

ErgoText and Federal Regulation W

This example illustrates the use of ErgoText, i.e., knowledge base definition via a corpus of English phrases. As a domain, we are using the fairly complex Federal banking regulation known as Regulation W. The knowledge and rule base in the example is an updated version of the one used for the Enterprise Data Management Council (EDMC) sponsored FIBO rules pilot described in more detail [here](#). The regulation concerns transactions between banks and their “affiliates” and is designed to limit risk. It defines who is an affiliate, which transactions are “covered” by the regulation (i.e., to which transactions does it apply) and whether a particular transaction is permitted by the regulation. For the purpose of this example and to make it manageable, we chose to represent only part of Regulation W. The information includes (among other things) a fictional scenario in which Pacific Bank is considering a loan of \$23 million to the Maui Sunset hotel group in order to open a new location on the island. Some questions that can be asked here are: Is this transaction allowed under Regulation W? As part of that, one must ascertain if Maui Sunset could be considered an “affiliate” of Pacific Bank under Regulation W.

Files: The knowledge base is thoroughly commented and should be easy to understand. As in every ErgoText knowledge base, it consists of two parts. The first part is the [actual knowledge base](#). It consists of facts and rules mostly written with the help of a corpus of predefined English phrase templates. The logical meaning of the sentences in the corpus is given in a separate file, which has a collection of [ErgoText template](#) definitions.

Queries: In the first of these files, around lines 85-105, one can find a list of sample queries that one might want to ask the knowledge base. The user can formulate additional queries using the predefined corpus or directly in the core Ergo syntax. Note that the first file, the knowledge base itself, should be generally understandable by a layman. However, to be able to formulate meaningful queries the user must have some familiarity with ErgoText and the core Ergo syntax.

Running the demo: Download both the [regwdemo.ergo](#) and [regwdemo.ergotxt](#) files, and place them in the same directory. Then:

1. Start Ergo Suite.
2. Load the file *regwdemo.ergo*.
3. Go to the *Tools* menu in Ergo Listener and select *Ergo Query*.
4. In the Ergo Query window, paste one of the suggested queries mentioned above. Let's say we choose the first one: `\(RegW prohibits the proposed transaction between ?Bank and ?Company for amount $?Amount million\)`. Don't forget to terminate the query with a dot! Note that this query has three variables: ?Bank, ?Company, and ?Amount, and

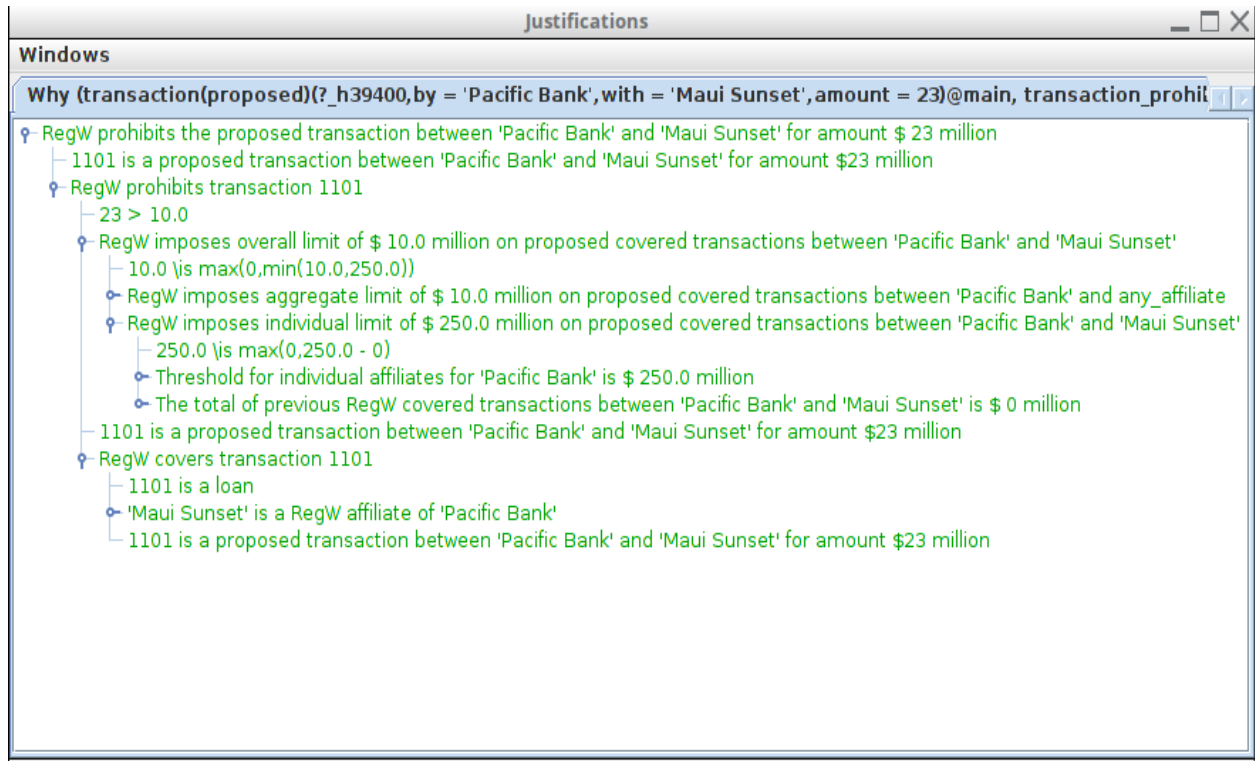
therefore the query result is expected to be a set of triples. In this example, the result will contain exactly one triple.

