

Explanation Using Defeasible Rulesets: Verifying ECCRS Predictions with Rational Closure

Daniel Schlesinger

SCHDAN066@myuct.ac.za

University of Cape Town

Cape Town, Western Cape, South Africa

Abstract

Symbolic rule-based classifiers make predictions traceable, but traceability alone does not guarantee an explanation a person would recognise as their own reasoning, and most rule formats do not preserve the default-and-exception structure that reasoning relies on. Exception Closed Conjunctive Rule Sets (ECCRS) preserve that structure directly in the rule bodies and resolve conflicts through a Most Specific Wins (MSW) operator, where the most specific applicable rules decide the label and the classifier abstains when none apply. This default-and-exception behaviour has a formal counterpart in logic, where conclusions that more specific information can later override are called non-monotonic, in contrast to classical deduction, where a conclusion, once reached, is never withdrawn. An alignment theorem states that MSW predictions agree exactly with Rational Closure (RC) entailment (a standard, minimal way of drawing such non-monotonic conclusions by ranking the knowledge base’s rules by how exceptional they are) over the corresponding conditional knowledge base, a set of rules of the form “if this, then typically that”, provided the ruleset satisfies three structural conditions: strict global exception closure, sanity, and no total override. This paper presents the Rational Closure component of a verification workbench built around that theorem. We validate the three conditions on three exported ECCRS rulesets, translate each into a conditional knowledge base, and implement Rational Closure, BaseRank followed by entailment checking, verified against an independent oracle across hundreds of random knowledge bases and the worked examples given in the literature. For every complete feature pattern a ruleset covers (every possible true/false setting of the features the rules mention), the workbench compares RC and MSW predictions and produces a two-sided trace linking the ECCRS narrative to the RC rank comparison. On all three rulesets exported for this project, Breast Cancer Wisconsin, Bank Marketing, and Mushroom, RC and MSW agree exactly on every complete feature pattern each ruleset covers, with no abstentions on either side; on a small scholarship ruleset that violates one of the three conditions, RC and MSW diverge on exactly the one pattern the violation predicts they should, and nowhere else. We also implement and measure an optimised RC procedure: grouping patterns by which rules they make applicable, instead of calling the satisfiability oracle once per pattern, cuts those calls by up to 964-fold on Bank Marketing, the largest of the three rulesets (904-fold against a naive ascending-scan baseline rather than the binary search shipped by default), verified by differential testing against the ungrouped computation. The result is a Rational Closure reasoner that forms the shared foundation for the Lexicographic Closure and System W components of the wider project.

1 Introduction

Machine learning models are increasingly deployed in consequential settings such as medical diagnosis, credit assessment [9], and criminal risk profiling, where a prediction on its own is not enough: those affected by a decision, and those responsible for it, need to understand why it was reached [7]. Post-hoc explanation methods approximate a black-box model with a simpler, interpretable surrogate after the fact; as Rudin [24] argues, a surrogate is not the model it describes, and in high-stakes settings that gap is hard to justify. This paper treats interpretability as a requirement from the outset rather than something to recover afterward.

Symbolic rule-based classifiers meet that requirement by representing a decision as explicit if-then rules over meaningful features, making every prediction traceable without extra explanation machinery. But traceability is weaker than explanation: a rule trace shows which conditions fired, not that the trace reflects how a person would justify the decision. Existing approaches expose this gap differently. Decision lists resolve conflicts by rule order, so reordering the same rules can change a prediction even though nothing about the rules changed [20]; unordered rule sets avoid that instability but need an external tie-break (scoring, voting, hand-tuned priorities) that sits outside the explanation. Neither captures the pattern underlying most human justification: a default held until a more specific exception overrides it.

Exception Closed Conjunctive Rule Sets (ECCRS) close this gap by encoding the default-and-exception relationship directly in the rule bodies, rather than in rule order or an external tie-break [18]; the alignment theorem this paper builds on is stated separately, in a later, unpublished note by the same lead author [17]. Predictions follow a Most Specific Wins (MSW) principle: the most specific applicable rules decide the label, and the classifier abstains if none apply, keeping every explanation short, stable under reordering, and built from the same structure a person would use to justify the decision.

This default-and-exception pattern is not just engineering convenience; it is a well-studied idea in logic. Classical deduction is monotonic: once a conclusion follows from what is known, no further information retracts it. Reasoning with defaults and exceptions cannot work that way. Learning that a particular bird is a penguin overturns the earlier default that it flies, without disturbing anything else believed about it, so it is instead *non-monotonic*, or *defeasible*: a conclusion holds only provisionally, and more specific information can withdraw it later. MSW alone does not connect ECCRS’s default-and-exception structure to a formal, checkable account of this reasoning. An alignment theorem closes that gap. Once an ECCRS ruleset is translated into a *conditional knowledge base* (a collection of “if this, then typically that” statements in the

style of Kraus, Lehmann and Magidor (KLM) [14]), MSW predictions agree exactly with what that knowledge base entails under *Rational Closure* (RC), a minimal way of drawing non-monotonic conclusions by ranking statements by how exceptional they are [16]. This holds provided the ruleset satisfies three structural conditions: strict global exception closure, sanity, and no total override [17].

Rational Closure is where this paper’s own contribution sits: it is the entailment method the alignment theorem is stated against, so any claim that ECCRS explanations are formally grounded rests on RC being implemented correctly, and it is the shared foundation the wider project builds on. Lexicographic Closure and System W, implemented and evaluated separately by Mochoeneng [19] and Baker [2] in their own codebases, both extend this same BaseRank ranking rather than replacing it. Neither shares code with it, and their results belong to their own papers, not this one; but an RC implementation that is wrong, slow, or untested would carry into both before either could produce anything of their own.

