Juris & informatics Collaboration
Striving to Expedite Civil Litigation

Artificial Intelligence
Reasons about the Courses of Trials

There is a need in the judicial world for the trial process to be accelerated. Research is underway to increase trial efficiency by applying logical processing to some civil litigation processes. Trial and legal documents contain expressions and words used by real people for communicating, and may seem quite removed from artificial intelligence and logic programming, but in fact they share much in common. We heard from Professor Ken Satoh, of the Informatics Principle Research Division, regarding the fruits of his research, and its future potential.

Avoiding the vagueness inherent in natural language through logic programming

Logic programming, explained simply, is the implementation of "mathematical logic" on computers. It is used as a description language for artificial intelligence. Professor Ken Satoh, of the NII Principles of Informatics Research Division, says, "The legal and judicial world has, so far, used natural language exclusively for all issues. I thought that adding a logical perspective, using symbolic notation and formulas, and applying logic programming, would result in the creation of new knowledge, and contribute to actual trials."

Ken Satoh
Professor
Principles of Informatics Research Division

There are three primary advantages of applying logical analysis to law. The first is that it enables one to discover implicit assumptions buried in legal documents. When using the logical approach, all assumptions must be explicit in order to produce results. However, because legislation is written in natural language, there are matters which are implied. These can be made explicit by logical expression. The second advantage is that if legal documents are logically analyzed, and it becomes possible to express them in strict logical forms, it will become possible to handle legal concepts themselves via computer. This will make it possible to resolve even more complex legal problems, and detect inconsistencies in the law. The third advantage is that by separating logical problems and legal problems, legal scholars will be able to concentrate exclusively on legal problems.

In other words, if logical approaches are applied to jurisprudence, things which were unclear will come to light, and previously complex work can be made more efficient.

The "presupposed ultimate fact theory" process which proves assertions

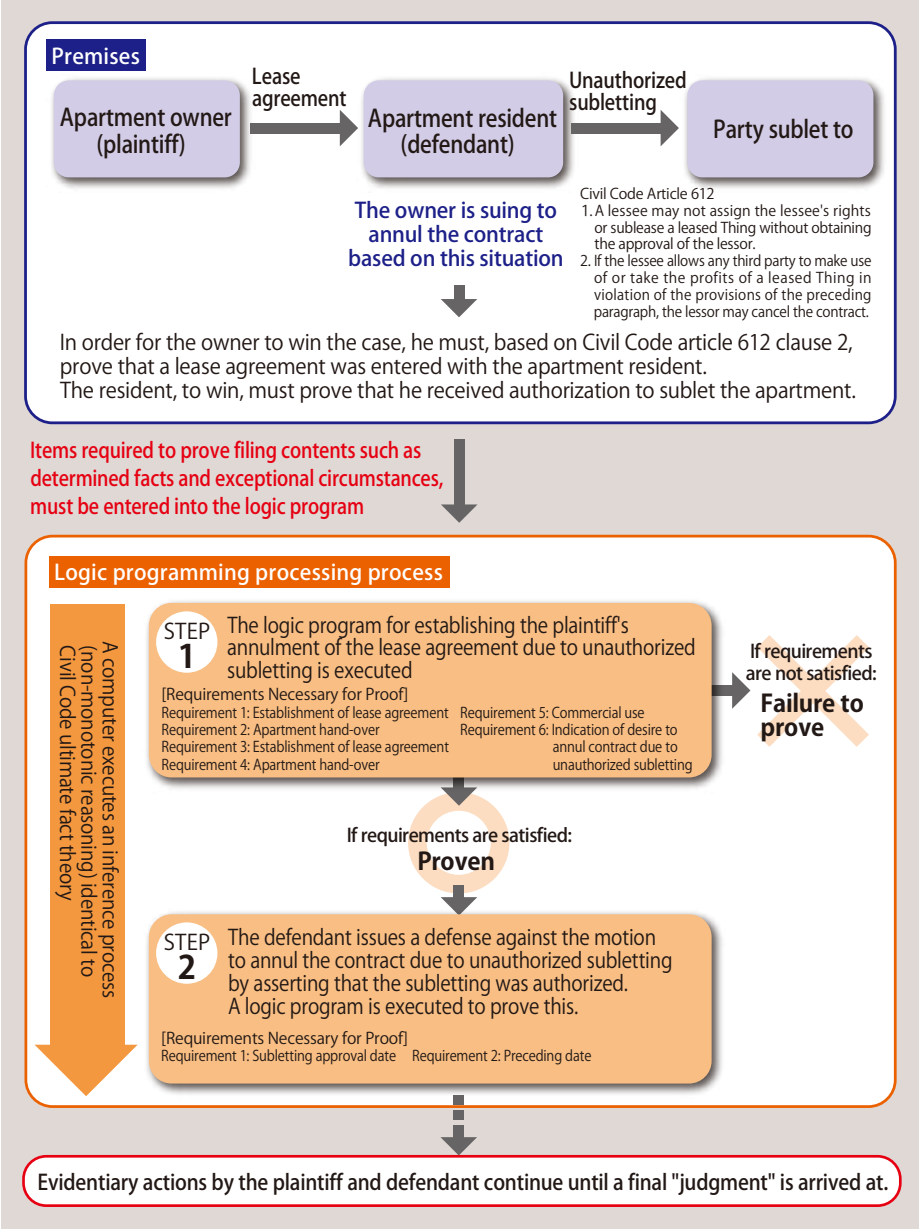
Professor Satoh is currently researching the deployment of logic programming in civil suits.