5. Type *<Enter>* or click the *Execute* button. You will see an answer under the query, as in the picture below.

The screenshot shows a window titled "Ergo Query" with a menu bar containing "Edit", "View", "History", and "Windows". Below the menu bar are three buttons: "Execute", "Pause", and "Abort". The query text is: `\(RegW prohibits the proposed transaction between ?Bank and ?Company for amount $?Amount million\)`. Below the query is a table with three columns: "?Bank", "?Company", and "?Amount". The table contains one row of data: "'Pacific Bank'", "'Maui Sunset'", and "23".

?Bank	?Company	?Amount
'Pacific Bank'	'Maui Sunset'	23

So, there is a proposed transaction between Pacific Bank and Maui Sunset for \$23 million, and it is prohibited under Regulation W. Naturally, the companies involved might want to know why and ask for an explanation. Right-clicking on the answer and selecting "Why?" (or double-clicking on the answer) will provide explanations shown in the picture below.



The picture shows the partially expanded explanation tree. It says that the answer is true because 1101 is a proposed transaction between our two companies and RegW prohibits it. The first statement has no further justification because it is a given fact, but explaining the second statement (prohibition of 1101) requires quite a number of steps. The first level explanation is that

- A. the amount was 23 and that 23 is more than 10
- B. RegW imposes a limit of \$10 million on the covered transactions between our two companies, and
- C. RegW covers Transaction 1101.

Sentences B and C include further levels of explanations, and the next level is also shown in the above figure. However, there are many more levels of reasoning available, which a user might find overwhelming or unnecessary. For this reason, these levels are hidden behind the horizontal handles (on lines 7, 10, 11, 15) until a user is interested in viewing them. Clicking on those handles will expand the explanation tree further. Clicking on the expanded vertical handles will contract the corresponding subtree into one line.

Exception Case Example

The regwdemo.ergo file contains an example of how to create an exception case where one or more rules may have higher priority than another rule and thereby block or override that rule. In

this case, a transaction that would be considered to be covered by Regulation W is exempted because of another condition, that there is a ready market for the the purchase of the asset.

The general case of a covered transaction is shown on lines 305-335 (rule 104d). The exception rules to the general case are shown on lines 348-367 (rules 104e, 104f, and 104g). The rule *tag* is used in the `\overrides` builtin predicate, which states that an exception exists. In this case, the following two tags are used:

```
@{'general case of covered transaction'} // tag marking general-case rules
@{'exemption case of covered transaction'} // tag of exception-case rules
```

That `\overrides` statement is given as a fact on lines 345-346:

```
\overrides('exemption case for covered transaction',
           'general case of covered transaction').
```

It states that if one of the exception rules is true, it will override the general case and exempt the transaction from being covered.

More details and examples on this type of reasoning in Ergo can be found in the [section of the Examples Bank on defeasible reasoning](#).