SCALABLE DEFEASIBLE REASONING V2

1 Knowledge Representation

Propositional logic is a framework that represents information about the world as logical statements, known as *formulas,* in a knowledge base.

E.g., **"Tweety is a penguin,"** this can be represented as:

$t \longrightarrow p$

This information can be reasoned about to form conclusions. With the statements:

"penguins are birds" (i.e., $p \longrightarrow b$)

"birds fly" (i.e., $b \longrightarrow f$)

We can conclude that **"Tweety can fly"** (i.e., $t \longrightarrow f$)

If the statement **"penguins don't fly"** was added to the knowledge base, we conclude that penguins cannot exist. Although the reasoning is correct, the outcome is undesirable.

Defeasible reasoning retracts previously held beliefs when presented with additional information.

E.g., changing **"birds fly"** ($b \longrightarrow f$) to **"birds typically fly"** (i.e., $b \mid\!\sim f$).

Using defeasible reasoning, we can conclude **Tweety can't fly**.

2a Rational Closure

1. Determines which defeasible conclusions can be drawn.
2. An algorithm for computing Rational Closure exists. Ranks defeasible statements automatically according to the generality of the statements (more general = higher ranked).
3. It performs a series of entailment checks and removes ranks until a conclusion can be drawn.

2b Key Aims of Rational Closure

- Develop a program to compute Rational Closure
- Test new optimisations to increase the scalability of Rational Closure which performs better than previous iterations.
- Acquire empirical results that show the effective performance of the optimization techniques.

Optimisations

- **Ternary Search:** Decreases the search range and number of entailment checks, with a time complexity of O(log n base 3).
- **Concurrency Approach:** Uses the *Fork/Join framework*. Users can enter multiple queries in one instance. The negations of the antecedents and the rank at which the antecedents becomes consistent are stored in a *HashMap*.

3a Lexicographic Closure

Lexicographic Closure is similar to Rational Closure, but it can be viewed as a refinement of Rational Closure. Each statement within a rank is checked before removing the entire rank and drawing any conclusions.

3b Key Aims of Lexicographic Closure

- Create a program to compute Lexicographic Closure
- Test new optimisations to obtain a version of the algorithm which performs better than previous iterations.

Optimisations

- **Power Set**: Decreases entailment checks. Improved by taking up less storage in memory.
- **Fibonacci Search**: Speeds up finding the rank to remove. New iteration on Ternary Search from previous year.
- **Concurrent Approach**: Computes multiple queries simultaneously.

4 Conclusions

Rational Closure: the Concurrency Approach is more scalable as it is able to query multiple sets at a time. An increase in rank size leads to a consistent increase in performance above all other implementations. This approach speeds up computation time by recovering the rank, instead of performing the calculation several times for the same antecedent.

Lexicographic Closure: the Power Set approach allowed for more statements to be handled within the ranks. The Fibonacci Search proved to be slower than the Ternary Search at larger scale knowledge bases. The concurrent Approach proved to be much faster than performing each query sequentially.

5 Future Work

Test a range of sequential thresholds on computers with more than two cores.

Improve upon knowledge base generation to produce more complex statements.

Improve the sub-algorithm for ranking statements with focus on ranking the knowledge base at a faster rate.

Extend the system to other languages such as C or C++, which are platform independent and much faster than Java.

Consider developing or utilizing other tools (i.e., similar to TweetyProject libraries) to increase performance.

Evashna Pillay
PLLEVA005@myuct.ac.za

Dhiresh Thakor Vallabh
THKDHI001@myuct.ac.za

Supervised by Professor Tommie Meyer
tmeyer@cair.org.za

Department of Computer Science, University of Cape Town