Logic Programming, Knowledge Representation, and Non Monotonic Reasoning

Agostino Dovier

Department of Mathematics and Computer Science,
University of Udine, Italy

Udine, Fall 2011

Summary

Introduction

Knowledge representation is one of the most important subareas of artificial intelligence. If we want to design an entity (a machine or a program) capable of behaving intelligently in some environment, then we need to supply this entity with sufficient knowledge about this environment. To do that, we need an unambiguous language capable of expressing this knowledge, together with some precise and well understood way of manipulating sets of sentences of the language which will allow us to draw inferences, answer queries, and to update both the knowledge base and the desired program behavior.

Chitta Baral and Michael Gelfond. Logic Programming and Knowledge Representation Journal of Logic Programming 19/20, 73–148, 1994. (disponibile on-line il RR di UTEP dalle pagine degli autori)

Introduction

> Expressing information in declarative sentences is far more modular than expressing it in segments of computer programs or in tables. Sentences can be true in a much wider context than specific programs can be used. The supplier of a fact does not have to understand much about how the receiver functions or how or whether the receiver will use it. The same fact can be used for many purposes, because the logical consequences of collections of facts can be available. [McC59]

Chitta Baral and Michael Gelfond. Logic Programming and Knowledge Representation Journal of Logic Programming 19/20, 73–148, 1994. (disponibile on-line il RR di UTEP dalle pagine degli autori)

Introduction

This idea has been further developed by many researchers with various backgrounds and interests. First, the classical logic of predicate calculus served as the main technical tool for the representation of knowledge. It has a well-defined semantics and a well-understood and powerful inference mechanism, and it proved to be sufficiently expressive for the representation of mathematical knowledge.

Chitta Baral and Michael Gelfond. Logic Programming and Knowledge Representation Journal of Logic Programming 19/20, 73–148, 1994. (disponibile on-line il RR di UTEP dalle pagine degli autori)

Introduction

It was soon realized, however, that for the representation of *commonsense knowledge*, this tool is inadequate. The difficulty is rather deep and related to the so-called "monotonicity" of theories based on predicate calculus. A logic is called *monotonic* if the addition of new axioms to a theory based on it never leads to the loss of any theorems proved in this theory. Commonsense reasoning is nonmonotonic: new information constantly forces us to withdraw previous conclusions. This observation has led to the development and investigation of new logical formalisms, *nonmonotonic logics*.

Chitta Baral and Michael Gelfond. Logic Programming and Knowledge Representation Journal of Logic Programming 19/20, 73–148, 1994. (disponibile on-line il RR di UTEP dalle pagine degli autori)

Introduzione

Il sillogismo

Socrate è un uomo

Introduzione

Il sillogismo

Socrate è un uomo
Tutti gli uomini sono mortali

Introduzione

Il sillogismo

Socrate è un uomo
Tutti gli uomini sono mortali
Pertanto Socrate è mortale

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim’ordine.

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\texttt{uomo}(\texttt{socrate}) \tag{1}$$

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\mathtt{uomo}(\mathtt{socrate}) \tag{1}$$

Tutti gli uomini sono mortali:

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\texttt{uomo}(\texttt{socrate}) \tag{1}$$

Tutti gli uomini sono mortali:

$$\forall X \, (\texttt{uomo}(X) \rightarrow \texttt{mortale}(X)) \tag{2}$$

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\texttt{uomo}(\texttt{socrate}) \tag{1}$$

Tutti gli uomini sono mortali:

$$\forall X\,(\texttt{uomo}(X) \to \texttt{mortale}(X)) \tag{2}$$

Riflettiamo sul verso dell'implicazione.

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\texttt{uomo}(\texttt{socrate}) \tag{1}$$

Tutti gli uomini sono mortali:

$$\forall X\,(\texttt{uomo}(X) \to \texttt{mortale}(X)) \tag{2}$$

Riflettiamo sul verso dell'implicazione. Ad esempio, se `rex` è un cane, avremo che `mortale(rex)` ma non `uomo(rex)`

Introduzione

Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\texttt{uomo}(\texttt{socrate}) \tag{1}$$

Tutti gli uomini sono mortali:

$$\forall X\,(\texttt{uomo}(X) \to \texttt{mortale}(X)) \tag{2}$$

Riflettiamo sul verso dell'implicazione. Ad esempio, se `rex` è un cane, avremo che `mortale(rex)` ma non `uomo(rex)` Si dovrebbe sapere che chi è cane non è uomo, a dire il vero, ne parleremo (CWA).

