Solving Mathematical Puzzles: A Challenging Competition for AI

Federico Chesani, Paola Mello, Michela Milano

■ *Recently, a number of noteworthy results have been achieved in various fields of artificial intelligence, and many aspects of the problem-solving process have received significant attention by the scientific community. In this context, the extraction of comprehensive knowledge suitable for problem solving and reasoning, from textual and pictorial problem descriptions, has been less investigated, but recognized as essential for autonomous thinking in artificial intelligence. In this work we present a challenge where methods and tools for deep understanding are strongly needed for enabling problem solving: we propose to solve mathematical puzzles by means of computers, starting from text and diagrams describing them, without any human intervention. We are aware that the proposed challenge is hard and difficult to solve nowadays (and in the foreseeable future), but even studying and solving only single parts of the proposed challenge would represent an important step forward for artificial intelligence.*

Since its origin, the holy grail of artificial intelligence has been to understand the nature of intelligence and to engineer systems that exhibit such intelligence through vision, language, emotion, motion, and reasoning. In such context, AI researchers have always looked for challenges to push forward the limit of what computers can do autonomously and to measure the level of "intelligence" achieved. Competitions have been and are currently run on conversational behavior (for example, the Loebner prize[1]), automatic control (for example, the International Aerial Robotics Competition[2] or the DARPA Grand Challenge on driverless cars[3]), cooperation and coordination in robotics (for example, the RoboCup[4]), logic reasoning and knowledge (for example, the CADE ATP System competition for theorem provers[5]), and natural language (for example, the EVALITA competition[6] for the Italian language). Historically, also games have raised the interest of the AI community: a number of competitions are still being held nowadays (for example, the World Computer Chess Championship,[7] Mario Championship[8] [Togelius et al. 2013] and its successor Platformer Competition,[9] the General Game Playing competition[10]). For a high-level overview of the field of AI in games see the paper by Yannakakis and Togelius (2014).

Some of these challenges have indeed brought many insights and advancements on various artificial intelligence fields. We mention in the following three of them, notable for the achieved results, as well as for their impact on the media and the consequent dissemination of the AI discipline to the general audience. In the last century, a famous challenge of AI concerned the game of chess, considered a symbol of complexity, problem solving, and strategic ability. As widely known, the competition between humans and the Deep Blue computer was definitely won by computers in 1997 when the world chess champion Garry Kasparov was defeated. A key lesson learned from Deep Blue's strength is that efficient heuristic search can be much more effective than sophisticated reasoning guided search. In fact, heuristic search was so successful that it led Kasparov to exclaim, "I could feel — I could smell — a new kind of intelligence across the table" (*Time Magazine*).[11] Despite a very good result in terms of problem solving, the search-intensive nature of the solution approach (in terms of number of configurations explored, memory, and processing power) was in fact the main criticism to the experiment. Noam Chomsky commented that Deep Blue defeating Kasparov in chess was "as interesting as a bulldozer that can win the Olympic weight-lifting competition."[12] Very recently, computers won against humans at the game Go, famous for being one of the last games where humans were still better players. In March 2016, the AlphaGO (Silver et al. 2016) prototype by Google won[13] against Lee Sedol, one of the world's best players. The distinct feature of AlphaGO is the adoption of deep learning techniques.

A second challenge is the Robot Soccer World Cup (RoboCup),[14] the international robotics competition and challenge launched in 1997. The official goal of the RoboCup challenge is the following:

> "By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup."[15]

Even if we are still far from reaching such an ambitious goal, the RoboCup challenge has promoted AI techniques in robotics, planning, self-adaptive systems, and machine vision, showing that even extremely ambitious and visionary challenges can bring huge benefits to a research field by approaching the goal step by step.

The third challenge is about Watson,[16] an AI system developed by the IBM research team. Watson was originally developed to answer questions on a quiz show called *Jeopardy*. In 2011, Watson competed against former human winners and won. Watson can be considered as a very advanced question-answering system that applies and integrates AI techniques such as natural language processing (NLP), knowledge extraction and representation, automated reasoning, and machine learning. The system is able to access "millions of pages" of structured and unstructured content in a very efficient way, thanks to the massive use of parallel processing. A huge amount of knowledge is learned, processed, and used together with a number of inference tasks, to perform question answering in an effective and efficient way. However, how to integrate more complex problem-solving capabilities and reasoning techniques is still a matter of research.

The challenge we propose here goes in the direction of extracting specific knowledge from general text and diagrams that is useful for reasoning and problem solving and can be stated as follows:

> By the middle of the 21st century, (a team of) fully autonomous agent(s) shall win a mathematical puzzle competition against primary school students, winners of the most recent competitions.

Mathematical puzzles are recreational games where a single human player is challenged with a problem, described by text and diagrams. To solve them, a human player uses understanding and intuition, as well as common sense, simple logical and mathematical or geometry knowledge, causal relations, and many other reasoning-related capabilities. The main research question then can be formulated as "What are the requirements, functionalities, and main software components needed for autonomously solving a mathematical puzzle, with no human intervention?" The focus is on deep reasoning featuring (1) the extraction of comprehensive knowledge from multimodal descriptions (texts, diagrams, sound and speech, gesture, and others); (2) the ability of determining both the proper model and the corresponding reasoning capability; and (3) the capability of effectively solving the problem (possibly with a feedback to the two previous steps). All these steps require considerable integration of many artificial intelligence techniques such as natural language and diagram understanding, problem solving and search, modeling, and machine learning. In particular, recent advances in machine learning open new research avenues for deep reasoning, as they enable generalization over manually collected knowledge.

Recently, a number of similar challenges have been proposed. We mention here the Aristo Project (Clark 2015), a challenge with the goal of having the computer pass elementary school math and science exams. Natural language comprehension and diagram understanding are required as fundamental steps, together with some form of inference and algebraic or mathematical solving techniques: for this reason, the Aristo challenge is quite related to our proposal. We will discuss Aristo and other challenges extensively in the related works and challenges section. While the large majority of existing systems that provide end-to-end problem solvers stick to one specific solution method (for example, mathematical equations, logic), our challenge is broader, as the choice of the appropriate model and solution method is considered a fundamental capability.

We are aware that the proposed challenge is hard and of difficult solution nowadays, but we strongly believe that even studying and solving only single parts of the problem would bring important steps forward in artificial intelligence. In addition, on the road to agent autonomy, it would be interesting to study which level of human intervention and interaction with the machine is needed to effectively collaborate to solve the problem.

Mathematical Puzzles

Recreational mathematics is a powerful source of inspiration. Michel Criton said:

> It is a tradition that comes to us with a history of almost four thousand years. We are talking about mathematical and logical entertainment puzzles (that is, recreational mathematics). This tradition has been transmitted from generation to generation and civilization in civilization, mainly thanks to great scientific minds that, to relax, spent a bit of their time on these activities, even if someone considered them as mere "trifles." For example, in Albert Einstein's library, there was a whole section devoted to recreational mathematics. (...) Lewis Carroll, Hamilton, Lagrange, Euler, Descartes, Pascal, Fermat, Cardano, Fibonacci, Alcuin, Diophantus, Archimedes: for these great minds the "recreational mathematics" was not only for fun, but also a powerful source of inspiration.[17]

Mathematical puzzles are an integral part of recreational mathematics and in general are worked on by humans, who must find a solution that satisfies the given problem description. Mathematical puzzles are very different in nature; they require mathematics to solve them, but also logic, intuition, and imagination are essential ingredients. For example, taking into consideration the puzzles available in the Bocconi website,[18] roughly two-thirds of the proposed games can be mapped into logic and constraint-satisfaction problems, or simple algebraic equations. Other puzzles instead comprise geometrical or recognition problems, classification, planning and scheduling tasks, and others.

