference.
* We presented a detailed analysis of the Internet shopping domain, exercising the general ontology and showing how the domain knowledge can be used by a shopping agent.
* Special-purpose representation systems, such as **semantic networks** and **description logics**, have been devised to help in organizing a hierarchy of categories. Inheritance is an important form of **inference**, allowing the properties of objects to be deduced from their membership in categories.
* The **closed-world assumption**, as implemented in logic programs, provides a simple way to avoid having to specify lots of negative information. It is best interpreted as a **default** that can be overridden by additional information
* **Nonmonotonic logics**, such as **circumscription** and **default logic**, are intended to capture default reasoning in general.
* **Truth maintenance systems** handle knowledge updates and revisions efficiently.

**Mạng Ngữ nghĩa (Semantic Networks):**

**Khái Niệm:**

* **Mạng Ngữ nghĩa là một phương pháp biểu diễn kiến thức trong trí tuệ nhân tạo.**
* **Nó sử dụng đồ thị hoặc mạng để mô tả các mối quan hệ giữa các khái niệm, đối tượng, sự kiện trong một hệ thống.**

**Thành Phần Chính:**

* **Đỉnh (Node): Đại diện cho các khái niệm hoặc đối tượng.**
* **Cạnh (Edge): Biểu thị mối quan hệ giữa các đỉnh.**

**Ưu Điểm:**

* **Dễ Hiểu: Mạng ngữ nghĩa thường dễ hiểu vì chúng tương tự như cách con người tổ chức thông tin.**
* **Biểu Diễn Mối Quan Hệ: Thích hợp cho việc biểu diễn mối quan hệ phức tạp giữa các khái niệm.**

**Hạn Chế:**

* **Khả Năng Biểu Diễn Hạn Chế: Không phải lúc nào cũng có thể biểu diễn mọi mối quan hệ phức tạp hiệu quả.**
* **Khó Khăn Trong Xử Lý Máy Tính: Có thể gặp khó khăn khi sử dụng cho máy tính xử lý vì cấu trúc đồ thị phức tạp.**

**Mô hình Logic Mô Tả (Description Logics):**

**Khái Niệm:**

* **Là một dạng ngôn ngữ logic được sử dụng để biểu diễn và tương tác với kiến thức về các đối tượng và mối quan hệ giữa chúng.**

**Thành Phần Chính:**

* **Khái Niệm Cơ Bản: Đại diện cho các khái niệm cơ bản.**
* **Thuộc Tính: Định nghĩa mối quan hệ giữa các khái niệm.**
* **Câu Lệnh (Assertion): Thông tin cụ thể về đối tượng và mối quan hệ.**

**Ưu Điểm:**

* **Tính Chính Xác: Sử dụng logic giúp mô tả chính xác mối quan hệ và thông tin.**
* **Biểu Diễn Phong Phú: Có khả năng biểu diễn mối quan hệ và ràng buộc phức tạp.**

**Hạn Chế:**

* **Khó Khăn Trong Hiểu Định và Sử Dụng: Đôi khi có thể khá phức tạp và khó hiểu đối với người không chuyên gia.**
* **Xử Lý Máy Tính Đòi Hỏi Nhiều Tài Nguyên: Việc xử lý mô hình logic mô tả đôi khi đòi hỏi nhiều tài nguyên tính toán.**

**Tóm Lược:**

* **Mạng ngữ nghĩa tập trung vào mối quan hệ và cấu trúc mạng để biểu diễn kiến thức.**
* **Mô hình logic mô tả sử dụng logic và các quy tắc để biểu diễn mối quan hệ và ràng buộc chính xác hơn.**
* **Cả hai phương pháp đều có ưu và nhược điểm, và lựa chọn giữa chúng thường phụ thuộc vào bối cảnh cụ thể và mục tiêu sử dụng trong lĩnh vực Trí Tuệ Nhân Tạo.**

**(Closed-World Assumption - CWA)**

**Giả Định Thế Giới Đóng Cửa trong Chương Trình Logic:**

* **Khi sử dụng trong logic lập trình, Giả định Thế giới Đóng Cửa (CWA) mang lại một cách tiện lợi để tránh việc phải chỉ định nhiều thông tin tiêu cực, đặc biệt là khi chúng ta có kiến thức giới hạn về một hệ thống.**

**Cách Thức Triển Khai:**

* **Trong logic lập trình, khi chúng ta không có thông tin nào chứng minh một sự kiện, trạng thái hoặc điều gì đó, ta giả định rằng nó là sai. CWA được hiểu như một giả định mặc định về việc sự vắng mặt thông tin nên được xem là phủ định.**