Introduzione
Il sillogismo

Modelliamo la cosa con la logica del prim'ordine.
Socrate è un uomo:

$$\texttt{uomo}(\texttt{socrate}) \tag{1}$$

Tutti gli uomini sono mortali:

$$\forall X\,(\texttt{uomo}(X) \rightarrow \texttt{mortale}(X)) \tag{2}$$

Riflettiamo sul verso dell'implicazione. Ad esempio, se `rex` è un cane, avremo che `mortale(rex)` ma non `uomo(rex)` Si dovrebbe sapere che chi è cane non è uomo, a dire il vero, ne parleremo (CWA).
Inoltre, vale che

$$(1),(2) \models \texttt{mortale}(\texttt{socrate})$$

Introduzione

Il sillogismo

Inoltre, se ora scopriamo che

$$\texttt{uomo}(\texttt{agostino}) \tag{3}$$

allora vale che

$$(1), (2), (3) \models \texttt{mortale}(\texttt{socrate})$$

e

$$(1), (2), (3) \models \texttt{mortale}(\texttt{agostino})$$

Introduzione
Il sillogismo

Inoltre, se ora scopriamo che

$$\texttt{uomo}(\texttt{agostino}) \tag{3}$$

allora vale che

$$(1),(2),(3) \models \texttt{mortale}(\texttt{socrate})$$

e

$$(1),(2),(3) \models \texttt{mortale}(\texttt{agostino})$$

All'aumentare delle premesse, aumentano i teoremi (e quelli vecchi rimangono validi). Siamo in presenza di monotonia.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.
Vogliamo andare dall'altra parte.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.
Vogliamo andare dall'altra parte.
Le sbarre non sono abbassate.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.
Vogliamo andare dall'altra parte.
Le sbarre non sono abbassate.
Alla luce di questa conoscenza sembrerebbe plausibile attraversare

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.
Vogliamo andare dall'altra parte.
Le sbarre non sono abbassate.
Alla luce di questa conoscenza sembrerebbe plausibile attraversare

Se c'è un guasto elettrico, allora le sbarre sono sempre alzate o sempre abbassate.

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.
Vogliamo andare dall'altra parte.
Le sbarre non sono abbassate.
Alla luce di questa conoscenza sembrerebbe plausibile attraversare

Se c'è un guasto elettrico, allora le sbarre sono sempre alzate o sempre abbassate.

E con questa nuova informazione, attraverseste?

Introduzione

La perdita della monotonia

Se si attraversano i binari quando passa il treno non si raggiunge l'altra parte.
Se si attraversano i binari quando non c'è il treno si raggiunge l'altra parte.
Se le sbarre sono abbassate c'è il treno.
Vogliamo andare dall'altra parte.
Le sbarre non sono abbassate.
Alla luce di questa conoscenza sembrerebbe plausibile attraversare

Se c'è un guasto elettrico, allora le sbarre sono sempre alzate o sempre abbassate.

E con questa nuova informazione, attraverseste?

E se sapessimo anche che c'è un guasto elettrico?

Introduzione

La perdita della monotonia

Nel commonsense reasoning si fa uso continuo di asserzioni note come normative statements, ovvero del tipo:

A's are normally B's

che ammettono però delle eccezioni.

Introduzione

La perdita della monotonia: i pinguini

Suppose that a reasoning agent has the following knowledge about birds: birds typically fly and penguins are non flying birds. He also knows that Tweety is a bird. Suppose now that the agent is hired to build a cage for Tweety, and he leaves off the roof on the grounds that he does not whether or nor Tweety can fly. It would be reasonable for us to view this argument as invalid and to refuse the agent's product. This would be not the case if Tweety could not fly for some reason (unknown to the agent), and we refused to pay for the bird cage because had unnecessarily put a roof on it. [McCarthy, 1959]

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.
Ogni pinguino è anormale.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.
Ogni pinguino è anormale.
Metti il tetto alla gabbietta per un uccello che vola.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.
Ogni pinguino è anormale.
Metti il tetto alla gabbietta per un uccello che vola.
tweety è un uccello.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.
Ogni pinguino è anormale.
Metti il tetto alla gabbietta per un uccello che vola.
tweety è un uccello.
Costruisci la gabbietta per tweety

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.
Ogni pinguino è anormale.
Metti il tetto alla gabbietta per un uccello che vola.
tweety è un uccello.
Costruisci la gabbietta per tweety
tweety è un canarino

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola.
Ogni pinguino è un uccello.
Ogni canarino è un uccello.
Ogni pinguino è anormale.
Metti il tetto alla gabbietta per un uccello che vola.
tweety è un uccello.
Costruisci la gabbietta per tweety
tweety è un canarino
Pensate invece a pingu, che è un pinguino.

