CLAIMS:

1. An online webserver for automated legal advice, the online webserver comprising:

a legal text data input to receive legal text and splitting the legal text into multiple text blocks;

a language processor to perform natural language processing on each of the multiple text blocks to extract atomic literals that correspond to terms in the legal text and that are shared across logic expressions for different text blocks;

a mapper to map the text blocks into a logic expression of the atomic literals;

a logic engine to evaluate the logic expression based on one or more assignments of the atomic literals;

a user interface module to

create a user interface comprising multiple user input elements, each of the multiple user input elements being associated with one of the atomic literals,

monitor user interaction with the user interface to detect user interaction with one of the multiple user input elements,

upon detection of user interaction with one of the multiple user input elements, creating an assignment of the atomic literal associated with the one of the multiple user input elements and calling the logic engine to evaluate the logic expression based on the created assignment, and

change the user interface based on evaluating the logic expression to thereby provide legal advice in the form of remaining atomic literals; and

a network connector to communicate with a client device by receiving user input from the client device and provide the user interface to be displayed on the client device.

2. The online webserver of claim 1, wherein the language processor is further configured to determine a parse tree of each of the text blocks and the mapper is configured to map the text blocks into the logic expression based on the parse tree.

3. The online webserver of claim 1 or 2, wherein the mapper is further configured to map the text blocks into the logic expression based on heuristics.

4. The online webserver of any one of the preceding claims, wherein the mapper is further configured to search for an override conjunction (‘unless’) in a text block and upon finding an override conjunction in the text block at a find position, split the text block at the find position into an earlier first text sub-block before the find position and a second text sub-block after the find position.

5. The online webserver of claim 4, wherein the mapper is further configured to map the first sub-block into a first logic expression and map the second text sub-block into a second logic expression that overrides the first logic expression.

6. The online webserver of any one of the preceding claims, wherein the mapper is further configured to search for a conditional conjunction in a text block and upon finding a conditional conjunction in the text block, splitting the text block into an earlier first text sub-block and a later second text sub-block.

7. The online webserver of claim 6, wherein the mapper is further configured to map the first sub-block into a first logic expression and map the second text sub-block into a second logic expression that implies the first logic expression.

8. The online webserver of claim 6 or 7, wherein splitting the text block is based on a parse tree generated by the language processor and having a first sub-tree and a second sub-tree connected at a node representing the conditional conjunction and the first text sub-block represents the first sub-tree and the second text sub-block represents the second sub-tree.

9. The online webserver of any one of the preceding claims, wherein the legal text data input is configured to split the legal text into paragraphs.

10. The online webserver of any one of the preceding claims, wherein the multiple user input elements comprise binary input elements.

11. The online webserver of any one of the preceding claims, wherein the user interface module is configured to change the user interface by reducing the multiple user input elements.

12. The online webserver of claim 11, wherein reducing the multiple user input elements comprises eliminating, by the logic engine, atomic literals in the logic expression based on the assignment of the atomic literal associated with the one of the multiple user input elements and de-activating, by the user interface module, user input elements associated with the eliminated atomic literals.

13. The online webserver of any one of the preceding claims, wherein

the language processor is configured to determine a similarity between atomic literals and a reference atomic literal

the user interface module is configured to

create a user interface comprising one or more user input elements, each of the one or more user input elements being associated with one of the atomic literals based on the similarity between the one of the atomic literals and the reference atomic literal,

monitor user interaction with the user interface to detect user interaction with one of the one or more user input elements,

upon detection of user interaction with the one of the one or more user input elements, merging the atomic literal associated with the one of the multiple user input elements with the reference atomic literal in the logic expression.

14. The online webserver of claim 13, wherein the language processor is configured to determine the similarity based on common words between a first part of the legal text associated with each of the atomic literals and a second part of the legal text associated with the reference atomic literal.

15. The online webserver of any one of the preceding claims, wherein the logic expression is a deontic defeasible logic expression.

16. The online webserver of any one of the preceding claims, wherein the language processor comprises a Stanford CoreNLP parser.

17. The online webserver of any one of the preceding claims, wherein the logic engine comprises a SPINdle logic reasoner engine.

18. The online webserver of any one of the preceding claims, wherein atomic literals comprise numerical literals.

19. The online webserver of any one of the preceding claims, wherein atomic literals comprise Boolean literals.

20. A method for creating logic rules from legal text, the method comprising:

receiving the legal text and splitting the legal text into multiple text blocks;

performing natural language processing for each of the multiple text blocks to extract atomic literals that correspond to terms in the legal text and that are shared across logic expressions for different text blocks;

mapping each of the text blocks into a logic expression of the atomic literals

evaluating the logic expression based on one or more assignments of the atomic literals;

creating a user interface comprising multiple user input elements, each of the multiple user input elements being associated with one of the atomic literals;

monitoring user interaction with the user interface to detect user interaction with one of the multiple user input elements;

upon detection of user interaction with one of the multiple user input elements, creating an assignment of the atomic literal associated with the one of the multiple user

input elements and calling the logic engine to evaluate the logic expression based on the created assignment;

changing the user interface based on evaluating the logic expression to thereby provide legal advice in the form of remaining atomic literals; and

communicating with a client device by receiving user input from the client device and providing the user interface to be displayed on the client device.

21. The method of claim 20, wherein performing natural language processing comprises determining a parse tree for each of the text blocks and mapping each of the text blocks into the logic expression is based on the parse tree.

22. The method of claim 20 or 21, wherein mapping each of the text blocks into the logic expression is based on heuristics.

23. A computer system for creating logic rules from legal text, the computer system comprising:

a data input port to receive legal text;

a processor to

split the legal text into multiple text blocks;

perform natural language processing for each of the multiple text blocks, to extract atomic literals that correspond to terms in the legal text and that are shared across logic expressions for different text blocks; and

map each of the text blocks into a logic expression of the atomic literals

evaluate the logic expression based on one or more assignments of the atomic literals;

create a user interface comprising multiple user input elements, each of the multiple user input elements being associated with one of the atomic literals;

monitor user interaction with the user interface to detect user interaction with one of the multiple user input elements;

upon detection of user interaction with one of the multiple user input elements, creating an assignment of the atomic literal associated with the one of the

multiple user input elements and calling the logic engine to evaluate the logic expression based on the created assignment;

change the user interface based on evaluating the logic expression to thereby provide legal advice in the form of remaining atomic literals; and

communicate with a client device by receiving user input from the client device and providing the user interface to be displayed on the client device.