**Tránh Đặc Tả Nhiều Thông Tin Tiêu Cực:**

* **Thay vì phải liệt kê tất cả các điều không xảy ra, CWA cho phép giả sử rằng mọi điều không được biểu diễn là không đúng. Điều này giúp giảm độ phức tạp và mức độ chi tiết cần thiết trong việc mô tả kiến thức.**

**Giải Thích Như Một Mặc Định Có Thể Bị Ghi Đè:**

* **CWA có thể được hiểu như một mặc định, nhưng có thể bị ghi đè bởi thông tin bổ sung sau này. Nếu có thêm thông tin được cung cấp, chúng ta có thể thay đổi hoặc mở rộng kiến thức của mình.**

**Ví Dụ Minh Họa:**

* **Giả sử bạn có một hệ thống đặt vé máy bay. Nếu hệ thống không thông báo rằng một chuyến bay đã hủy, bạn có thể giả định rằng chuyến bay đó vẫn diễn ra (CWA). Tuy nhiên, nếu bạn có thông tin cụ thể về việc hủy chuyến bay đó, bạn có thể ghi đè lên giả định mặc định bằng cách cung cấp thông tin mới.**

**Tóm Lược:**

* **Giả định Thế giới Đóng Cửa trong chương trình logic cung cấp một giải pháp đơn giản để xử lý thông tin tiêu cực trong môi trường có kiến thức giới hạn. Nó được hiểu như một giả định mặc định có thể thay đổi hoặc bổ sung bằng thông tin thêm vào sau này.**

**Logic Phi Tăng (Nonmonotonic Logic)**

**1. Định nghĩa:**

* **Logic phi tăng là một loại logic trong trí tuệ nhân tạo được thiết kế để xử lý tình huống mà kiến thức của chúng ta có thể thay đổi khi chúng ta nhận thêm thông tin mà không nhất thiết phải loại bỏ thông tin đã có.**

**2. Tính Chất Chính:**

* **Không Monoton: Trái ngược với các hệ thống logic monotonic, logic phi tăng không tuân theo nguyên tắc "monotonicity," tức là thêm vào kiến thức mới có thể dẫn đến sự thay đổi, mở rộng hoặc thậm chí là thu hẹp của kiến thức đã có.**

**Circumscription (Điều Kiện Hạn Chế):**

**Đặc Điểm Chính:**

* **Mục Đích Chính: Được thiết kế để giảm thiểu sự không chắc chắn trong lập luận mặc định.**
* **Nguyên Lý: Định rõ một tập hợp các giả định mặc định và cố gắng giảm thiểu sự mâu thuẫn bằng cách giới hạn (circumscribe) miền giả định cho một số ý định.**
* **Ứng Dụng: Phổ biến trong lĩnh vực trí tuệ nhân tạo để mô hình hóa tư duy mặc định.**

**Ví Dụ Minh Họa:**

* **Nếu chúng ta biết "Tất cả chim thiên nga đều trắng," nhưng không biết gì về những loại chim khác, thì circumscription có thể giả sử "Tất cả những loại chim ngoại trừ thiên nga có thể là bất kỳ màu nào." Điều này giảm bớt sự mâu thuẫn khi chúng ta biết rằng có những thiên nga đen.**

**Default Logic (Logic Mặc Định):**

**Đặc Điểm Chính:**

* **Mục Đích Chính: Xử lý vấn đề khi thông tin bị thiếu.**
* **Nguyên Lý: Sử dụng các quy tắc mặc định để mô tả giả định có thể được áp dụng nếu không có thông tin cụ thể nào được biết đến.**
* **Ứng Dụng: Được sử dụng trong các hệ thống chấp nhận sự không chắc chắn và thiếu thông tin.**

**Ví Dụ Minh Họa:**

* **Nếu chúng ta biết "John thường đi làm bằng xe buýt," nhưng không có thông tin cụ thể về việc John có ô tô hay không, một quy tắc mặc định có thể là "John có ô tô." Điều này có thể được sử dụng khi không có chứng cứ chứng minh hoặc phản minh điều ngược lại.**