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola (per $\leftarrow$, pensate alle api)

$$\forall X\,(\mathtt{uccello}(X) \wedge \neg\mathtt{ab}(X) \rightarrow \mathtt{vola}(X)) \tag{4}$$

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola (per $\leftarrow$, pensate alle api)

$$\forall X\,(\mathtt{uccello}(X) \wedge \neg\mathtt{ab}(X) \rightarrow \mathtt{vola}(X)) \tag{4}$$

Ogni pinguino è un uccello. Ogni canarino è un uccello.

$$\forall X\,(\mathtt{pinguino}(X) \rightarrow \mathtt{uccello}(X)) \tag{5}$$

$$\forall X\,(\mathtt{canarino}(X) \rightarrow \mathtt{uccello}(X)) \tag{6}$$

Introduzione

La perdita della monotonia: i pinguini

Ogni uccello, che non sia anormale, vola (per $\leftarrow$, pensate alle api)

$$\forall X\,(\texttt{uccello}(X) \wedge \neg\texttt{ab}(X) \rightarrow \texttt{vola}(X)) \tag{4}$$

Ogni pinguino è un uccello. Ogni canarino è un uccello.

$$\forall X\,(\texttt{pinguino}(X) \rightarrow \texttt{uccello}(X)) \tag{5}$$

$$\forall X\,(\texttt{canarino}(X) \rightarrow \texttt{uccello}(X)) \tag{6}$$

Ogni pinguino è anormale, metti il tetto alla gabbietta per un uccello che vola, tweety è un uccello.

$$\forall X\,(\texttt{pinguino}(X) \rightarrow \texttt{ab}(X)) \tag{7}$$

$$\forall X\,(\texttt{uccello}(X) \wedge \texttt{vola}(X) \rightarrow \texttt{metti\_tetto\_gabbia}(X)) \tag{8}$$

$$\texttt{uccello}(\texttt{tweety}) \tag{9}$$

$$(4), (5), (6), (7), (8), \texttt{CWA} \models \texttt{metti\_tetto\_gabbia}(\texttt{tweety})$$

Introduzione

La perdita della monotonia: i pinguini

Se ora so che tweety è un canarino.

$$\mathtt{canarino}(\mathtt{tweety}) \tag{10}$$

vale ancora che

$$(4),(5),(6),(7),(8),(9),(10)\mathtt{CWA} \models \mathtt{metti\_tetto\_gabbia}(\mathtt{tweety})$$

Introduzione

La perdita della monotonia: i pinguini

Se ora so che tweety è un canarino.

$$\texttt{canarino}(\texttt{tweety}) \tag{10}$$

vale ancora che

$$(4),(5),(6),(7),(8),(9),(10)\texttt{CWA} \models \texttt{metti\_tetto\_gabbia}(\texttt{tweety})$$

Ragioniamo con pingu. Se so che pingu è un uccello:

$$\texttt{uccello}(\texttt{pingu}) \tag{11}$$

vale che

$$(4)\text{–}(11), \texttt{CWA} \models \texttt{metti\_tetto\_gabbia}(\texttt{pingu})$$

Ma se scoprissi ora che

$$\texttt{pinguino}(\texttt{pingu}) \tag{12}$$

$$(4)\text{–}(11),(12)\texttt{CWA} \not\models \texttt{metti\_tetto\_gabbia}(\texttt{pingu})$$

Introduzione

La ricerca della perdita della monotonia

Abbiamo perso la monotonia.

Introduzione

La ricerca della perdita della monotonia

Abbiamo perso la monotonia.
Dove?