There are many competitions in this regard for the students of primary and secondary schools organized by associations, research organizations, and universities. For example, every year the Fédération Française des Jeux Mathématiques organizes the international championship of mathematical puzzles,[19] where the competitors should solve a number of puzzles (of various difficulty levels) in the least possible time. For Italian students the organization of the games is led by the center PRISTEM-ELEUSI, part of the Bocconi University. The competition on mathematical puzzles consists of a series of games that students must solve individually in a time of 90 minutes. The degree of difficulty (level) of the "games" is determined according to the various categories of students who participate. For our purposes a first, yet extremely ambitious, achievement could be for an autonomous agent to win an official competition with students of the primary school (first level of difficulty). A large amount of material (in the Italian language) is available at the Bocconi University website. Further material can be found on the web, where a large instance set of mathematical and logic puzzles is available.

Challenge Description

In a computer-aided problem-solving process, there is always a substantial human intervention that enables the encoding of a problem described by text and diagrams in a model and a solution algorithm. Human intervention is essential for identifying problem components (decision variables, constraints, logical relations, objective functions), and the hidden knowledge in the description of the problem: the human player enables the transition from the description of the problem to a model and a solution approach.

In figure 1, human intervention is depicted as a number of (not necessarily) sequential steps: a domain expert reads the text and diagrams of the problem, he/she decides the modeling and solving approach based on his/her experience, and frames the model. At this point, an automatic problem-solving procedure completes the job, producing one or many solutions if they exist.[20] The challenge we propose is gradually to remove the human intervention and let the computer perform the whole task autonomously.

Possible steps for automatic problem solving are those shown in figure 1: (1) Read and understand text and diagrams (when available). (2) Identify a suitable modeling and solving technique. (3) Identify problem components and hidden knowledge. (4) Frame the problem model — represent the original problem and its components by means of an equivalent, machine-understandable model, suitable for reasoning on top of it. (5) Solve the problem by running an automated problem-solving procedure.

For the sake of understanding, these steps are introduced here as sequential and consecutive. However, it is a matter of discussion whether they are in the correct order, if there should be one or more iterations, and how much each step influences the other ones. Indeed, many steps and many interactions between these steps can be conceived for the problem-solving process. For example, iterations might be required if the result of one step is not satisfactory for the next steps, or if the resulting model is not accurate or is even incorrect. In turn, this opens up a number of research directions in solution understanding and model evaluation.

Deep Reasoning in Practice: Examples on Mathematical Puzzles

To make our challenge more practical, we propose in the following a few examples. In particular, two problems (Three Friends and the 10-Digit Puzzle) illustrate the need for a deep understanding of a problem described through natural language text: for the sake of clarity, we discuss them by showing a possible

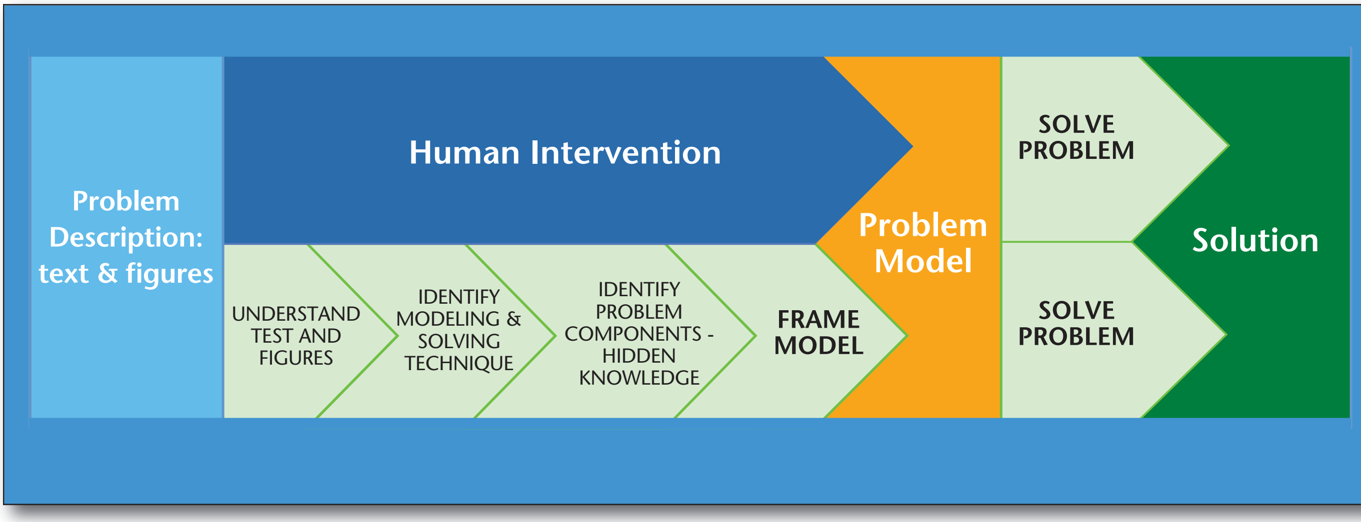

Figure 1. Human Intervention in a Problem-Solving Process.

mapping of the solving process with regard to the steps introduced in figure 1. The third problem is presented to show the need for symbolic reasoning capabilities, such as for example logical inference.

Finally, the fourth and fifth problems are just sketched (with regard to the solution process), but we introduce them to stress the needed interplay of multimodal comprehension in deep reasoning: both problems indeed come with text and diagrams.

Example 1: Three Friends

This first problem belongs to the easiest category, aimed toward primary school students.

> Jacob, Lucy, and Frank are three friends. All together they are 28 years old. In how many years they will be together 37 years old?

Step 1: Understanding Text (and Pictures)

Natural language processing techniques allow the extraction of lexical, syntactical, and semantics information contained in the text. Notice, however, that, in the context of this challenge, understanding a text means also that a computer has to identify the problem components like assertions, goals, constraints, and others. In this specific puzzle, this would mean discovering at least the explicit knowledge contained in the text, namely that:

There are three friends.

All together here means the sum operator.

All together refers to the age of each friend.

The sum of their *ages* now is 28.

Ages and *years* are natural numbers.

How many refers to a quantity X of years.

In X *years* the *sum* of their *ages* will be 37.

We have to find X.

We highlighted in italic a few words that should have a semantic link with specific concepts. For example, the expression "All together" should be related to the concept of mathematical sum, while "age" and "years" should be linked with the natural numbers concept. All the items in the list except the last one refer to the problem's assertions. The last item instead refers to the goal.

Step 2: Identify Modeling and Solving Techniques

The specific problem of the three friends could be modeled as a system of linear equations. Notice that once a model has been identified, there is still an open choice about the best solving technique to be adopted: one solution might be to adopt some algebraic method for linear equations. Another method would be to exploit constraint-satisfaction problem (CSP) techniques for problem solving. In both cases some sort of metaknowledge for reasoning would be required, in order to select the specific solving technique.

Step 3: Identify Problem Components and Hidden Knowledge

The problem components are the following. We have three variables A_J, A_L, and A_F that represent the ages of the three friends. X is the number of years that we have to find. The three variables as well as X, are integers. Moreover the domain of X could be defined as $X \in [0..28]$.

There is (at least) one other piece of hidden information that requires a commonsense knowledge base about the time and the flowing of time. In particular, the time will flow with the same speed for all three friends. Summing up, the information that the three friends will together be 37 years old can be modeled as the fact that after X years their ages will be respectively $A_J + X$, $A_L + X$, and $A_F + X$.