This paper presents the Rational Closure component of a verification workbench built around the alignment theorem. Our contribution is fourfold: validating the three structural conditions on three exported ECCRS rulesets; translating each into a KLM conditional knowledge base; implementing Rational Closure (BaseRank followed by entailment checking), verified against an independent from-first-principles oracle and extended into an optimised procedure that removes almost all of the satisfiability calls the evaluation requires; and using the result to check RC and MSW predictions against each other for consistency with the alignment theorem, instance by instance, on rulesets exported for Breast Cancer Wisconsin, Bank Marketing, and Mushroom. Agreement on these rulesets doubles as an end-to-end verification of the implementation, since a wrong ranking or broken entailment check would surface as a disagreement with an independently computed MSW baseline. All three cover their entire feature-pattern space, exhaustively checking the two cases where a ruleset actually decides a label; the third case, where no rule applies, is exercised separately on a small scholarship ruleset used as a worked counter-example. The validator, the translation, and the two-sided trace design are shared infrastructure the other two components also build on; the Rational Closure implementation, its optimisation (Section 3.5), and this evaluation are individually assessed as this paper’s own.

The rest of the paper is organised as follows. Section 2 places ECCRS and Rational Closure within the wider literature. Section 3 describes the workbench: assumption validation, translation, the naive and optimised Rational Closure procedures, and the explanation traces. Section 4 reports agreement and runtime results. Section 5 concludes.

2 Related Work

2.1 Interpretability and Symbolic Learning

Interpretability and explainability are related but distinct properties: a model can be interpretable, in that its internal workings can be inspected, without being explainable in the stronger sense of justifying its own reasoning [10]. The COMPAS recidivism tool illustrates the stakes of collapsing this distinction: its predictions were later shown, and are now well known, to be racially biased, a

flaw that a purely post-hoc account of its behaviour could not have surfaced. Post-hoc methods such as LIME address interpretability by fitting a simpler, local surrogate model around a black-box prediction [22]. This is the same surrogate-is-not-the-model gap Section 1 already raises [24].

Models that are transparent from the start avoid this gap. CORELS learns certifiably optimal rule lists and matches black-box accuracy on datasets such as COMPAS, showing that transparency need not cost predictive performance [1]. Short Boolean formula classifiers reach a similar conclusion by a different route, using cross-validation to bound formula length and finding that the same limit which prevents overfitting also keeps formulas short enough to read [11]. Neither captures the default-and-exception structure most human justification relies on: a rule list’s order-based conflict resolution (Section 1) encodes an exception as a separate, higher-priority rule with no explicit link back to the default it overrides, and a Boolean formula buries the same relationship inside a nested logical structure. Both are traceable; neither reads as the reason a person would actually give.

2.2 ECCRS as a Symbolic Classifier

Exception Closed Conjunctive Rule Sets (ECCRS) close the default-and-exception gap directly: each rule pairs a body of signed literals with a class label, and whenever two rules with opposite labels can both apply to the same instance, one body must be a strict subset of the other [17]. This exception closure property is what lets predictions follow a Most Specific Wins (MSW) principle, where the most specific applicable rules decide the label and the classifier abstains when none apply, without needing any tie-breaking rule outside the ruleset itself. ECCRS distinguishes a relaxed closure, constraining only rules that conflict on the training data, from the stronger strict closure that extends the same requirement across the entire feature space, the version the alignment theorem we build on requires [17]. ECCRS itself is due to [18]; the alignment theorem is stated separately, in an unpublished ECCRS specification note by the same lead author, written for this honours project [17]. We take its proof as given and test its predictions empirically rather than reproducing it here. The structural conditions and the worked example that motivates them are given in full in Section 3.

2.3 The KLM Framework

Classical logic (the standard deductive logic of mathematics and most of computer science, where a conclusion is derived from a set of premises by fixed rules of inference) is monotonic: once a conclusion follows from a set of premises, no additional information can retract it. This is unworkable for reasoning about exceptions, since a knowledge base stating that birds fly and that penguins are birds that do not fly becomes simply inconsistent rather than exception-tolerant. Early proposals such as Default Logic introduced tentative conclusions that could later be withdrawn [21], but without a general framework for comparing different notions of non-monotonic, or *defeasible*, inference. Kraus, Lehmann, and Magidor (KLM) supplied that framework by characterising families of non-monotonic consequence relations through the properties they satisfy, rather than committing to a single reasoning system

[14]. Its semantic core is the preferential model, which orders possible worlds (complete assignments of true or false to every atom) by plausibility: a defeasible conditional $\alpha \sim \beta$ (read “if α , then typically β ”) holds when every most plausible world satisfying α also satisfies β . Lehmann and Magidor refine this into ranked preferential models, where each world is assigned a rank reflecting how typical it is, with lower ranks marking more normal situations [16]. This ranked structure is what Rational Closure, the entailment method this paper implements, is built on.

2.4 Rational Closure

Rational Closure ranks the conditionals in a knowledge base by how exceptional their antecedents (the “if” side of the conditional) are, then answers a query by provisionally setting aside the most general information whenever it conflicts with something more specific [16]. A conditional’s base rank is the lowest rank at which its antecedent is no longer exceptional, where exceptional means the antecedent is inconsistent with everything not yet set aside. The BaseRank algorithm computes these ranks iteratively: starting from the whole knowledge base, it repeatedly removes every conditional whose antecedent is currently exceptional, assigns the removed layer the next rank, and stops when nothing further can be removed, at which point anything left over is assigned an infinite rank and treated as classical fact. Query answering follows directly from these ranks: $K \models_{RC} \alpha \sim \beta$ holds exactly when $br_K(\alpha) < br_K(\alpha \wedge \neg\beta)$, or when α itself has infinite rank [16]. Because both base ranking and query answering reduce to a bounded number of calls to a classical entailment check, Rational Closure is no harder, complexity-wise, than the entailment problem it is built on [12].

Rational Closure’s main limitation, well known in the literature, is that it does not let an exceptional subclass inherit generic properties from its superclass: knowing that penguins do not fly is enough to rank penguins as exceptional birds, and once a subclass is exceptional at some rank, none of its superclass’s properties at that rank or below are available to it, including properties, such as having wings, that have nothing to do with why the subclass was exceptional in the first place. This over-blocking is what motivates the stronger entailment methods described next.