**Tóm Lược:**

* **Cả Circumscription và Default Logic đều là phương pháp trong Logic Phi Tăng (Nonmonotonic Logic) để xử lý tình huống khi kiến thức có thể thay đổi mà không cần phải loại bỏ thông tin đã có. Circumscription tập trung vào giảm thiểu mâu thuẫn bằng cách định rõ giả định mặc định, trong khi Default Logic tập trung vào xử lý thông tin thiếu bằng cách sử dụng các quy tắc mặc định. Cả hai đều có ứng dụng trong việc mô hình hóa tư duy mặc định trong lĩnh vực trí tuệ nhân tạo.**

**Truth Maintenance Systems (TMS) là một phương tiện quan trọng trong trí tuệ nhân tạo để duy trì sự nhất quán và theo dõi sự thay đổi trong kiến thức trong các môi trường động và không chắc chắn. Chúng đóng vai trò quan trọng trong việc duy trì thông tin nhất quán trong khi hệ thống phải đối mặt với sự thay đổi và cập nhật liên tục.**

**Nhóm 4: chương 13 (quantifying uncertainty) và chương 14 (Probabilistic reasoning)**

**chương 13 (quantifying uncertainty)**

This chapter has suggested probability theory as a suitable foundation for uncertain reasoning and provided a gentle introduction to its use.

* Uncertainty arises because of both laziness and ignorance. It is inescapable in complex, nondeterministic, or partially observable environments.
* Probabilities express the agent’s inability to reach a definite decision regarding the truth of a sentence. Probabilities summarize the agent’s beliefs relative to the evidence.
* Decision theory combines the agent’s beliefs and desires, defining the best action as the one that maximizes expected utility
* Basic probability statements include **prior probabilities** and **conditional probabilities** over simple and complex propositions.
* The axioms of probability constrain the possible assignments of probabilities to propositions. An agent that violates the axioms must behave irrationally in some cases.
* **The full joint probability** **distribution** specifies the probability of each complete assignment of values to random variables. It is usually too large to create or use in its explicit form, but when it is available it can be used to answer queries simply by adding up entries for the possible worlds corresponding to the query propositions.
* **Absolute** **independence** between subsets of random variables allows the full joint distribution to be factored into smaller joint distributions, greatly reducing its complexity. Absolute independence seldom occurs in practice.
* **Bayes’ rule** allows unknown probabilities to be computed from known conditional probabilities, usually in the causal direction. Applying Bayes’ rule with many pieces of evidence runs into the same scaling problems as does the full joint distribution.
* **Conditional independence** brought about by direct causal relationships in the domain might allow the full joint distribution to be factored into smaller, conditional distributions. The **naive Bayes** model assumes the conditional independence of all effect variables, given a single cause variable, and grows linearly with the number of effects.
* A wumpus-world agent can calculate probabilities for unobserved aspects of the world, thereby improving on the decisions of a purely logical agent. Conditional independence makes these calculations tractable.

**Xác suất Tiên nghiệm (Prior Probabilities):**

* **Xác suất tiên nghiệm là xác suất của một sự kiện trước khi có bất kỳ thông tin thực nghiệm nào được thu thập.**
* **Đây là cách chúng ta biểu diễn sự tin tưởng ban đầu của mình về xác suất của một sự kiện, thường dựa trên kiến thức trước đó hoặc suy luận.**

**Xác suất Có điều kiện (Conditional Probabilities):**

* **Xác suất có điều kiện là xác suất của một sự kiện A xảy ra dưới điều kiện đã biết một sự kiện B đã xảy ra.**
* **Nó thường được sử dụng để mô tả mức độ phụ thuộc giữa các sự kiện.**

**Conditional independence** (độc lập có điều kiện) là một khái niệm trong xác suất, miêu tả mức độ độc lập giữa hai biến ngẫu nhiên khi biết một biến ngẫu nhiên thứ ba. Nó đặc biệt quan trọng trong việc mô hình hóa mối quan hệ giữa các biến trong các mô hình xác suất.

**Bayesian modeling** is a statistical model where probability is influenced by the belief of the likelihood of a certain outcome

**chương 14 (Probabilistic reasoning)**

This chapter has described **Bayesian networks**, a well-developed representation for uncertain knowledge. Bayesian networks play a role roughly analogous to that of propositional logic for definite knowledge.