Step 4: Frame the Model

Given the problem components, we can now state the following model based on linear equations:

$A_J + A_L + A_F = 28$

$(A_J + X) + (A_L + X) + (A_F + X) = 37$

Step 5: Solve the Problem

Once the problem has been properly framed into a (formal) model and a solving technique has been chosen for solving it, it is possible to run the algorithms and compute the solution.

> The solution to this problem is $X = 3$.

Example 2: 10-Digit Puzzle

The following example, taken from Martin Gardner's *Mathematical Puzzle Tales* (Gardner 1981), is more difficult for a human player, at least with regard to the three friends example. Indeed, it is targeted to secondary school students, and would require more advanced problem-solving capabilities.

> Find a 10-digit number where the first digit is how many zeros there are in the number, the second digit is how many 1s in the number, and so on, until the 10th digit, which is how many 9s in the number.

From a human viewpoint, the understanding of the text and the problem modeling (steps 1–3) should be straightforward, while solving the problem (step 4) is more difficult. If a generate-and-test approach has to be avoided, a number of different reasoning actions have to be implemented by a human, and a more creative process is called in.

However, from a computer viewpoint, steps 1–3 are rather complex, while step 4 (solving the problem) might be definitely easier, for example by exploiting constraint-satisfaction techniques aimed at exploring large search spaces. In the following, we simply sketch how this problem can be faced by following the steps previously described.

Step 1: Understanding Text (and Pictures)

Understanding text here would amount to discovering the following:

> Find a number made of 10 *digits* (this is the goal).
>
> A *digit* is a number from 0 to 9.
>
> *"How many zeros"* in the number means counting the occurrences of zeros among the digits of the number itself. This information should be generalized for all the other digits by correctly understanding the "and so on, until …" part of the sentence.
>
> The occurrences of the i value in the list of digits should be equal to the ith digit. This step is again a consequence of a generalization from the "and so on, until…" part of the sentence.

Step 2: Identify the Modeling and Solving Technique

Given the structure of the problem, a constraint model and a constraint-solving technique based on backtrack search (with possible propagation) can be suitable.

Step 3: Identify Problem Components and Hidden Knowledge

Since the problem can be encoded into a constraint-satisfaction problem, the main components that have to be identified are variables, domains, and constraints.

> There are 10 variables, each one representing a digit of the number to be determined.
>
> Each digit has a position (from 0 to 9).
>
> Each variable has an integer domain between 0 and 9.
>
> As a constraint, the number of occurrences of i in the digits should be equal to the ith digit.

Step 4: Model the Problem

In a constraint model we have variables, domains, and constraints (expressed in a constraint language of choice). Using constraint logic programming, it is possible to model the problem directly as a set of 10 variables, each with domain [0 … 9], and each variable being subjected to a global constraint that captures the number of occurrences of the ith digit.

Step 5: Solve the Problem

The problem-solving process in this case exploits a constraint solver that uses propagation algorithms and search.

> The solution of this problem is the number 6210001000.

Notice that more information could be inferred. In our case such information is not needed for determining a solution, although this is not the case in general. For example, we would infer that:

> The sum of all digits is 10.
>
> The weighted sum of all digits where the weight is their position is again 10.
>
> The first digit cannot be a zero, and hence there is at least one or more zeros.
>
> No digit can take the value 9.

In the constraint-programming literature, these are called redundant constraints, namely constraints that are subsumed by other constraints of the problem. Depending on the technique used to solve the problem, redundant constraints should either be identified and removed from the model or should be left in the model. For example, in linear integer programming solvers redundant constraints do not bring any advantage to the solving algorithm and are removed in preprocessing. In constraint programming on finite domain solvers where propagation is not complete, instead, they might help reducing the search space and find a solution more quickly.

Example 3: Knights and Knaves

There are many mathematical puzzles that are indeed logic puzzles, and that are typically aimed at determining the truth value of a proposition, given some assertions whose truth value is a priori known. The following problem is taken from Smullyan (1978).

> There is an island where every inhabitants is either a knight or a knave (exclusive or). Knights always tell the truth, while knaves always lie. You are a tourist just arrived in the island, and you met two inhabitant A

and *B*. *A* says "I am a knave, or *B* is a knight." What are *A* and *B?*

Step 1: Understanding Text (and Diagrams)
In this case, identifying the problem's parts or components (for example, assertions, goals) amounts to discovering at least that:

There are two inhabitants *A* and *B*.

Either *A* is a knight, or *A* is a knave (exclusive or operator).

Either *B* is a knight, or *B* is a knave (exclusive or operator).

If *A* is a knight, then *A* always says the truth. Otherwise, he always lies.

If *B* is a knight, then *B* always says the truth. Otherwise, he always lies.

A says that *A* is a knave, or *B* is a knight (inclusive or operator).

Is *A* a knight or a knave? Is *B* a knight or a knave?

Step 2: Identify Modeling and Solving Techniques
Being a logical problem, a number of different approaches can be exploited for automatic theorem proving. This problem can be represented through propositional logic: as a consequence, techniques like, for example, truth tables or resolution might be suitable for solving the problem.

Step 3: Identify Problem Components
and Hidden Knowledge
Problem components are four propositional symbols

a_is_knight, a_is_knave

b_is_knight, b_is_knave

with obvious meaning. We have not identified any hidden knowledge useful to solve this problem.

Step 4: Frame the Model
The problem model is made of the logical, propositional sentences, together with the goals to be proved, namely the truth value of *a_is_knight* and *b-_is_knight*. Given the exclusive or between knights and knaves, we also have:

a_is_knight ex-or *a_is_knave*

b_is_knight ex-or *b_is_knave*

Finally, from the sentence, *A* says "I am a knave, or *B* is a knight." we should be able to understand also that:

a_is_knight → (*a_is_knave* ∨ *b_is_knight*)

a_is_knave → ¬ (*a_is_knave* ∨ *b_is_knight*)

Step 5: Solve the Problem
To solve this problem we can, for example, use resolution (Robinson 1965): taking as the goal the propositional fact that *a_is_knight*, it is possible to prove that *A* is a knight in two resolution steps. Similarly for the *B* inhabitant.

Example 4: Triangles

This example has a very simple text: How many triangles are in figure 2?

Notice that the accompanying picture is quite easy to understand: generally speaking, pictures can be much more complex, and could require a deeper understanding, as well understanding of the textual part. For example, understanding such an easy picture would require spatial interpretation and reasoning, the ability to cope with spatial and textual knowledge in a correlated manner, and some form of abstraction toward simple concepts such as triangles (Seo et al. 2014).

Example 5: The Castle Puzzle

This problem[21] is again defined partly through natural language text, and partly by means of an accompanying diagram: Given the picture in figure 3, which is bigger: the total area of blue or the total area of white?

Different ways for solving this puzzle exist. For example, one way would be to understand that:

The picture is a rectangle 8 units wide and 10 units height.

In the picture there are two areas: a blue one and a white one.

The blue part from the picture is composed by two triangles and a rectangle; from the rectangle a small square has been subtracted.

The question addressed by the problem is to determine which of the two areas is bigger.

Given this knowledge, and a background knowledge about simple geometry concepts (that is, how to compute areas of rectangles and triangles), it would be possible to compare the two areas and solve the problem by planning a set of actions such as determining the important data for each shape (number and type of different geometric shapes composing the blue or white areas, size of the edges for each rectangle, and edges of the triangles), compute the areas, and compare them. Note that for the computation of the blue area the machine should understand that it can be derived by summing the area of the rectangle and of the two triangles, and by subtracting from the total the area of the small white square. Solving the problem therefore means the ordered execution of the actions determined by such a plan.