2.5 Lexicographic Closure and System W

Lexicographic Closure and System W both extend Rational Closure’s ranked semantics, rather than replacing it, to recover conclusions its over-blocking discards. Lexicographic Closure re-orders possible worlds by how seriously they violate the knowledge base, preferring to keep specific, low-rank information over general, high-rank information when something has to give [15]. System W goes further, ordering worlds by which conditionals they actually falsify under a tolerance partition derived from the same ranked structure, which lets it recover conclusions that both Rational Closure and Lexicographic Closure miss [13]. Both are covered in full in the other two papers from this project [2, 19]; what matters here is that getting Rational Closure’s ranking right, and running it on rulesets larger than a worked textbook example, was a prerequisite for the rest of the project, not a side concern of this paper alone.

2.6 Translation and Explanation Traces

Darwiche and Hirth give the formal basis for explaining a symbolic classifier’s decisions, defining a complete reason as the necessary and sufficient condition for a prediction and showing that shorter, sufficient reasons can be derived from it [6]. Rienstra shows that these sufficient reasons can be needlessly long when they include features that are technically relevant but redundant given relationships already implicit in the data, closing that gap with non-monotonic background theories rather than purely propositional ones [23]. This is the direction this project’s translation into a KLM knowledge base takes. Chama et al. extend this to defeasible knowledge bases directly, giving an algorithm for identifying the minimal subset of a ranked knowledge base responsible for a given entailment [5], close to what the two-sided explanation trace in this paper needs to report: which ranked conditionals were decisive and which were set aside as too exceptional, building on the same ranking-based justification approach in Chama’s earlier thesis [4]. This is the basis for producing an ECCRS-side and a Rational-Closure-side trace for the same query, and for treating their agreement as part of the verification process.

What is missing from this line of work is a Rational Closure implementation that has actually been built and tested against ECCRS rulesets, not just textbook examples; that runs efficiently on rulesets learned from real data instead of the five-rule scholarship illustration the alignment theorem was first stated against; and that checks the theorem empirically on such rulesets in the first place. This paper addresses these gaps for the Rational Closure case, which the Lexicographic Closure and System W components of the project then build on.

3 Design and Implementation

3.1 Overview

The workbench is a four-stage pipeline (Figure 1). A parser reads an ECCRS ruleset $\mathcal{R}$ from a plain-text rule file and, when present, a set of labelled feature-pattern instances. An assumption validator checks $\mathcal{R}$ against the three structural conditions the alignment theorem requires: strict global exception closure, sanity, and no total override (Section 3.2). A translator maps $\mathcal{R}$ onto a KLM conditional knowledge base $K(\mathcal{R})$ (Section 3.3), over which Rational Closure is run in both its naive form (Section 3.4) and its optimised form (Section 3.5). The Most Specific Wins operator is evaluated directly on the original ruleset, independently of the translation, so it can serve as the ground truth the Rational Closure side is checked against rather than something derived from the pipeline it verifies. For any query pattern the workbench produces a two-sided explanation trace and an agreement report comparing Rational Closure’s predictions against Most Specific Wins, across the full pattern space or a supplied dataset (Section 3.6).

The pipeline is linear and has no learning component: rulesets arrive already learned, and every stage is deterministic, so the same ruleset and query always produce the same trace, which lets each stage’s correctness be checked in isolation against a known-answer test from the literature, not just against the pipeline’s own output.

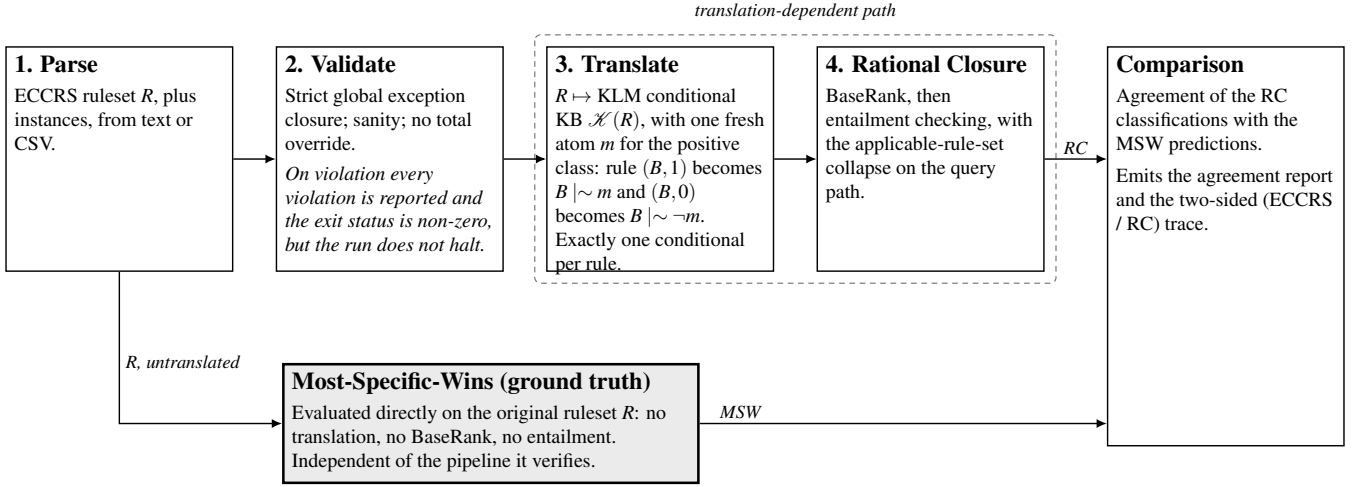


Figure 1: Workbench pipeline. A ruleset R is parsed, checked against the three structural conditions, translated into the KLM conditional knowledge base $\mathcal{K}(R)$ (one conditional per rule, over a single fresh class atom m), and queried under Rational Closure. Most-Specific-Wins is evaluated on R itself: it branches off after parsing and bypasses the translation, so the ground truth is not derived from the pipeline it is used to verify. A validation failure is reported in full and sets a non-zero exit status, but does not halt the run. Both paths meet at the comparison stage, which emits the agreement report and the two-sided trace.