* A Bayesian network is a directed acyclic graph whose nodes correspond to random variables; each node has a conditional distribution for the node, given its parents.
* Bayesian networks provide a concise way to represent **conditional independence** relationships in the domain.
* A Bayesian network specifies a full joint distribution; each joint entry is defined as the product of the corresponding entries in the local conditional distributions. A Bayesian network is often exponentially smaller than an explicitly enumerated joint distribution.
* Many conditional distributions can be represented compactly by canonical families distributions. **Hybrid Bayesian networks**, which include both discrete and continuous variables, use a variety of canonical distributions
* Inference in Bayesian networks means computing the probability distribution of a set of query variables, given a set of evidence variables. Exact inference algorithms, such as variable elimination, evaluate sums of products of conditional probabilities as efficiently as possible.
* In polytrees (singly connected networks), exact inference takes time linear in the size of the network. In the general case, the problem is intractable.
* Stochastic approximation techniques such as **likelihood weighting** **and Markov chain** **Monte Carlo** can give reasonable estimates of the true posterior probabilities in a network and can cope with much larger networks than can exact algorithms.
* Probability theory can be combined with representational ideas from first-order logic to produce very powerful systems for reasoning under uncertainty. **Relational probability** **models** (RPMs) include representational restrictions that guarantee a well-defined probability distribution that can be expressed as an equivalent Bayesian network. **Open universe probability models** handle **existence** **and identity uncertainty**, defining probability distributions over the infinite space of first-order possible worlds.
* Various alternative systems for reasoning under uncertainty have been suggested. Generally speaking, **truth-functional** systems are not well suited for such reasoning

**Mạng Bayesian** là một mô hình xác suất đồ thị sử dụng suy diễn Bayesian để tính toán xác suất. Mục đích của mạng Bayesian là mô hình hóa các phụ thuộc có điều kiện của các biến ngẫu nhiên bằng cách biểu diễn các phụ thuộc đó dưới dạng các cạnh của đồ thị 1. Mạng Bayesian có những đặc điểm đặc biệt sau:

* Mạng Bayesian có khả năng xử lý dữ liệu không chắc chắn: Mạng Bayesian có thể xử lý dữ liệu không chắc chắn và không đầy đủ, giúp cho việc dự đoán kết quả trở nên chính xác hơn.
* Mạng Bayesian có khả năng học từ dữ liệu: Mạng Bayesian có khả năng học từ dữ liệu, tức là nó có thể tự động học các phụ thuộc có điều kiện của các biến ngẫu nhiên từ dữ liệu đầu vào.
* Mạng Bayesian có khả năng dự đoán tương lai: Mạng Bayesian có khả năng dự đoán tương lai dựa trên các phụ thuộc có điều kiện của các biến ngẫu nhiên.
* Mạng Bayesian có khả năng giải thích kết quả: Mạng Bayesian có khả năng giải thích kết quả, giúp cho người dùng hiểu được lý do tại sao một kết quả được dự đoán.
* Mạng Bayesian có khả năng tích hợp các kiến thức chuyên môn: Mạng Bayesian có khả năng tích hợp các kiến thức chuyên môn vào quá trình dự đoán, giúp cho kết quả dự đoán trở nên chính xác hơn 1.

## 14. Probabilistic Reasoning - Lý luận xác suất

Chương này đã mô tả mạng Bayesian, một cách biểu diễn được phát triển tốt cho tri thức không chắc chắn. Mạng Bayesian đóng một vai trò gần giống với vai trò của logic mệnh đề đối với tri thức xác định.

### Representing knowledge in an uncertain domain - Biểu diễn tri thức trong miền không xác định

* Mạng Bayesian là một đồ thị tuần hoàn có hướng có các nút tương ứng với các biến ngẫu nhiên; mỗi nút có một phân bố có điều kiện cho nút đó, dựa trên nút cha của nó.
* Mạng Bayesian cung cấp một cách ngắn gọn để biểu diễn các mối quan hệ độc lập có điều kiện trong miền.

### The semantics of bayesian networks - Ngữ nghĩa của mạng bayesian

* Mạng Bayes xác định một phân phối chung đầy đủ trong đó mỗi mục nhập chung là sản phẩm của các mục tương ứng trong phân phối có điều kiện cục bộ. Nó được biết đến là nhỏ hơn theo cấp số nhân so với phân phối chung được liệt kê rõ ràng.
* Nhiều phân phối có điều kiện có thể được biểu diễn một cách cô đọng bằng các họ phân phối chuẩn. Mạng Bayes lai, bao gồm cả biến rời rạc và biến liên tục, sử dụng nhiều phân phối chuẩn khác nhau.

### Exact inference in bayesian networks - Suy luận chính xác trong mạng bayesian

* Suy luận trong mạng Bayes liên quan đến việc tính toán phân bố xác suất của một tập hợp các biến truy vấn dựa trên một tập hợp các biến bằng chứng. Các thuật toán suy luận chính xác, như loại bỏ biến, đánh giá hiệu quả tổng các tích của xác suất có điều kiện để đạt được điều này.
* Trong polytrees (mạng kết nối đơn), suy luận chính xác cần có thời gian tuyến tính theo kích thước của mạng. Trong trường hợp tổng quát, vấn đề là khó giải quyết.