Notice that we assume a background knowledge about the area of triangles and rectangles. Indeed, a human player might observe that the two triangles can be joined to form a rectangle (after a rotation/translation of one triangle). In such a case, to solve the puzzle it would suffice to count the number of square units: no knowledge about the computation of areas would be needed.

Another approach instead, would be to exploit the low-level picture representation: a computer, for example, might solve the problem by simply counting the number of pixels of one color. Such a solution would not require any understanding of what is in the image, and we believe it is not a solution exploitable by a human player.

Related Works and Challenges

Addressing the automatic solution of mathematical, geometric, logic, and science problems is an AI research area that dates back to 1960s. The first attempt to solve mathematical problems expressed in (a restricted) natural language is STUDENT (Bobrow 1964). Starting from that seminal work, a number of systems have been presented, focusing not only on the mathematical domain, but also in geometry, physics, mechanics, statistics, and other domains. For a review, the interested reader can refer to Mukherjee and Garain (2008). It is worth noticing that the initial difficulties encountered by the first attempts pushed the research toward the definition of systems specialized on restricted domains and problems. The advancements in AI witnessed in recent years have provided new fuel for novel and more general solutions and approaches.

The proposed challenge is grounded in a number of recent research works that address the automatic solution of mathematical, geometric, logic, and science problems/puzzles. Starting from information, explicitly stated in text and diagrams, they exploit deep reasoning and problem-solving capability to solve the problems, thus going beyond pure question-answering systems. In the following, we cite just a few works, with no claim of being exhaustive with regard to the state of the art.

The Allen Institute Aristo Challenge[22] is aimed at enabling a computer to pass elementary school science and math tests. Aristo makes advances in the areas of knowledge representation, modeling, reasoning, and language. With respect to science exams (Clark, Harrison, and Balasubramanian 2013), Aristo acquires and stores a vast amount of knowledge in computable form by using natural language parsing and processing. Starting from questions, diagrams, and answer options, it selects or generates the correct answer by combining knowledge retrieval, statistics, and inference methods. In addition, Aristo provides an extensive problem data set that can be used as a benchmark for further research.

A deeper emphasis on problem solving with respect to query answering can be found in the Euclid project[23] (Seo et al. 2014) of the Allen Institute. In Euclid, an end-to-end system that solves high school math and geometry problems is developed. In this context, specific systems providing modeling components have been presented:

GEOS (Seo et al. 2014, 2015) can be considered as the first automated system able to solve unaltered Scolastic Aptitude Test (SAT) problems related to geometry questions by combining text understanding and diagram interpretation.

ARIS (Hosseini et al. 2014) deals with arithmetic word problems that involve only sums or subtractions. In particular, ARIS analyzes each of the sentences in the problem statement to identify relevant variables and values and then maps this semantic information into equations that represent the problem.

Kushman and colleagues (2014) consider algebraic problems. A link between significant objects in the text and components of an algebraic equation is established by selecting and using suitable templates. The level of uncertainty in inferring this link is resolved by using probabilistic learning models. A

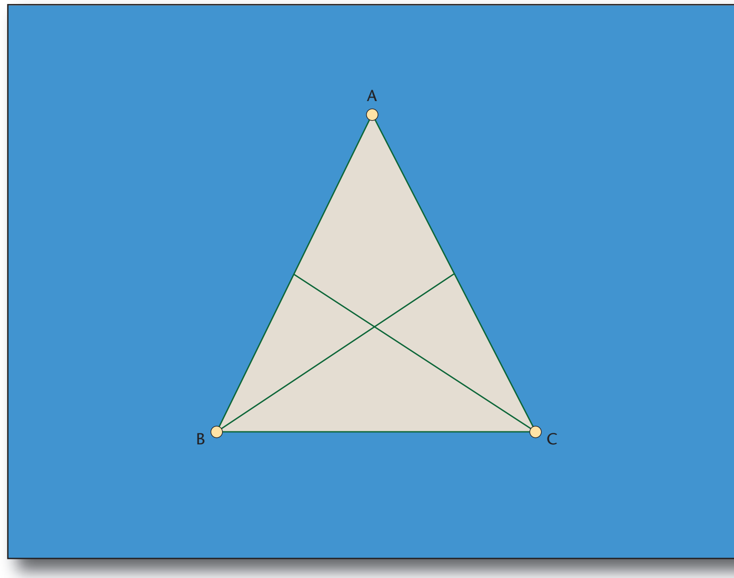

Figure 2. Triangles.

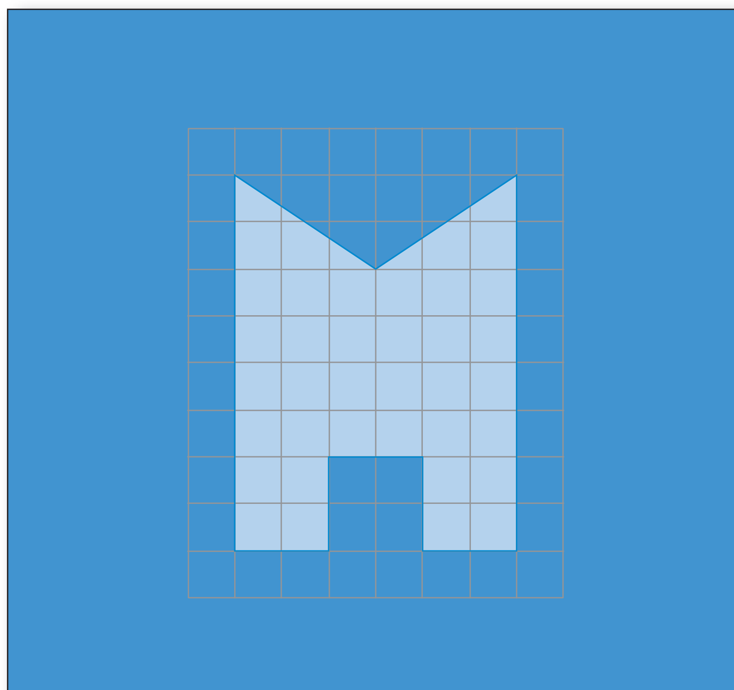

Figure 3. The Castle.

similar approach is pursued by Zhou, Dai, and Chen (2015), where quadratic programming is exploited to decide the best match between the text elements and the equation template components. Templates, in the form of equation tree structures, are exploited also by Koncel-Kedziorski and colleagues (2015), where integer linear programming is used for generating the space of trees and machine-learning techniques are used to select the best matching tree.

Following a semantic approach, Morton and Qu (2013) use a framework based on fuzzy logic and ontologies to solve mathematical word problems with the primary purpose of teaching users. To compute results, the search engine WolframAlpha and its integration within Mathematica are used.

In the paper by Shi et al. (2015) a new representation language (DOL) has been designed to bridge natural language text and math expressions. The parsing of natural language into DOL is performed using a context-free grammar. Then, a reasoning step recognizes the text portions of interest in the resolution, such as mathematical sentences, and these are subsequently turned into numerical expressions.

Inference (FOL) and statistical approaches for NLP, are exploited by Liang et al. (2016) to transform the text description in a tag-based form, and then into a first-order logic program. Inference is then performed to determine the solution.

Solving mathematical puzzles requires also comprehension of the role played by quantities in natural language. In the paper by Roy, Vieira, and Roth (2015) a series of features are described to support reasoning on quantities expressed in natural language. Two different numerical reasoning tasks are investigated and addressed: quantity entailment, and the problem of automatically understanding and solving elementary school math word problems.

The increasing interest in solving mathematical word problems is also witnessed by the creation of several data sets. Recently, Koncel-Kedziorski and colleagues (2016) proposed a framework for unifying all these sets into a single repository, with the possibility of extending it with new problem types and instances.