3.2 Assumption Validation

Before translation, the ruleset is checked against the three conditions the alignment theorem requires [17]. Sanity rules out two rules with identical bodies but different labels, since such a pair could never be resolved by specificity. Strict global exception closure requires that whenever two rules with opposite labels could both apply to the same instance, one body is a strict subset of the other, checked across the entire Boolean feature space, not only the training instances. Because this check is pairwise, it is restricted to opposite-label pairs whose bodies do not already disagree on some feature; such pairs can never be jointly applicable, so requiring them to be comparable would reject well-formed rulesets. No total override requires every rule to survive as the sole decider on at least one prediction; a rule is overridden only by a strictly more specific rule with the opposite label [18]; a more specific rule with the same label reinforces rather than replaces it. A rule can fail to ever be the most specific applicable rule without being overridden in this stronger sense, most likely because it is refined at every branch by more specific rules of both labels; the two properties are logically distinct, and a ruleset can satisfy one while failing the other.

The original design halted on the first violation found; the workbench instead reports every violation, states that the alignment guarantee no longer applies, exits non-zero, but keeps running, because halting would have hidden the cases most worth reporting. The published scholarship ruleset used to illustrate MSW itself violates strict global exception closure, and the resulting RC/MSW comparison on it, including the single pattern where the two disagree, is a useful worked example precisely because the violation stays visible.

3.3 Translation

Translation introduces a single fresh propositional atom m , standing for the positive class, and maps each ECCRS rule (B, ℓ) onto exactly one defeasible conditional in the KLM knowledge base $K(\mathcal{R})$: a rule with label 1 becomes $B \sim m$, and a rule with label 0 becomes $B \sim \neg m$. The resulting knowledge base has exactly one conditional per rule and preserves the specificity ordering exactly, since $B \sim m$'s antecedent is unchanged from the original rule body. The fresh atom m must be chosen above every atom index in the ruleset or the underlying dataset, not the ruleset alone: a class atom that coincides with a real, unused dataset feature is a latent bug that passes every test built from rule-mentioned atoms and only corrupts a query built directly from dataset values.

3.4 Naive Rational Closure

The naive procedure is BaseRank followed by a direct application of the entailment condition, implemented exactly as specified in the literature, and serves as the correctness reference for everything built on top of it.

BaseRank assigns every conditional in $K(\mathcal{R})$ a rank reflecting how exceptional its antecedent is: a body α is exceptional for a set of materialised clauses Σ (the classical implications obtained by replacing each conditional's $\sim$ with $\rightarrow$) when $\Sigma \cup \{\alpha\}$ is unsatisfiable, i.e. when Σ classically entails $\neg\alpha$. The ranking loop starts from the full set of conditionals and, at each iteration, removes every conditional exceptional with respect to the conditionals still remaining, assigning the removed layer the current rank before incrementing it; conditionals never removed, because the process reaches a fixed point while they are still present, receive an infinite rank and are treated as classical fact. Exceptionality must be judged against the clauses still remaining, not the full original knowledge base. The two only diverge once a rank has already been assigned,

so checking against the original set silently over-ranks anything whose antecedent conflicts with an already-resolved conditional. An earlier pass at this exact computation made precisely that error, caught only by checking it line by line against the published algorithm [12].

Query answering follows the ranks directly: the base rank of a formula, $\text{br}_K(\alpha)$, is the lowest rank at which α is not exceptional, found in the naive procedure by an ascending linear scan. Entailment holds, $K \models_{\text{RC}} \alpha \vdash \beta$, exactly when $\text{br}_K(\alpha) < \text{br}_K(\alpha \wedge \neg\beta)$, or when α itself has infinite rank [16], meaning α is exceptional for the infinite-rank conditionals too, leaving no world that satisfies it. This is distinct from an α merely exceptional at every finite rank, which is answered against those conditionals in the ordinary way; conflating the two is an easy error, since a single infinite value stands for both the classical level and the absence of one. For a complete feature pattern F , the workbench asks this twice, for $F \vdash m$ and $F \vdash \neg m$; under the validated assumptions exactly one holds, or neither, matching Rational Closure’s abstention to Most Specific Wins’.

Every step is checked against a published worked example before anything of our own: the base-rank computation against a knowledge base and rank table given in the literature, and the entailment condition against a specific query result reported alongside it [12]. Beyond these fixed examples, the whole ranking-and-entailment stack is differentially tested against an independent, from-first-principles oracle that recomputes exceptionality by brute-force truth-table enumeration and rank by an ascending scan with none of the naive procedure’s own structure reused, across 312 randomly generated knowledge bases and 3120 queries. Seven of the knowledge bases place a conditional at infinite rank, and 229 of the queries have an antecedent exceptional at every finite rank and so must be decided against the classical conditionals alone. Passing both justifies treating this procedure as the reference the rest of the workbench is checked against.

3.5 Optimisation

BaseRank’s ranking loop and the naive ascending scan behind br_K both call the underlying satisfiability check a number of times that grows with the ruleset, and the full agreement analysis calls br_K three times for every one of the 2^a complete feature patterns over a rule atoms. Kaliski’s complexity result points to where the effort should go: Rational Closure only ever calls a classical entailment oracle a polynomial number of times, so it is no harder, computationally, than that oracle itself [12]. The same shape of result (a defeasible reasoning problem decided by a bounded number of calls to a classical, monotonic reasoner treated as a replaceable black box) has been shown for Rational Closure over a description logic rather than a propositional one [3]; the specific bound does not carry over, but it is independent evidence that counting oracle calls is the right way to measure a procedure built this way. Optimising this system has three levers, ranked by expected payoff: fewer calls to the satisfiability oracle, cheaper calls by exploiting the structure of the clauses generated, and a faster oracle itself. We work strictly in that order: removing most of the calls is worth more, and easier to justify, than a constant-factor improvement inside the solver.