### Approximate inference in bayesian networks - Suy luận gần đúng trong mạng bayesian

* Các kỹ thuật xấp xỉ ngẫu nhiên như trọng số khả năng và chuỗi Markov Monte Carlo có thể đưa ra ước tính hợp lý về xác suất thực sự sau đó trong mạng và có thể xử lý các mạng lớn hơn nhiều so với các thuật toán chính xác.

### Relational and first-order probability models - Mô hình xác suất quan hệ và xác suất bậc nhất

* Lý thuyết xác suất có thể được kết hợp với các ý tưởng biểu diễn từ logic bậc nhất để tạo ra các hệ thống rất mạnh mẽ cho việc suy luận trong điều kiện không chắc chắn. Các mô hình xác suất quan hệ (RPM) bao gồm các hạn chế về mặt biểu diễn đảm bảo phân bố xác suất được xác định rõ ràng có thể được biểu thị dưới dạng mạng Bayesian tương đương. Các mô hình xác suất vũ trụ mở xử lý sự tồn tại và tính không chắc chắn về danh tính, xác định sự phân bố xác suất trên không gian vô hạn của các thế giới khả hữu bậc nhất.
* Nhiều hệ thống thay thế khác nhau cho việc suy luận trong điều kiện không chắc chắn đã được đề xuất. Nói chung, các hệ thống chức năng xác thực không phù hợp lắm cho việc lập luận như vậy.

**Nhóm 5: chương 15 (Probabilistic reasoning over time) và chương 16 (making simple decision)**

**chương 15 (Probabilistic reasoning over time)**

This chapter has addressed the general problem of representing and reasoning about probabilistic temporal processes. The main points are as follows:

• The changing state of the world is handled by using a set of random variables to represent the state at each point in time.

• Representations can be designed to satisfy the **Markov property**, so that the future is independent of the past given the present. Combined with the assumption that the process is **stationary**—that is, the dynamics do not change over time—this greatly simplifies the representation.

* A temporal probability model can be thought of as containing **a transition model** describing the state evolution and a **sensor model** describing the observation process.

• The principal inference tasks in temporal models are **filtering, prediction, smoothing**, and computing the most likely explanation. Each of these can be achieved using simple, recursive algorithms whose run time is linear in the length of the sequence.

• Three families of temporal models were studied in more depth: **hidden Markov models, Kalman filters, and dynamic Bayesian networks** (which include the other two as special cases).

• Unless special assumptions are made, as in Kalman filters, exact inference with many state variables is intractable. In practice, the **particle filtering** algorithm seems to be an effective approximation algorithm.

• When trying to keep track of many objects, uncertainty arises as to which observations belong to which objects—the **data association** problem. The number of association hypotheses is typically intractably large, but MCMC and particle filtering algorithms for data association work well in practice.

## 15. Probabilistic Reasoning over time - Lý luận xác suất theo thời gian

Chương này đã giải quyết vấn đề chung về biểu diễn và suy luận về các quá trình thời gian xác suất. Các điểm chính như sau:

### Time and Uncertainty - Thời gian và sự không chắc chắn

* Trạng thái thay đổi của thế giới được xử lý bằng cách sử dụng một tập hợp các biến ngẫu nhiên để thể hiện trạng thái tại mỗi thời điểm.
* Bằng cách đảm bảo rằng các biểu diễn thỏa mãn tính chất Markov, tương lai trở nên độc lập với quá khứ xét đến hiện tại. Khi kết hợp với giả định về tính dừng, trong đó động lực học không thay đổi theo thời gian, nó làm đơn giản hóa đáng kể việc biểu diễn.
* Một mô hình xác suất tạm thời có thể được coi là chứa một mô hình chuyển tiếp mô tả sự tiến hóa trạng thái và một mô hình cảm biến mô tả quá trình quan sát.

### Inference in temporal models - Suy luận trong các mô hình thời gian

* Nhiệm vụ suy luận chính trong các mô hình thời gian là lọc, dự đoán, làm mịn và tính toán lời giải thích hợp lý nhất. Mỗi điều này có thể đạt được bằng cách sử dụng các thuật toán đệ quy đơn giản có thời gian chạy tuyến tính theo độ dài của chuỗi.