The Todai Robot Project[24] aims at creating an artificially intelligent agent obtaining a high score in the Japan National Center Test for University Admissions (McGoogan 2015, Strickland 2013), and passing the entrance exam of the University of Tokyo in 2021. Natural language comprehension is particularly stressed in order to support query answering and algebraic problems.

Some mathematical puzzles can be viewed as math word problems, while others can be assimilated to logic puzzles. An obvious way to solve logic puzzles is the use of theorem provers for first-order logic. The translation of a problem (expressed in natural language) to a logic semantics (exploitable in automated reasoning) is a hard and challenging problem for AI. In the paper by Lev et al. (2004) a method is proposed that uses an intermediate language called Semantic Logic, a general-purpose language with events, groups of variables, modal operators such as "necessary" and "possibly," and generalized quantifiers. The representation in the semantic logic can be translated in first-order logic and then solved by using a suitable theorem prover.

In a paper by Mitra and Baral (2015) the system LOGICIA is introduced. The authors claim it is the first system able to solve logical grid puzzles in a fully automated manner. Puzzles are addressed by translating them to answer set programming (ASP; Gelfond and Lifschitz [1991]), and then solved by an ASP solver.

A different, more general approach is presented by Forbus, Klenk, and Hinrichs (2009), where a central role is given to the analogy reasoning capability typical of human beings. Analogy is exploited in the Companions cognitive architecture for matching, memory retrieval, and generalization tasks. The approach has been applied in different domains, and in particular in the test-taking setting for physics problems. The interesting aspect is that the Companions architecture does not have, initially, any specific knowledge about physics, but exploits analogy to retrieve relevant solutions by looking into previously accumulated examples and extrapolating/adapting existing solutions to solve the new problem.

In table 1 we report some related works, classified with regard to important dimensions of the proposed challenge: the domain they address; whether they take as input only natural language descriptions or diagrams as well; whether they support problem description using "everyday language" or rather restrict problem description to a specific, limited language; and the solver used to solve the problem model (once a model has been extracted). We do not report the AI technique used by each system to identify or extract a model from the natural and diagram problem description: roughly speaking, the majority of the works exploit rule-based, with or without logic inference, and statistical approaches.

Our challenge differs from the mentioned works in many aspects. In order to address our challenge, in contrast with some works focused on SAT, no specific deep knowledge on some disciplines is required (for example, math, physics, geometry). Instead, commonssense reasoning capabilities are needed. Indeed, mathematical puzzles often require the ability to reason about space, time (qualitative and quantitative), causality, and events.

Moreover, a number of different puzzle categories should be addressed within the challenge: some categories can be assimilated to math word problems, some are more similar to logical puzzles, and some ask for a different variety of skills and reasoning abilities. Therefore, the challenge also requires taking into account the problem of establishing the appropriate modeling and problem-solving technique. In

Bibliographic Reference	Domain	Diagrams	Everyday language	Solving techniques
Clark, Harrison, and Balasub ramanian (2013)	Fourth Grade Science Problems	✓	✓	Query Answering and Inference
Seo et al. (2015)	Geometric Problems	✓		Algebraic/Geometric Solvers
Morton and Qu (2013)	Math Word Problems		✓	Algebraic Solvers
Shi et al. (2015)	Math Word Problems			Algebraic solvers
Roy, Vieira, and Roth (2015)	Math Word Problems		✓	Algebraic Solvers
Hosseini, et al. (2014)	Math Word Problems		✓	Algebraic Solvers
Kushman, et al. (2014)	Math Word Problems		✓	Algebraic Solvers
Zhou, Dai, and Chen (2015)	Math Word Problems		✓	Algebraic Solvers
Koncel-Kedziorski et al. (2015)	Math Word Problems		✓	Algebraic Solvers
Liang, et al. (2016)	Math Word Problems		✓	FOL Reasoner
Lev, et al. (2004)	Logical Puzzles		✓	FOL Reasoner
Todai Project[24]	University Access Tests	✓	✓	Query Answering and Algebraic Solvers
Forbus, Klenk, and Hinrichs (2009)	High-Achool Physics Tests	✓	✓	Analogical Reasoning
Mitra and Baral (2015)	Logical Grid Puzzles		✓	Answer Set Programming Solver

Table 1. Overview of Recent Related Works and Challenges.

turn, such choices might have consequences on the methods used for translating the input problem in a suitable form, as well as in the process of identifying problem components.

Discussion and Open Research Avenues

The proposed challenge is extremely hard if we consider its ultimate goal, that is, to set a real competition between computers and humans on mathematical puzzles. However, it allows for a number of different intermediate steps with increasing difficulties, thus providing a nice playground for AI researchers. Having in mind the challenge, and for the sake of discussion, we cite a few research directions, knowing that being exhaustive would be impossible.

Means for Evaluating the Challenge

Given the breadth and the complexity of the challenge, the reader might question whether it is broad enough to foster research advances with regard to existing state-of-the-art solutions. In this respect, an important feature of research challenges is the possibility of confronting them in a stepwise fashion, thus providing short-term goals as well as long-term ones. The stepwise nature of the proposed challenge is threefold. First, we can approach problems at increasing complexity levels: among mathematical puzzles of the primary school, we could first approach simpler then more complex problems, providing different scores in the competition. Second, we can proceed gradually, starting from mathematical puzzles that can be solved using a single technique (for example constraint solving) to an ensemble of solution techniques that should be selected by the agent.

Third, the stepwise approach to autonomy could help to invesigate collaboration between humans and machines in the problem-solving activity, in line with the concept proposed by Barbara Grosz and colleagues (Grosz, Hunsberger, and Kraus 1999; Gal et al. 2010).

Following these lines of development, tests devoted to assess results and advancements can be designed to measure the effectiveness of the problem-solving process and the level of autonomy reached by the agent: specifically, each step and each advancement should be identified and measured. Firstly we could consider a single-solution technique and simply count the number of correctly solved problems. In this case, given that the solving technique is fixed, we could consider how good and how appropriate the devised models are.

Enlarging the scope to multiple solution methods, again the number of correctly solved problems can be counted, but also how good and how appropriate the devised solution techniques are can be evaluated. Of course, we could also consider how fast the automatic solution process is.

Moving on the road to full autonomy, the challenge can be approached by using intermediate steps that require interactions between a machine and humans. The level of interaction basically measures the level of human intervention in the solution process. How effective is the interaction and how much human intervention is needed are certainly two ways of measuring the level of achieved autonomy.

Finally only at a future stage will we be able to evaluate the competition against human players. Competitions require an additional set of competencies on the computer's side, addressing the strategy for approaching the competition problems. Indeed, humans taking part in the mathematical puzzle competitions are evaluated on a number of criteria, such as the time used for solving problems, the correctness of the solution, as well as the complexity of the problem itself (more complex problems bring a higher score to the human player). To win such a competition, the computer would need a further reasoning layer, focused on metaknowledge on its own solving capabilities, as well as strategies for selecting which problems to solve (and in which order), and how much time to allocate to each problem.

Beyond the Turing Test

A number of different challenges have been proposed in AI in order to establish the level of intelligence exhibited by machines. The most famous one is surely the Turing test. To cite some recent ones, we have the Coffee Test (a machine preparing a coffee in an average American home (Wozniak and Moon 2007; Adams et al. 2012), the Robot College Student Test (a machine enrolling, studying, passing university courses, and obtaining a degree, by Goertzel[25]), and the Employment Test (a machine working in an economically important job; Nilsson [2005]).