Two structural changes are already shipped as part of the reference implementation. First, br_K ’s cumulative per-rank clause sets are built once, in a single backward pass over the ranked knowledge base, instead of being re-materialised on every query. Second, the ascending linear scan used to find $\text{br}_K(\alpha)$ is replaced by a binary search. Non-exceptionality is monotone in the rank: dropping constraints from a clause set can never turn a satisfiable set unsatisfiable, so as the rank rises the antecedent moves through an exceptional prefix and then a non-exceptional suffix with a single crossing point, and binary search finds it in $O(\log k)$ satisfiability calls instead of $O(k)$ for k distinct ranks. This replacement’s soundness is checked against a literal transcription of the published ascending-scan procedure across the same random-knowledge-base corpus used to validate the naive procedure, so a monotonicity argument that happened to be wrong would show up as a test failure.

Whether this pays off is measured, not assumed: Section 4 reports the binary search’s satisfiability-call count against the ascending scan it replaced. The binary search remains the shipped implementation regardless of that comparison’s outcome: the asymptotic argument, $O(\log k)$ against $O(k)$, bounds the worst case for whatever k a future ruleset in the wider project turns out to have, even where it does not pay off on the rulesets measured here. Repeated queries are also memoised, which matters once the same sub-query recurs across many feature patterns.

The larger optimisation, and the one this paper is built around, starts from the same observation. Both Most Specific Wins and Rational Closure depend on a complete feature pattern F only through which rule bodies F satisfies, $\mathcal{A}(F)$: immediate for MSW, since $\mathcal{A}(F)$ is its entire input, and true for RC because every materialised clause has the form $\{-b : b \in B\} \cup \{\ell\}$, which F can only falsify by satisfying every literal in B . Two patterns with the same applicable-rule-set are therefore indistinguishable to both operators. Evaluating one representative pattern per applicable-rule-set class instead of every pattern in it is exact, not approximate, provided each class is weighted by how many patterns belong to it.

The design first considered searching directly over rule subsets S , testing whether S is exactly realisable, meaning $\bigwedge_{r \in S} B_r \wedge \bigwedge_{r \notin S} \neg B_r$ is satisfiable, to avoid visiting the 2^a pattern space at all. This does not pay off here: all three rulesets have more rules than rule atoms (12 over 6 for Breast Cancer Wisconsin, 13 over 10 for Mushroom, 21 over 14 for Bank Marketing), so the candidate subsets, 2^n , already number 64, 8, and 128 times more than the 2^a pattern space it was meant to replace, before anything is saved by grouping. It would also trade one hard problem for another: weighting a realisable subset by how many complete patterns produce it is a model-counting question, harder than the satisfiability question the rest of the workbench relies on.

What is implemented instead keeps the 2^a enumeration and changes what happens inside it. The workbench walks the pattern space once, and for each pattern F computes $\mathcal{A}(F)$ directly against the rule bodies (a lookup, not a satisfiability question, since F already fixes every atom’s truth value), groups patterns by this set, and keeps a running count and one representative per group. RC and MSW then each run once per class, and every statistic is weighted by the class’s count; because every pattern is still visited during grouping, class sizes come out as exact counts, so the model-counting difficulty the subset-search would have introduced

never arises. Grouping by $\mathcal{A}(F)$ before any clause is built, using a plain feature comparison rather than a satisfiability call, exploits the ruleset’s structure before the solver is ever invoked, rather than inside it.

This removes almost all of the satisfiability calls the agreement analysis makes; Section 4 reports the exact counts, the resulting fold-reductions, and the wall-clock time. The grouped computation is checked against the ungrouped one directly, pattern by pattern, on all five rulesets used in this paper (the three exported datasets together with the five-rule scholarship counter-example of Section 3.2 and a closed positive-control variant of it that does satisfy the three structural conditions), so it is checked to still answer every question the slower version answered.

This optimisation removes the satisfiability calls the 2^a enumeration makes, not the enumeration itself: the workbench still visits every one of the 2^a patterns once, so the workbench’s refusal to enumerate complete patterns past 20 rule atoms stands exactly where it stood before. The rule-subset search would have lifted that ceiling, which is why it was the original plan; it was not attempted because it costs more than the enumeration it was meant to avoid, for the reason given above. A ruleset with more rule atoms than rules, the reverse of every ruleset used here, is the case where that trade would turn around.

A second optimisation was designed but not built. Every query the agreement analysis makes is over a complete pattern, so every materialised clause reduces immediately to either a trivially satisfied clause or a unit literal on m (a clause with a single literal left), and deciding satisfiability collapses to checking whether the triggered units force both m and $\neg m$, a single linear scan with no branching. This is not a heuristic: every ECCRS-derived clause has at most one positive literal, which makes the whole clause set Horn regardless of the ruleset’s label mix, and definite Horn (exactly one positive literal per clause) when every label is 1. It was not attempted because its ceiling is small next to the class collapse’s: every surviving query already costs about six satisfiability calls on average, each already reduced to a handful of units by the pattern being complete, and the class collapse has already taken Bank Marketing’s evaluation two orders of magnitude below where the cost of an individual call would start to matter, so the remaining time went to the optimisation with the larger ceiling. The same differential-testing discipline would apply to this path unchanged, should it be built later.

Incremental SAT, parallelisation, and optimising the solver itself were considered and set aside: all three make individual calls cheaper, and the class collapse had already removed almost all of the calls [8]. Agreement percentages remain exact, not sampled, because every pattern, or every class of patterns, is accounted for.

We report satisfiability-call counts rather than wall-clock time wherever possible: they are deterministic and machine-independent, and connect directly to the polynomial-calls claim the whole approach rests on. Wall-clock time is secondary, meaningful mainly on Bank Marketing, where the difference is measured in seconds rather than milliseconds.