### Hidden Markov models - Mô hình Markov ẩn

* Ba họ mô hình thời gian đã được nghiên cứu sâu hơn: mô hình Markov ẩn, bộ lọc Kalman và mạng Bayesian động (bao gồm hai họ còn lại là trường hợp đặc biệt).

### Keeping track of many objects - Theo dõi nhiều đối tượng

* Trừ khi các giả định đặc biệt được đưa ra, như trong bộ lọc Kalman, việc suy luận chính xác với nhiều biến trạng thái là khó có thể thực hiện được. Trong thực tế, thuật toán lọc hạt dường như là một thuật toán gần đúng hiệu quả.
* Khi cố gắng theo dõi nhiều đối tượng, nảy sinh sự không chắc chắn về việc quan sát nào thuộc về đối tượng nào - vấn đề liên kết dữ liệu. Số lượng các giả thuyết liên kết thường rất lớn, nhưng các thuật toán MCMC và lọc hạt để liên kết dữ liệu hoạt động tốt trong thực tế.

## 16. Making simple decisions - Đưa ra những quyết định đơn giản

Chương này cho thấy cách kết hợp lý thuyết tiện ích với xác suất để cho phép một tác nhân lựa chọn các hành động sẽ tối đa hóa hiệu suất mong đợi của nó.

### The basis of utility theory - Cơ sở của lý thuyết tiện ích

* Lý thuyết xác suất mô tả những gì một tác nhân nên tin trên cơ sở bằng chứng, lý thuyết tiện ích mô tả những gì một tác nhân muốn và lý thuyết quyết định kết hợp cả hai lại với nhau để mô tả những gì một tác nhân nên làm.
* Chúng ta có thể sử dụng lý thuyết quyết định để xây dựng một hệ thống đưa ra quyết định bằng cách xem xét tất cả các hành động có thể xảy ra và chọn ra hành động dẫn đến kết quả mong đợi tốt nhất. Một hệ thống như vậy được gọi là một tác nhân hợp lý.

### Utility functions - Các chức năng tiện ích

* Lý thuyết tiện ích mô tả rằng một tác nhân có sở thích giữa các cuộc xổ số phù hợp với một tập hợp các tiên đề cơ bản, có thể được mô tả là có chức năng tiện ích. Ngoài ra, tác nhân chọn các hành động như thể tối đa hóa tiện ích mong đợi của nó.

### Multiattribute utility functions - Các hàm tiện ích đa thuộc tính

* Lý thuyết tiện ích đa thuộc tính liên quan đến các tiện ích phụ thuộc vào nhiều thuộc tính riêng biệt của các trạng thái. Ưu thế ngẫu nhiên là một phương pháp có giá trị để đưa ra các quyết định rõ ràng, ngay cả khi không có giá trị tiện ích chính xác cho các thuộc tính.

### Decision networks - Mạng quyết định

* Mạng quyết định cung cấp một hình thức đơn giản để thể hiện và giải quyết các vấn đề về quyết định. Chúng là phần mở rộng tự nhiên của mạng Bayesian, chứa các nút quyết định và tiện ích bên cạnh các nút cơ hội.

### The value of information - Giá trị của thông tin

* Đôi khi, việc giải quyết một vấn đề liên quan đến việc tìm kiếm thêm thông tin trước khi đưa ra quyết định. Giá trị của thông tin được định nghĩa là sự cải thiện mong đợi về tính hữu dụng so với việc đưa ra quyết định mà không có thông tin.

### Decision-theoretic expert systems - Hệ thống chuyên gia lý thuyết quyết định

* Các hệ thống chuyên gia với khả năng thông tin tiện ích cung cấp các tính năng nâng cao ngoài các hệ thống suy luận cơ bản. Ngoài việc ra quyết định, họ có thể đánh giá giá trị của thông tin để xác định nên đặt câu hỏi nào, đề xuất kế hoạch dự phòng và đánh giá tác động của những thay đổi nhỏ trong đánh giá xác suất và tiện ích đối với quyết định của họ.

**Nhóm 6: Chương 17 (making complex decision) và chương 20(learning probabilistic model)**

**Nhóm 7: chương 21(Reinforce learning)**

**Nhóm 8: chương 18 (learning from example )**

**Nhóm 9: chương 19 (knowledge in learning) và chương 6 (constraint satisfication problem)**

**Nhóm 10: chương 22 (natural language processing) và chương 23(natural language for communication)**

**Nhóm 11: chương 24(perception) và chương 25 (robotics)**