Within the AAAI 2015 conference, a workshop titled Beyond the Turing Test[26] was aimed at establishing a structured setting for holding Turinglike competitions. Inspired from that event, an entire issue of *AI Magazine* (spring 2016) was dedicated to a very interesting discussion on new proposals and tests for measuring the intelligence of machines.

Here, we just stress that our proposed challenge has a number of connections with and similarities to some of the new tests presented in the journal issue. The test proposal more similar to our challenge is the one based on standardized tests by Clark and Etzioni (2016), even if our challenge has a specific attention to the problem-solving skills. Moreover, our challenge shares the need of commonsense reasoning as proposed by Davis (2016) for the SQUABO-Basic test; it has a multimodal dimension since it takes into account visual and textual problem description, which is advocated as important by Zitnick et al. (2016); it provides many (but not all) of the dimensions as defined by Adams, Banavar, and Campbell (2016) for the I-athlon Turing test (namely, diagram understanding, natural language understanding, collaboration, competition, reasoning, reasoning under uncertainty, creativity, learning, planning, and commonsense physics). In this light, one of the strengths of our challenge is that solving mathematical puzzles advocates for a number of different capabilities or intelligence dimensions that are typical of human intelligence and that humans usually exploit in an integrated manner. However, mathematical puzzles do not require specialized knowledge, since the targets are primary school students. Instead, it is the conjunction of the many different intelligence dimensions that makes mathematical puzzles and our challenge so interesting and stimulating for AI.

The Turing test has been recently extended also to multiagent systems (Grosz 2012, 2013), where it "is imaginable that a computer (intelligent agent) team member could behave … in such a way that people on the team will not notice it is not human."

Including human actors in the process can be considered a way of evaluating, in a collaborative setting, the trade-off between interactive and noninteractive approaches to artificial intelligence and provides useful insights on the level of human intervention needed and autonomy reached by the problem solver.

Natural Language Processing

Text understanding is a fundamental step in this challenge, as well as in the other mentioned challenges, that could be supported by semantic tools such as WordNet.[27]

Mathematical and algebraic problems, having a limited interpretation domain and being formal with few or no ambiguities, lend themselves to being successfully addressed with semantic-based NLP approaches. However, some mathematical problems are described using everyday language and refer to everyday situations/contexts. This is the typical case of mathematical puzzles — deeply understanding them implies a complex semantic analysis, as well as the integration of different kind of (background) knowledge and inference techniques.

The complexity of the approach could be reduced if the text-understanding process can be goal driven, and if the problem-solving techniques can be taken into account. For example, in the three friends puzzle, the fact that the three children are friends is useless from a problem-solving perspective. The same model could have been extracted in case Frank, Lucy, and Jacob were not friends, and if they were dogs or trees. Moreover, if we decide to solve the problem through constraint satisfaction techniques, the language-understanding process could focus on the identification of decision variables, domains, and constraints within the problem description, ignoring useless information. Therefore, we focus on deriving structured information suitable for reasoning, avoiding the need of a complete semantic understanding.[28]

Multimodal Information Extraction

A deep understanding of text and images (and in general of multimodal problem descriptions) is needed, possibly in a tight integrated way. The natural language processing and image processing communities have recently achieved noteworthy results (each in its own field). However, extracting the right knowledge from multimodal descriptions still offers a huge opportunity for research: it clearly enables a better representation of contents and concepts, even if it introduces a further level of complexity. Indeed, such integration requires a semantically valid, unifying paradigm and language for knowledge representation and extraction (Pastra and Wilks 2004).

Commonsense Knowledge and Deep Understanding

Natural language texts invariably assume some implicit knowledge. In particular, in the mathematical puzzles domain, a lot of background knowledge and commonsense reasoning is usually not present in the problem description, but has to be made explicit in order to solve the problem. For example, precise understanding of semantic phenomena like modals and quantifiers, time and space relations, could be needed.

To discover hidden knowledge in texts, a large commonsense knowledge base is needed. A number of different initiatives in the AI community are going on for building such a common knowledge base. For example, the Open Mind Common Sense project and the ConceptNet[29] are crowdsourcing the contributions of thousands of people across the world, making such information available in a number of different formal languages. Another, very famous attempt to build up a large, commonsense ontology and knowledge base is the Cyc project (Lenat 1995), started in 1984 and currently still being developed and supported.

Machine Learning for Knowledge Extraction and Problem Solving

Knowledge extraction, model revision, identification of the solving technique (to cite some) are all tasks that could greatly benefit from machine learning, a discipline that has obtained very valuable results in the last decades. For example, some recent works exploit deep networks for extracting specific knowledge (Lippi and Torroni 2015) for reasoning. However, they still lack many of the functionalities needed for achieving our goal entirely. In particular, as pointed out in a more general context by Gary Marcus,[30] we need to identify and represent causal relationships, integrating commonsense knowledge about math, logic, geometry, and information about what objects are, what they are for, and how they are typically used. Hence, machine learning (and deep learning in particular) is just one element in a big plethora of artificial intelligence techniques for knowledge representation and reasoning that has to be suitably used and integrated.

Within the challenge scenario, we foresee two main roles for machine learning. First, it could be used to improve each step of the process defined in figure 1; second, it could help the entire process. With respect to the first role, improving the single steps, machine learning could be exploited to learn how to classify for example natural language structures into model components, how to select a proper solution technique from a portfolio, and others.

Concerning the second role of machine learning, we foresee a significant contribution provided by case-based reasoning to define a similarity measure among problem descriptions. If we assume that similar problems can be tackled with the same technique, we could exploit the experience obtained in the solution of one problem for those that are considered close to it. This would help shape the entire process and make it more efficient and effective.

Specialized Versus General Problem Solvers

An important feature characterizing our challenge is the emphasis on problem-solving skills. This issue is recognized by the AI community as a fundamental goal toward the evolution of autonomous intelligent agents. For this purpose, we have introduced the challenge by proposing also a possible, limited number of steps and interactions/loops: specifically, we devoted one step to identifying a possible model of the problem, and a further step to choosing the best solving technique. This requires some sort of metaknowledge for reasoning, in order to select the specific solving technique. For example, if we opt for a CSP, two influential methods, developed mainly in the field of constraint satisfaction and optimization, aim to automate this process (Xu, Hoos, and Leyton-Brown 2010): automated algorithm configuration and portfolio-based algorithm selection. The former has the advantage of requiring no domain knowledge, but produces only a single solver; the latter exploits per instance variation, but requires a set of relatively uncorrelated candidate solvers. While these techniques are mainly devoted to achieve better performances (Hutter et al. 2014), the purpose here is to select a solving technique that is more suitable for the devised model.

Implicitly, in the previous discussion we took the assumption of having a plethora of different, machine-oriented specialized solving techniques and tools to choose from, following the big switch statement approach, where "… separate narrowly specialized programs corresponding to the individual tasks are combined together in a simplistic harness" (Adams et al. 2012).

The "big switch statement" can be effective in a number of different problems and contexts, and results have been obtained by exploiting this approach in a number of different AI applications. However, it has also been criticized, as for example in the paper by Nilsson (2005), where:

> I think AI should strive toward building a small number of general-purpose machines that are able to learn and to be taught skills (and to improve on them) rather than programming a much larger number of machines each possessing a different individual skill set from the start.

Clearly, Nilsson puts a bigger emphasis on the artificial general intelligence (AGI), where the cognitive and learning processes are one of the main features to stress. Indeed, tackling the challenge also means confronting a number of important competency areas usually associated with human-level general intelligence, such as memory, communication, learning, and quantitative reasoning (Adams et al. 2012). Therefore, we believe that our challenge can be interpreted also as a possible scenario within the context of the AGI landscape.