Introduzione

La ricerca della perdita della monotonia

Abbiamo perso la monotonia.
Dove?
Lo investigheremo nelle prossime lezioni. Per ora vi accenno solo che il punto critico è questo:

$$\forall X\,(\mathtt{uccello}(X) \wedge \neg\mathtt{ab}(X) \rightarrow \mathtt{vola}(X)) \qquad (13)$$

Introduzione

La ricerca della perdita della monotonia

Abbiamo perso la monotonia.
Dove?
Lo investigheremo nelle prossime lezioni. Per ora vi accenno solo che il punto critico è questo:

$$\forall X\,(\texttt{uccello}(X) \wedge \neg\texttt{ab}(X) \rightarrow \texttt{vola}(X)) \tag{13}$$

Che non è una clausola di Horn

Introduzione

La ricerca della perdita della monotonia

Abbiamo perso la monotonia.
Dove?
Lo investigheremo nelle prossime lezioni. Per ora vi accenno solo che il punto critico è questo:

$$\forall X\,(\texttt{uccello}(X) \wedge \neg\texttt{ab}(X) \rightarrow \texttt{vola}(X)) \tag{13}$$

Che non è una clausola di Horn
Vedremo anche come lo studio dei frammenti di logica predicativa adatti alla KR e NMR ha portato dei risultati molto eleganti della teoria della complessità computazionale.

Introduzione

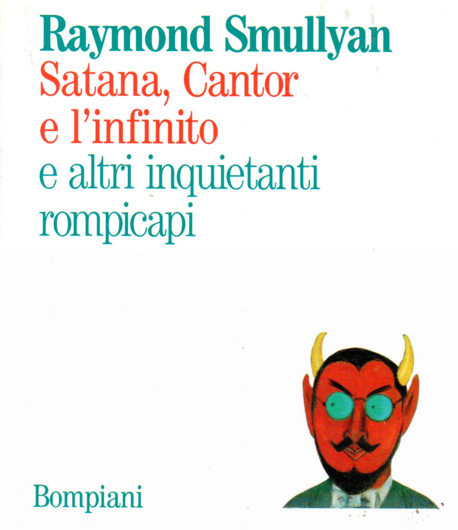
La ricerca della perdita della monotonia

Abbiamo perso la monotonia.
Dove?
Lo investigheremo nelle prossime lezioni. Per ora vi accenno solo che il punto critico è questo:

$$\forall X\, (\mathtt{uccello}(X) \wedge \neg \mathtt{ab}(X) \rightarrow \mathtt{vola}(X)) \tag{13}$$

Che non è una clausola di Horn
Vedremo anche come lo studio dei frammenti di logica predicativa adatti alla KR e NMR ha portato dei risultati molto eleganti della teoria della complessità computazionale.
Tali risultati sono anche UTILI, nel senso che avremo dei linguaggi per diverse classi di complessità (tra cui NP) corredati di risolutori con efficienza al massimo delle conoscenze attuali.

Smullyan's puzzles



Smullyan's puzzles

Con una fitta di apprensione come non aveva mai provato prima, un antropologo di nome Abercrombie mise piede sull'isola dei Cavalieri e dei Fanti. Sapeva che quest'isola era popolata da gente quanto mai bizzarra: i cavalieri, che facevano solo affermazioni vere, e i fanti, che facevano solo affermazioni false. "Come farò a sapere qualcosa di quest'isola se non riesco a distinguere chi mente e chi dice la verità?" si chiedeva Abercrombie.

Abercrombie si rendeva conto che, prima di poter scoprire alcunché, avrebbe dovuto farsi un amico, qualcuno di cui potesse essere certo che gli avrebbe sempre detto la verità. Così, quando si imbatté nel primo gruppo di indigeni — tre persone, i cui nomi erano presumibilmente Arturo, Bernardo e Carlo —, Abercrombie pensò tra sé: "Ecco una buona occasione per trovarmi un cavaliere." Anzitutto Abercrombie chiese ad Arturo: "Bernardo e Carlo sono entrambi cavalieri?" Arturo rispose: "Sì." Allora Abercrombie chiese: "Bernardo è un cavaliere?" Con sua grande sorpresa, Arturo rispose: "No."

Carlo è un cavaliere o un fante?

Smullyan's puzzles

Abercrombie fu informato da quello dei tre che sapeva ormai essere un cavaliere che l'isola aveva uno stregone.

"Oh, ottimo!" esclamò Abercrombie. "Noi antropologi abbiamo un interesse particolare per gli stregoni, i maghi, i guaritori, gli sciamani e altra gente del genere. Dove posso trovarlo?"

"Deve chiedere al re" fu la risposta.

Bene, l'antropologo riuscì a ottenere un'udienza dal re, e gli disse che desiderava incontrare lo stregone.