In civil suits, judges arrive at their conclusions by comparing the assertions of plaintiffs and defendants against substantive laws, such as civil and trade law. Articles of substantive law specify legal effects which pertain to specific facts, but do not go as far as specifying who, proving what facts, would be sufficient to make a determination of whether or not there were specific legal effects. For example, in a case wherein payment for purchased goods has not been made, the party wishing the payment to be made actively presents evidence that there was a purchasing contract, but the law does not clearly specify which claims, made by which parties, are sufficient to prove facts. The theory of presupposed ultimate fact is used by judges as a guideline when trying to determine, in a case such as the above, which party, the plaintiff or the defendant, must prove the case's facts, and what conclusion the judge should arrive at. Professor Satoh says, "this presupposed ultimate fact theory process is extremely similar to the non-monotonic reasoning(*) I have been researching, so I recognized the significance of developing corresponding logic programming."

Using AI to simulate uncertain aspects of judicial cases

The logical concept of "non-monotonicity" refers to the possibility that a previous conclusion may be retracted due to the addition of new information. Much of formal

Figure Logic Programming Application Example "Trial Regarding Contract Annulment Due to Unauthorized Subletting"



logic is monotonic — that is, even if new information is added, existing information is not eliminated. Non-monotonicity can be used to establish hypotheses given incomplete information. For example, assume that there is a hypothesis: "Cars drive on streets". If nothing is added to this statement, a car would be presumed to drive on the street. However, if one later found out that the car in question were a racecar, such as an F1 car, the conclusion that it drives on streets would be retracted. In cases such as this, where there is incomplete information, non-monotonic reasoning provides a logical description how to make deductions.

In that sense, trials are, as a rule, "non-monotonic". If they were monotonic, the results of a case once won would never be overturned. However, in reality, even if evidence presented in a first trial results in victory, new appealing evidence presented at an appeal trial might result in defeat, a non-monotonic change in the trial's verdict. Professor Satoh's belief was that the aforementioned presupposed ultimate fact theory is a formulation for rational deduction given incomplete information, and that as such non-monotonic reasoning could be applied. This led to his creation of ultimate fact theory logical programming like that shown in the diagram.

The diagram shows a simulation of a trial regarding the annulment of a contract due to unauthorized subleasing. A certain apartment owner claimed to the court that "a resident with a rental contract subleased the apartment to a third party without permission, so I would like the contract to be annulled." In this case, the owner must prove that he let the apartment to the resident in accordance with the proper procedures, and that the resident sublease the apartment to a third party. He does not, however, have to prove that he did not authorize the subleasing. Instead, the resident is responsible for proving that the owner gave his approval for the subleasing. Non-monotonic reasoning-based logical programming is used to have a computer carry out the same inference process as used in presupposed ultimate fact theory. Describing essential items, such as determined facts, precedents, and exceptions, results in a system which can be used to arrive at a conclusion — that is, whether the contract can or cannot be annulled.

Working towards the creation of the new academic field of "juris-informatics"

The goal of Professor Satoh's research is the creation of a new academic discipline, "juris-informatics", combining jurisprudence and informatics. However, he also points out that "logic programming cannot be applied to all aspects of laws and trials." For example, in a criminal case, when determining intent to kill, a conclusion must be reached taking into account an overall view, including how the incident occurred, how the murder weapon was used, how the defendant behaved after the incident, and the like. For this type of fact-finding, and interpretation based on common sense, logic programming cannot be used to reach optimal conclusions. Instead, human insight and judgment must be relied on.

In other words, there are elements of trials which require that conclusions be drawn only by people. As such, the goal of present research into juris-informatics is not "creation of artificial intelligence which will make all case judgments", but "application to logical work which is difficult using natural language". By making a clear delineation between human and computer roles, logic programming and artificial intelligence in the law field can achieve well-defined research goals.

(Written by Junichi Morimoto)

*Non-monotonic reasoning: Logic in which conclusions and results may change when additional inferences are added. For example, the verdicts may differ between first and second trials if additional evidence is discovered. In monotonic reasoning, conclusions and results do not change, even if new logical formulas are added.