Computational Thinking Versus AI Thinking

The general steps to be performed by an end-to-end problem solver are very related to some key ingredients of computational thinking:[31] decomposition — "breaking down data, processes, or problems into smaller, manageable parts"; pattern recognition — "observing patterns, trends, and regularities in data"; abstraction — "identifying and extracting relevant information to define main idea(s)"; and algorithm design — "creating an ordered series of instructions for solving similar problems or for doing a task."

While computational thinking (Wing 2006) is oriented toward data processing and algorithms, in the context of our challenge there is a bigger emphasis toward the problem-solving aspects and the more general process of artificial intelligence thinking (Zeng 2013). Compared with computational thinking, "AI thinking goes beyond the algorithm-based perspectives and emphasizes items such as how to leverage knowledge bases and case bases in problem solving, how to capture and reason about commonsense, how to enable processing of semantics and contexts, and how to deal with unstructured data, among others."

Conclusions

Computers beating humans in mathematical puzzles that are described in terms of text and diagrams is the challenge proposed in this paper. Even if it is still far from being achieved, we believe that scientific research in this direction can provide important improvements in many AI disciplines and reduce modern AI fragmentation. In the intermediate steps, intelligent agents and students could work together, with different skills and expertise, to solve mathematical puzzles. Beside bringing insights to collaborative interactions between humans and machines, this would also be beneficial for the dissemination of artificial intelligence in educational settings. Finally, this challenge could be used to disseminate AI results within the general public, business, and policy makers, as the challenge is easy to understand but poses, at the same time, a number of deep and complex issues.

Notes

1. www.loebner.net/Prizef/loebner-prize.html.
2. www.aerialroboticscompetition.org.
3. www.darpa.mil/About/History/Archives.aspx.
4. www.robocup.org.
5. www.cs.miami.edu/ tptp/CASC/.
6. www.evalita.it.
7. www.grappa.univ-lille3.fr/icga/game.php?id=1.
8. www.marioai.org.
9. www.platformersai.com.
10. games.stanford.edu/index.php/homepage.
11. content.time.com/time/magazine/article/0,9171,984305,00.html#ixzz1DyffA0Dl.
12. Quoted from Noam Chomsky, *Powers and Prospects: Reflections on Human Nature and the Social Order.*
13. Go Grandmaster Lee Sedol Grabs Consolation Win Against Google's AI, https://www.wired.com/2016/03/go-grandmaster-lee-sedol-grabs-c.
14. www.robocup.org/about-robocup/objective.
15. www.robocup.org/about-robocup/objective.
16. www.ibm.com/watson.
17. Freely translated from the introduction of *Les jeux mathématiques*, Michel Criton, PUF Edition, 1977.
18. matematica.unibocconi.it/giochi-matematici.
19. www.animath.fr/spip.php?rubrique42.
20. Note that, for the sake of this challenge, we restrict our focus to mathematical puzzles for which a solution always exists and is unique.

21. Freely inspired by a similar puzzle by Peter Grabarchuk, www.peterpuzzle.com/web/MathsIsFun.htm.

22. allenai.org/aristo.html.

23. allenai.org/euclid.html.

24. 21robot.org.

25. www.newscientist.com/article/mg21528813.600-what-counts-as-a-conscious-thinking-machine.

26. www.math.unipd.it/ frossi/BeyondTuring2015.

27. https://wordnet.princeton.edu.

28. See the Workshop on reasoning with the text projects.ict.usc.edu/rwt2011/ for a general discussion.

29. conceptnet5.media.mit.edu.

30. G. Marcus, Is "Deep Learning" a Revolution in Artificial Intelligence?' *The New Yorker,* 25 November 2012.

31. www.google.com/edu/resources/programs/exploring-computational-thinking/index.html#!ct-overview.

References

Adams, S.; Arel, I.; Bach, J.; Coop, R.; Furlan, R.; Goertzel, B.; Hall, J. S.; Samsonovich, A. V.; Scheutz, M.; Schlesinger, M.; Shapiro, S. C.; and Sowa, J. F. 2012. Mapping the Landscape of Human-Level Artificial General Intelligence. *AI Magazine* 33(1). doi.org/10.1609/aimag.v33i1.2322

Adams, S. S.; Banavar, G.; and Campbell, M. 2016. I-athlon: Towards a Multidimensional Turing Test. *AI Magazine* 37(1): 78–84. doi.org/10.1609/aimag.v37i1.2643

Bobrow, D. G. 1964. Natural Language Input for a Computer Problem Solving System. Technical Report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Cambridge, MA. (hdl.handle.net/1721.1/6903)

Chomsky, N. 1999. *Powers and Prospects: Reflections on Human Nature and the Social Order.* Boston: South End Press.

Clark, P. 2015. Elementary School Science and Math Tests as a Driver for AI: Take the Aristo Challenge! In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence,* 4019–4021. Palo Alto, CA: AAAI Press.

Clark, P., and Etzioni, O. 2016. My Computer Is an Honor Student — But How Intelligent Is It? Standardized Tests as a Measure of AI. *AI Magazine* 37(1): 5–12. doi.org/10.1609/aimag.v37i1.2636

Clark, P.; Harrison, P.; and Balasubramanian, N. 2013. A Study of the Knowledge Base Requirements for Passing an Elementary Science Test. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction,* AKBC '13, 37–42. New York, Association for Computing Machinery. doi.org/10.1145/2509558.2509565

Davis, E. 2016. How to Write Science Questions That Are Easy for People and Hard for Computers. *AI Magazine* 37(1): 13–22. doi.org/10.1609/aimag.v37i1.2637

Forbus, K. D.; Klenk, M.; and Hinrichs, T. R. 2009. Companion Cognitive Systems: Design Goals and Lessons Learned So Far. *IEEE Intelligent Systems* 24(4): 36–46. doi.org/10.1109/MIS.2009.71

Gal, Y.; Grosz, B. J.; Kraus, S.; Pfeffer, A.; and Shieber, S. M. 2010. Agent Decision-Making in Open Mixed Networks. *Artificial Intelligence* 174(18): 1460–1480. doi.org/10.1016/j.artint.2010.09.002

Garner, M. 1981. *Mathematical Puzzle Tales.* New York: Clarkson N. Potter, Inc.

Gelfond, M., and Lifschitz, V. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9(3/4): 365–386. doi.org/10.1007/BF03037169

Grosz, B. J. 2012. What Question Would Turing Pose today? *AI Magazine* 33(4): 73–81.

Grosz, B. J. 2013. A MultiAgent Systems "Turing Challenge." In *International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '13,* 3–4. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Grosz, B. J.; Hunsberger, L.; and Kraus, S. 1999. Planning and Acting Together. *AI Magazine* 20(4): 23–34. doi.org/10.1609/aimag.v33i4.2441

Hosseini, M. J.; Hajishirzi, H.; Etzioni, O.; and Kushman, N. 2014. Learning to Solve Arithmetic Word Problems with Verb Categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 523–533. Stroudsburg, PA: The Association for Computational Linguistics. doi.org/10.3115/v1/D14-1058

Hutter, F.; Xu, L.; Hoos, H. H.; and Leyton-Brown, K. 2014. Algorithm Runtime Prediction: Methods and Evaluation. *Artificial Intelligence* 206: 79–111. doi.org/10.1016/j.artint.2013.10.003

Koncel-Kedziorski, R.; Hajishirzi, H.; Sabharwal, A.; Etzioni, O.; and Ang, S. D. 2015. Parsing Algebraic Word Problems into Equations. *Transactions of the Association for Computational Linguistics* 3: 585–597.