3.6 Explanation Traces

For a given feature pattern F , the workbench produces two accounts of the same prediction side by side. The ECCRS-side trace lists every applicable rule, $\mathcal{A}(F)$, identifies which are strictly overridden by a more specific applicable rule carrying the opposite label, and reports the common label of the inclusion-maximal set $\mathcal{M}(F)$ (the applicable rules whose body no other applicable rule’s body strictly contains) or an abstention when $\mathcal{M}(F)$ is empty or its members disagree. The Rational-Closure-side trace states both candidate queries, $F \vdash m$ and $F \vdash \neg m$, reports which the entailment condition confirms, and records the rank comparison that decided it: $\text{br}(F) < \text{br}(F \wedge \neg m)$ for a positive prediction, or the mirror comparison for a negative one. The Rational Closure side then maps the conditional at the decisive rank back to the ECCRS rule that produced it, so both traces converge on the same rule.

The two traces are checked against each other, not presented as independent descriptions of the same answer. $\mathcal{M}(F)$ and the decisive conditional trace back to the same rule whenever the assumptions hold, so a case where the two point to different rules is precisely what the workbench is built to catch.

3.7 Materials

The workbench is implemented in Python. Every data structure and algorithm, including the satisfiability engine of Section 3.4, is written from scratch instead of pulled from an existing library, because the optimisation in Section 3.5 needs a SAT primitive we can extend and control, not a black box, and the workbench carries no runtime dependencies at all.

The ruleset format is a plain text file of numbered rules, each a conjunction of signed feature literals implying a class label; labelled instances, when supplied, are read from the same feature space. The evaluation targets ECCRS rulesets exported from classifiers learned on three datasets, Breast Cancer Wisconsin, Bank Marketing, and Mushroom, supplied by the ECCRS side of the project in place of the Ionosphere, Breast Cancer Wisconsin, and German Credit rulesets originally proposed. Breast Cancer Wisconsin’s class-label convention does not match the DHCE model [18], which uses 0 for benign and 1 for malignant: this project’s ruleset has the opposite polarity ($a(10) = 1$ in 458 of 699 instances, matching the published benign majority, so here 1 is benign). This does not establish that a different underlying dataset was used, only that the label encoding differs; this project cannot assert its Breast Cancer Wisconsin ruleset matches that specific benchmark, and whether Bank Marketing or Mushroom match any of DHCE’s other benchmarks has not been checked either. All three rulesets were validated, translated, and evaluated for RC/MSW agreement, reported together in Section 4.

3.8 Evaluation Procedure

Two questions are asked of the workbench, measured separately rather than blended into one score.

Correctness is measured as agreement between Rational Closure and Most Specific Wins across every complete feature pattern a ruleset covers, not only the patterns in a supplied dataset. A complete feature pattern, in the theorem’s sense, is a full assignment over every feature atom a dataset has (eight for Breast Cancer Wisconsin, seventy-nine for Bank Marketing, a hundred and six for

Mushroom), far more than either operator needs: both depend on a pattern only through which rule bodies it satisfies. The workbench therefore enumerates patterns over the rule-mentioned atoms alone, 2^a of them rather than 2^N : every dataset atom no rule references is free to vary without changing either operator’s verdict, so each rule-atom pattern stands in for 2^{N-a} dataset patterns all receiving the same answer, and the agreement percentage over the smaller space is exactly what the full space would give. The same argument, at the level of whole rule sets instead of individual atoms, licenses the applicable-rule-set collapse of Section 3.5. One consequence: the pattern-enumeration limit (Section 3.5) bounds how many atoms a ruleset’s rules may reference, not how many features the dataset has: Bank Marketing stays tractable at 14 rule atoms despite 79 dataset features, because only those 14 can affect either verdict.

None of the five rulesets used in this paper produces an infinite rank (Breast Cancer Wisconsin, Bank Marketing and Mushroom each resolve into three or four distinct ranks, and both scholarship fixtures into three), and a validated ruleset arguably cannot: an infinite rank requires two opposite-label conditionals jointly unsatisfiable at every finite level, exactly what sanity and strict global exception closure rule out. This is a scope condition on what follows, not a general property of Rational Closure.

Each complete feature pattern is classified into one of three outcomes: Rational Closure and Most Specific Wins settle on the same label, they disagree, or neither operator decides because no rule applies. All three are reported as separate counts, since collapsing the no-rule-applies case into either agreement or disagreement would misrepresent what happened; agreement itself, however, follows the theorem’s own statement in counting mutual abstention as one of the ways the two operators agree, alongside the two same-label cases. Disagreement also covers the case where one operator predicts a label and the other abstains: the theorem’s three-case statement gives only two ways to agree (the same label, or both abstaining) and a one-sided abstention is neither. Efficiency is measured against the naive procedure of Section 3.4, using the metrics set out in Section 3.5: satisfiability-call counts as the primary measure, wall-clock time as secondary, and the number of patterns evaluated against the number of applicable-rule-set classes evaluated, to isolate what the class-collapse optimisation contributes.

4 Results and Discussion

This section reports what the workbench described in Section 3 produced when run on the three exported ECCRS rulesets (Breast Cancer Wisconsin, Mushroom, and Bank Marketing) and on the five-rule scholarship ruleset used throughout this paper as a worked counter-example. It asks whether Rational Closure (RC) and Most Specific Wins (MSW) agree on the translated and original versions of the same ruleset, and how much the optimisation of Section 3.5 reduces the cost of checking that agreement. Both operators answer the same question about a complete feature pattern F , a full true/false assignment to every atom some rule mentions, but reach it differently. MSW looks at the applicable rules, those whose body F satisfies, and lets the inclusion-maximal rules among them decide the label; it abstains when no rule applies, or when the maximal rules disagree. RC instead ranks the translated knowledge base’s

Table 1: Rational Closure/Most Specific Wins agreement over every complete feature pattern each ruleset’s rule atoms admit (2^a). *all* is that full pattern count and *realis.* the subset that can physically occur under one-hot mutual exclusion; the counts reported are over the full space. Agreement counts a jointly-abstained pattern as agreement, following the alignment theorem’s three-case statement ([17], §7.1), so *agree* is recoverable as the pattern total less *disag.* Datasets are Breast Cancer Wisconsin, Mushroom, and Bank Marketing.

