Knowledge Compilation For Defeasible Reasoning

Luke Slater

TL;DR

Our work tackles *defeasible reasoning*—reasoning with "usually true" statements like *if it's a bird, it probably flies*. These conditionals are organized by plausibility: more general or typical rules get higher ranks, and lower-ranked, conflicting ones are progressively ignored until the knowledge base is consistent with the query. This ranking-based approach captures how humans reason about exceptions without collapsing into contradictions. The catch? It's computationally nasty. So we use knowledge compilation—turning the logic into compact decision diagrams—to make this process fast enough to be practical.

Symbolic Artificial Intelligence

Symbolic AI represents knowledge through explicit rules and symbols—basically logic you can read, not just numbers in a matrix. It's the branch of AI that tries to *reason* rather than just *predict*, aiming to make machines that understand and manipulate structured knowledge the way humans handle concepts and language.

$$(A \wedge B) \vee (C \wedge D)$$

Conditional Logic

Conditional logic deals with statements like "if A, then normally B," capturing general rules that can tolerate exceptions.

BIRD ~> FLIES

Conditional Rankings

Conditional rankings assign plausibility levels to rules so that when conflicts arise, less reliable or more specific conditionals are retracted first.

∞	PENGUIN => BIRD
1	PENGUIN ~> NOT FLIES
0	BIRD ~> FLIES BIRD ~> WINGS

Conditional Entailment

Conditional entailment is about drawing conclusions that hold *for now*—they follow from the current set of plausible rules but can be withdrawn if new, more specific information appears. For instance, from the above ranking we get the following (non) inference:

PENGUIN $\not\sim>$ WINGS

Knowledge Compilation

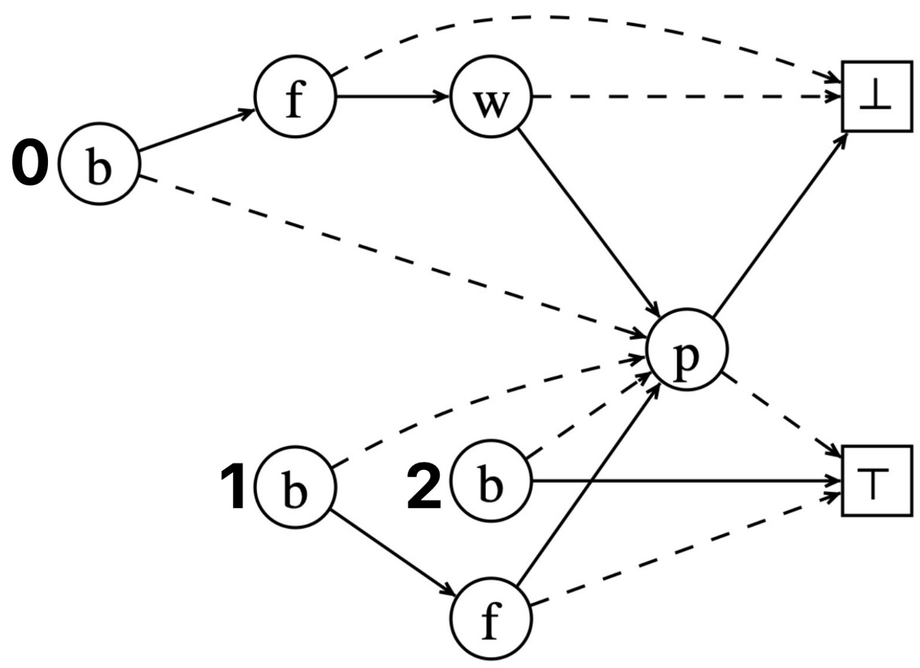
Knowledge compilation transforms a symbolic theory into an efficient circuit-like structure that precomputes much of the reasoning work. Once compiled, complex inferences can be answered quickly by simple evaluations, trading one expensive upfront computation for many fast queries later.

Binary Decision Diagrams

Binary Decision Diagrams (BDDs) are compact graph-based representations of Boolean formulas. They capture all possible truth assignments in a structured way, making logical operations and equivalence checks fast, deterministic, and memory-efficient once the diagram is built.

Encoding The Ranking

Each layer of the plausibility ranking is translated into its own decision diagram, capturing the conditionals valid at that level. By precompiling these ranked theories, the system can efficiently determine which rules survive or are discarded during reasoning



Results

By compiling the ranked theories into decision diagrams, inference ran up to 100× faster than a state-of-the-art SAT solver. The compiled structure eliminates repeated search, turning expensive reasoning into quick lookups.

Student: Luke Slater Email: sltluk001@myuct.ac.za
Primary Supervisor: Thomas Meyer
Secondary Supervisor: Jesse Heyninck

University of Cape Town • iYunivesithi yaseKapa • Universiteit van Kaapstad

SCHOOL OF IT