Koncel-Kedziorski, R.; Roy, S.; Amini, A.; Kushman, N.; and Hajishirzi, H. 2016. MAWPS: A Math Word Problem Repository. In *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 1152–1157. Stroudsburg, PA: The Association for Computational Linguistics. doi.org/10.18653/v1/n16-1136

Kushman, N.; Zettlemoyer, L.; Barzilay, R.; and Artzi, Y. 2014. Learning to Automatically Solve Algebra Word Problems. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, 271–281. doi.org/10.3115/v1/p14-1026

Lenat, D. B. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11): 32–38. doi.org/10.1145/219717.219745

Lev, I.; Maccartney, B.; Manning, C. D.; and Levy, R. 2004. Solving Logic Puzzles: From Robust Processing to Precise Semantics. In *Proceedings of the 2nd Workshop on Text Meaning and Interpretation*, ACL-04. Stroudsburg, PA: The Association for Computational Linguistics. doi.org/10.3115/1628275.1628277

Liang, C.; Hsu, K.; Huang, C.; Li, C.; Miao, S.; and Su, K. 2016. A Tag-Based English Math Word Problem Solver with Understanding, Reasoning and Explanation. In *Proceedings of the Demonstrations Session, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 67–71. Stroudsburg, PA: The Association for Computational Linguistics.

Lippi, M., and Torroni, P. 2015. Context-Independent Claim Detection for Argument Mining. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence,* IJCAI 2015, 185–191. Palo Alto, CA: AAAI Press.

McGoogan, C. 2015. This Japanese AI Is Competing for Top University Places. *Wired.* Tuesday, 17 November. (www.wired.co.uk/article/ai-passes-japanese-university-entrance-exam).

Mitra, A., and Baral, C. 2015. Learning to Automatically Solve Logic Grid Puzzles. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* EMNLP 2015, 1023–1033. Stroudsburg, PA: The Association for Computational Linguistics. doi.org/10.18653/v1/D15-1118

Morton, K., and Qu, Y. 2013. A Novel Framework for Math Word Problem Solving. *International Journal of Information and Education Technology* 3(1) (February). doi.org/10.7763/ijiet.2013.v3.240

Mukherjee, A., and Garain, U. 2008. A Review of Methods for Automatic Understanding of Natural Language Mathematical Problems. *Artificial Intelligence Review* 29(2): 93–122. doi.org/10.1007/s10462-009-9110-0

Nilsson, N. J. 2005. Human-Level Artificial Intelligence? Be Serious! *AI Magazine* 26(4): 68–75. doi.org/10.1609/aimag.v26i4.1850

Pastra, K., and Wilks, Y. 2004. Vision-Language Integration in AI: A Reality Check. In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004. Frontiers of Artificial Intelligence and Applications 285*, 937–944. Amsterdam, The Netherlands: IOS Press.

Robinson, J. A. 1965. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM* 12(1): 23–41. doi.org/10.1145/321250.321253

Roy, S.; Vieira, T.; and Roth, D. 2015. Reasoning About Quantities in Natural Language. *Transactions of the Association for Computational Linguistics* 3: 1–13.

Seo, M. J.; Hajishirzi, H.; Farhadi, A.; and Etzioni, O. 2014. Diagram Understanding in Geometry Questions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence,* 2831–2838. Palo Alto, CA: AAAI Press. doi.org/10.18653/v1/D15-1171

Seo, M. J.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; and Malcolm, C. 2015. Solving Geometry Problems: Combining Text and Diagram Interpretation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* EMNLP 2015, 1466–1476. Stroudsburg, PA: The Association for Computational Linguistics.

Shi, S.; Wang, Y.; Lin, C.; Liu, X.; and Rui, Y. 2015. Automatically Solving Number Word Problems by Semantic Parsing and Reasoning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* EMNLP 2015, 1132–1142. Stroudsburg, PA: The Association for Computational Linguistics. doi.org/10.18653/v1/D15-1135

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* 529: 484–503. doi.org/10.1038/nature16961

Smullyan, R. 1978. What Is the Name of This Book? — The Riddle of Dracula and Other Logical Puzzles. Engelwood Cliffs, NJ: Prentice-Hall.

Strickland, E. 2013. Can an AI Get into the University Of Tokyo? *IEEE Spectrum News*, 21 August. (spectrum.ieee.org/robotics/artificial-intelligence/can-an-ai-get-into-the-university-of-tokyo).

Togelius, J.; Shaker, N.; Karakovskiy, S.; and Yannakakis, G. N. 2013. The Mario AI Championship 2009–2012. *AI Magazine* 34(3): 89–92. doi.org/10.1609/aimag.v34i3.2492

Wing, J. M. 2006. Computational Thinking. *Communications of the ACM* 49(3): 33–35. doi.org/10.1145/1118178.1118215

Wozniak, S., and Moon, P. 2007. Three Minutes with Steve Wozniak. *ABC News*, July 20. (abcnews.go.com/Technology/PCWorld/story?id=3396207)

Xu, L.; Hoos, H.; and Leyton-Brown, K. 2010. Hydra: Automatically Configuring Algorithms for Portfolio-Based Selection. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (AAAI 2010). Palo Alto: AAAI Press.

Yannakakis, G., and Togelius, J. 2014. A Panorama of Artificial and Computational Intelligence in Games. *IEEE Transactions on Computational Intelligence and AI in Games* 7(4): 317–335. doi.org/10.1109/TCIAIG.2014.2339221

Zeng, D. 2013. From Computational Thinking to AI Thinking. *IEEE Intelligent Systems* 28(6): 2–4. doi.org/10.1109/MIS.2013.141

Zhou, L.; Dai, S.; and Chen, L. 2015. Learn to Solve Algebra Word Problems Using Quadratic Programming, 817–822. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* EMNLP 2015. Stroudsburg, PA: The Association for Computational Linguistics. doi.org/10.18653/v1/D15-1096

Zitnick, C. L.; Agrawal, A.; Antol, S.; Mitchell, M.; Batra, D.; and Parikh, D. 2016. Measuring Machine Intelligence Through Visual Question Answering. *AI Magazine* 37(1): 63–72. doi.org/10.1609/aimag.v37i1.2647

Federico Chesani, PhD, is a research assistant at the Department of Computer Science and Engineering at the University of Bologna. His research interests cover rule-based systems, complex event processing, business process management, and theoretical and practical aspects of logic programing with a focus on abductive reasoning. He is author of more than 70 papers in international conferences and journals and has been involved in several national and international projects.

Paola Mello, PhD, is a full professor at the Department of Computer Science and Engineering of the University of Bologna. Her research activity focuses on artificial intelligence, knowledge representation and reasoning, logic programming, multiagent systems, applications of expert systems with particular emphasis on medical domain. Moreover, she is currently active on formal specification and on the automatic verification of (interaction) protocols, workflow patterns, medical guidelines, and web services compositions through the use of declarative languages. Mello is author of more than 200 scientific papers, and she has been involved in a number of projects, mainly in the context of computational logic. From 2010 to 2014 she was the president of the Italian Association for Artificial Intelligence. From 2012 to 2015 she was the head of the Department of Computer Science and Engineering at the University of Bologna.

Michela Milano, PhD, is an associate professor at the Department of Computer Science and Engineering at the University of Bologna. Her research interests cover theoretical and practical aspects of constraint reasoning and optimization. Application areas include planning and scheduling, computational sustainability, embedded system design, energy systems, and smart cities. Milano is author of more than 130 papers and is editor in chief of the *Constraints Journal* and area editor for *INFORMS Journal on Computing and Constraint Programming Letters*. She was program chair of CPAIOR 2005, CPAIOR 2010, and CP2012. Milano is the coordinator of the EU-funded ePolicy project and is participating to two other EU project on smart cities.