Dataset	Rule set		Patterns		Agreement		
	rules	atoms	all	realis.	%	abst.	disag.
Cancer	12	6	64	64	100.0	0	0
Mushroom	13	10	1024	512	100.0	0	0
Bank	21	14	16384	2880	100.0	0	0

conditionals by how exceptional their antecedents are and compares F ’s base rank (the lowest rank at which F is no longer exceptional) against the base rank of F together with each candidate label. Both procedures reduce every query to a bounded number of calls to the workbench’s own satisfiability (SAT) solver, which is why satisfiability-call counts, not wall-clock time, are the primary efficiency measure throughout this section, following Kaliski’s complexity result for Rational Closure [12].

Correctness is measured first over every complete feature pattern each ruleset’s rule atoms admit, 2^a patterns for a rule atoms, the exhaustive population the alignment theorem is stated over, not only the patterns a training or test split happens to contain (Section 3.8). Table 1 gives, for each ruleset, the number of these patterns on which RC and MSW agree, disagree, or both abstain, reported as three separate raw counts so neither figure is hidden inside another. The agreement column follows the theorem’s own statement of agreement: the fraction of patterns where the two operators either match on a label or jointly abstain. The theorem’s three-case statement (Section 7.1 of the ECCRS specification note [17]) lists mutual abstention as one of the three ways the two operators line up, not an exception carved out of it. On all three rulesets, RC and MSW agree on every one of the 2^a patterns (64 of 64 on Breast Cancer Wisconsin, 1024 of 1024 on Mushroom, 16384 of 16384 on Bank Marketing) with zero disagreements and, on all three, zero patterns where no rule applies: every pattern each ruleset’s rule atoms admit is covered by at least one rule. This means the theorem’s two label-deciding cases are exercised exhaustively on all three rulesets, while its third case, no rule applies, is instead tested separately on the scholarship ruleset below. Restricting to the realisable subset under one-hot mutual exclusion (at most one bin of a given attribute can be true in any real instance, so some 2^a patterns over Mushroom’s and Bank Marketing’s rule atoms cannot physically occur) gives the identical figure: 512 of 512 realisable patterns for Mushroom, 2880 of 2880 for Bank Marketing, since a population with zero disagreements on the full space cannot acquire one by shrinking to a subset of it. Breast Cancer Wisconsin’s rule atoms are all independent binary indicators, so all 64 of its patterns are already realisable.

The same agreement holds over the datasets’ own instances, not only the complete pattern space. On the stratified 25% holdout, RC

and MSW agree on all 175 Breast Cancer Wisconsin instances, all 2031 Mushroom instances, and all 11303 Bank Marketing instances; over each dataset’s full instance set they agree on all 699, 8124, and 45211 instances respectively, with zero disagreements and zero mutual abstentions throughout. This paper does not report how accurate the supplied rulesets are against ground truth. Accuracy is a property of the learned ruleset the ECCRS side exported, not of this workbench, and the alignment theorem makes no claim about it, only about whether RC and MSW agree with each other; on every population measured, they do.

These agreement figures are not guaranteed by construction. The two operators are computed along disjoint paths (Figure 1): MSW is evaluated directly on the ruleset and never sees the translation, and the RC side is not degenerate either. BaseRank stratifies the translated knowledge base into four distinct ranks on Breast Cancer Wisconsin and three on Mushroom and Bank Marketing, taking 23, 24, and 46 satisfiability calls to compute those ranks (Section 3.4), so every verdict above turns on an actual rank comparison, not a lookup against a ranking collapsed to a single level. A translation that failed to preserve the specificity ordering, a BaseRank that judged exceptionality against the original knowledge base rather than the conditionals still remaining (the mistake an earlier pass made, caught only by checking it line by line against the published algorithm; Section 3.4), or an entailment condition that mishandled the infinite-rank disjunct, would each have produced disagreements on these rulesets rather than none. The override path is exercised heavily: of the 13509 holdout instances, 771 (23 on Breast Cancer Wisconsin, 332 on Mushroom, 416 on Bank Marketing) have an applicable-rule set that strictly contains the maximal-rule set (at least one applicable rule is actually overridden), and RC agrees with MSW on every one of them. The same holds over the full pattern space: 7000 of the 17472 complete feature patterns across the three rulesets (24 of 64, 320 of 1024, and 6656 of 16384) exercise the override path, and the two operators agree on every one of those too.

One implementation decision changed the Bank Marketing instance figure above materially. Bank Marketing’s full instance file records only the active bin of each one-hot attribute; the inactive bins are not written at all, rather than written as false. An earlier version of the workbench left any unmentioned atom as unknown rather than false, so a rule requiring an inactive bin to be false was spuriously judged inapplicable, and MSW abstained on 36146 of Bank Marketing’s 45211 instances, reporting instance-level agreement of 20.1% as a pure artefact of the reading, not of any actual disagreement between the operators. Completing a projected instance with false for any unmentioned atom (the closed-world reading a one-hot indicator supports, since an absent bin encodes a different active category rather than a missing observation) raised Bank Marketing’s instance-level agreement to the 100% reported above. Breast Cancer Wisconsin and Mushroom record every zero explicitly and are unaffected by either reading.