"Oh, non può incontrarlo," disse il re, "se non incontra prima il suo apprendista. Se l'apprendista stregone darà di lei un giudizio favorevole, allora le consentirà di incontrare il suo maestro; altrimenti non glielo permetterà."

"Ha un apprendista, lo stregone?" chiese l'antropologo.

"Certamente!" rispose il re. "C'è una famosa composizione musicale su di lui: credo che il compositore sia Dukas. Comunque, se lei desidera incontrare l'apprendista stregone, adesso è a casa sua, che è la terza casa di via delle Palme. Al momento, ha due ospiti. Se, arrivando, lei riesce a dedurre quale delle tre persone presenti è l'apprendista stregone, credo che farà abbastanza colpo su di lui perché acconsenta a farle incontrare lo stregone. Buona fortuna!"

Smullyan's puzzles

Una breve camminata condusse l'antropologo alla casa. Quando entrò, erano presenti tre persone.

"Chi di voi è l'apprendista stregone?" chiese Abercrombie.

"Io," rispose uno.

"Sono io l'apprendista stregone!" gridò un secondo.

Il terzo rimase in silenzio.

"Può dirmi qualcosa anche lei?" chiese Abercrombie.

"È buffo," rispose la terza persona, con un sorriso scaltro. "Al massimo, uno soltanto di noi tre dice la verità!"

È possibile dedurre quale dei tre è l'apprendista stregone?

The zebra puzzle

Vi sono cinque case di cinque colori diversi: rosso, verde, avorio, blu, giallo.

The zebra puzzle

Vi sono cinque case di cinque colori diversi: rosso, verde, avorio, blu, giallo.
Gli inquilini delle case hanno nazionalità diversa. Essi provengono da Giappone, Inghilterra, Norvegia, Ucraina, e Spagna.

The zebra puzzle

Vi sono cinque case di cinque colori diversi: rosso, verde, avorio, blu, giallo.
Gli inquilini delle case hanno nazionalità diversa. Essi provengono da Giappone, Inghilterra, Norvegia, Ucraina, e Spagna.
Ogni inquilino possiede un animale. Gli animali sono: cavallo, chiocciola, zebra, volpe, e cane.

The zebra puzzle

Vi sono cinque case di cinque colori diversi: rosso, verde, avorio, blu, giallo.
Gli inquilini delle case hanno nazionalità diversa. Essi provengono da Giappone, Inghilterra, Norvegia, Ucraina, e Spagna.
Ogni inquilino possiede un animale. Gli animali sono: cavallo, chiocciola, zebra, volpe, e cane.
Inoltre sappiamo che ogni inquilino beve usualmente una sola delle seguenti bevande: acqua, caffè, tea, latte, aranciata

The zebra puzzle

Vi sono cinque case di cinque colori diversi: rosso, verde, avorio, blu, giallo.
Gli inquilini delle case hanno nazionalità diversa. Essi provengono da Giappone, Inghilterra, Norvegia, Ucraina, e Spagna.
Ogni inquilino possiede un animale. Gli animali sono: cavallo, chiocciola, zebra, volpe, e cane.
Inoltre sappiamo che ogni inquilino beve usualmente una sola delle seguenti bevande: acqua, caffè, tea, latte, aranciata
e guida una (sola) auto di una delle seguenti marche: Fiat, Lancia, BMW, Skoda, Audi.

The zebra puzzle

Sono note inoltre le seguenti informazioni:

- l'inglese vive nella casa rossa;
- lo spagnolo ha il cane;
- il norvegese vive nella prima casa a sinistra;
- nel garage della casa gialla c'è una Skoda;
- chi guida la BMW vive nella casa vicina a chi possiede la volpe;
- il norvegese vive in una casa vicino alla casa blu;
- chi possiede la Lancia possiede anche la chiocciola;
- chi guida la Fiat beve aranciata;
- l'ucraino beve tea;
- il giapponese guida l'Audi;
- la Skoda è parcheggiata nel garage di una casa vicina alla casa dove c'è il cavallo;
- nella casa verde si beve caffè;
- la casa verde è immediatamente a destra della casa avorio;
- il latte si beve nella casa di mezzo (la terza).

The zebra puzzle

La domanda e': E' stata ritrovata una zebra in mezzo alla strada. Chi la ha lasciata scappare?

La capra e il cavolo




La capra e il cavolo

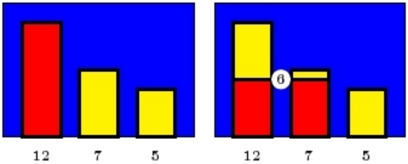
Un pastore deve attraversare un fiume portando sull'altra riva un lupo e una capra affamati e un cavolo gigante. Ha a disposizione una barca a remi con la quale può traghettare un solo oggetto o animale alla volta. Ma, attenzione, non può lasciare da soli:

- il lupo e la capra perché il lupo si mangia la capra;
- la capra ed il cavolo perché la capra si mangia il cavolo.

Quanti viaggi deve fare per portare sull'altra riva il lupo, la capra ed il cavolo?

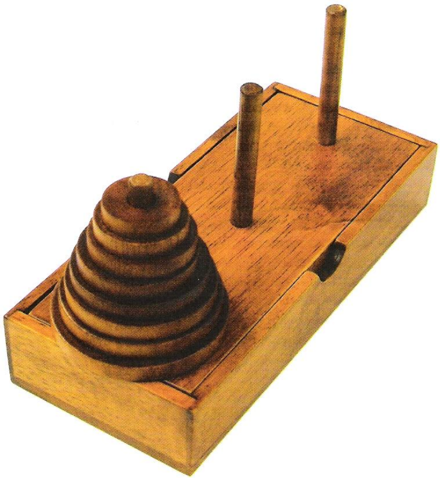
The three barrels

"There are three barrels of capacity N (even number), $N/2+1$, and $N/2-1$, resp. At the beginning the largest barrel is full of Taylor's Porto, the other two are empty. We wish to reach a state in which the two larger barrels contain the same amount of porto. The only permissible action is to pour porto from one barrel to another, until the latter is full or the former is empty."



The Hanoi Tower

The Hanoi Tower

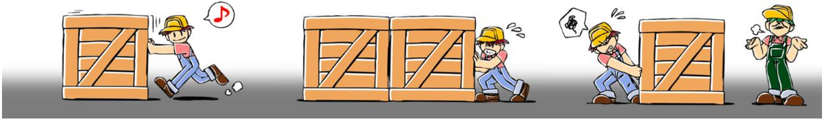


Sokoban History

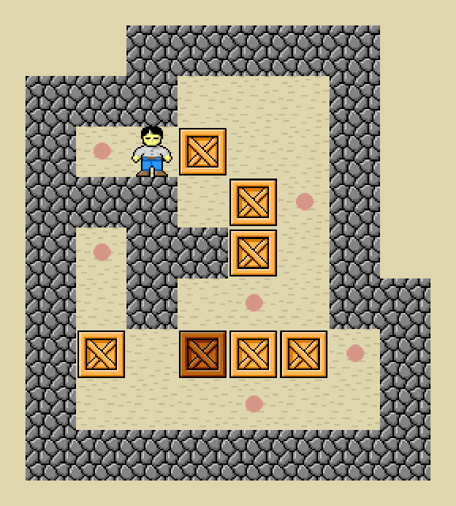
- Sokoban is a type of transport puzzle invented by *Hiroyuki Imabayashi* in 1980
- Published by the Japanese company Thinking Rabbit, Inc. in 1982.
- Sokoban means “warehouse-keeper” (magazziniere) in Japanese.
- Thinking Rabbit joined Square Co., Ltd.

Sokoban Rules

(from `http://www.sokoban.jp/`)

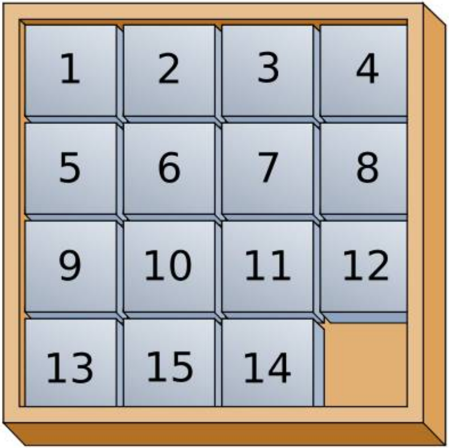


Sokoban @ work



Peg solitaire

Sam Lloyd's Puzzle



Sudoku

		1						
		2		3				4
			5			6		7
5			1	4				
	7						2	
				7	8			9
8		7			9			
4				6		3		
						5		