A second, independent check comes from outside this codebase entirely: the datasets ship an independent implementation’s output for the same rulesets, verdicts on the three structural conditions checked with the z3 SMT solver, and Most Specific Wins predictions per instance including applicable and inclusion-maximal rules. Table 2 compares this workbench’s validator and MSW operator

against that output, instance by instance, on the label, the applicable-rule set, the maximal-rule set, and all three structural conditions, across all 13509 instances in the three datasets’ stratified holdouts. Every axis matches exactly: no instance’s label, applicable set, or maximal set differs between the two implementations, and both agree on all nine condition checks. This includes the 771 instances discussed above where the override path fires: this workbench’s applicable set, maximal set, and predicted label agree with the independent implementation’s on every one of them, not only on the instances a single rule decides outright.

The divergence table this section owes for each ruleset (the specific pattern, which condition failed, and how the two predictions differ) is empty for all three exported rulesets: the validator (Section 3.2) raises no violation of strict global exception closure, sanity, or no total override on any of them, and Table 1 already shows zero disagreements. An empty divergence table on a validated ruleset is itself the result the alignment theorem predicts once its three hypotheses are confirmed to hold, not merely the absence of a finding.

The scholarship ruleset introduced in Section 3.2 is the one case where the table is not empty. Its five rules (a high-GPA default, a disciplinary-flag exception to it, a research-project exception to that exception, a community-service default, and the same disciplinary-flag exception applied to it) violate strict global exception closure three times: rules r_1/r_5 , r_2/r_4 , and r_3/r_5 each carry opposite labels and can both apply to the same pattern, without either body being a strict subset of the other. Over its full $2^5 = 32$ -pattern space, RC and MSW agree on 19 patterns, disagree on exactly 1, and both abstain on 12, an agreement figure of 96.9% (31 of 32) under the same both-abstain-counts-as-agreement convention Table 1 uses. The one divergence is at the pattern where every atom is true: r_3 (body $a_1 \wedge a_4 \wedge a_5$, label 1) and r_5 (body $a_2 \wedge a_3 \wedge a_4$, label 0) are both applicable and both inclusion-maximal, since neither body contains the other, so MSW abstains on the tie, exactly the pair the validator flags as violating strict closure. RC does not abstain here; it entails class 1. The two operators disagree at precisely the pattern where the theorem’s hypotheses fail, and nowhere else in the scholarship ruleset’s pattern space.

Table 3 gives the two-sided trace the workbench produces for Instance A of the specification note [17], $F_A = a_1 \wedge \neg a_2 \wedge \neg a_3 \wedge a_4 \wedge a_5$, reproduced from the workbench’s own output. On the ECCRS side, r_1 applies and is overridden by r_2 , which is overridden by r_3 ; r_3 is the sole inclusion-maximal applicable rule, so MSW predicts class 1. On the RC side, the base rank of F_A is 2, of $F_A \wedge m$ also 2, and of $F_A \wedge \neg m$ infinite, so the entailment condition holds and RC entails class 1; the decisive conditional is at rank 2 and traces back to r_3 . The two traces converge on the same rule, as Section 3.6 predicts.

Efficiency is measured on the same three rulesets, using satisfiability-call counts as the primary metric (Section 3.5). Three procedures are compared over the full pattern space of each ruleset: the naive published ascending scan, the binary-search procedure currently shipped, and the class-collapse optimisation, which evaluates one representative pattern per applicable-rule-set class instead of one per pattern. Figure 2 plots the total calls each procedure makes; the exact counts, naive/shipped/collapsed, are 360/400/64 on Breast Cancer Wisconsin, 5440/6144/66 on Mushroom, and 92160/98304/102 on Bank Marketing.

Table 2: Cross-check of this workbench’s validator and MSW operator against an independent implementation supplied with the datasets, on the same stratified holdouts reported above (175/2031/11303 instances). “3 conditions” is strict global exception closure, sanity, and no total override together.

Dataset	Instances	MSW label	Applicable set	Maximal set	3 conditions
Breast Cancer Wisconsin	175	175/175	175/175	175/175	3/3
Mushroom	2031	2031/2031	2031/2031	2031/2031	3/3
Bank Marketing	11303	11303/11303	11303/11303	11303/11303	3/3
Total	13509	13509/13509	13509/13509	13509/13509	9/9

Table 3: Two-sided explanation trace for the scholarship worked example’s Instance A, $F_A = a_1 \wedge \neg a_2 \wedge \neg a_3 \wedge a_4 \wedge a_5$.

ECCRS side	RC side
$r_1: a_1 \Rightarrow 1$ (overridden by r_2, r_3)	Candidate queries: does F_A typically entail m (class 1), or typically entail $\neg m$ (class 0)?
$r_2: a_1 \wedge a_4 \Rightarrow 0$ (overridden by r_3)	$\text{br}(F_A) = 2, \text{br}(F_A \wedge m) = 2, \text{br}(F_A \wedge \neg m) = \infty$
$r_3: a_1 \wedge a_4 \wedge a_5 \Rightarrow 1$ (inclusion-maximal)	$\text{br}(F_A) < \text{br}(F_A \wedge \neg m)$ holds, so RC entails class 1
MSW label: 1	Decisive conditional at rank 2, from ECCRS rule r_3
Both sides agree: class 1.	

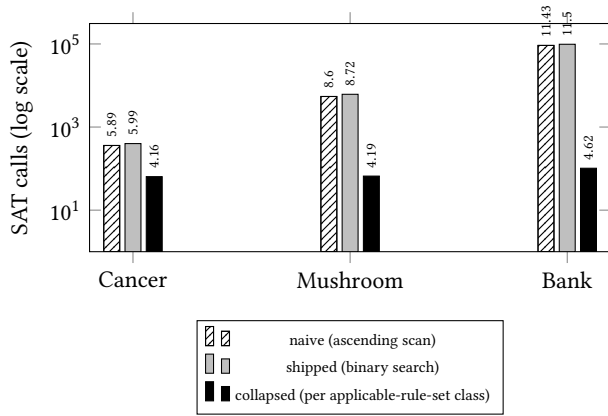


Figure 2: Total satisfiability calls the pattern-space agreement analysis makes under three procedures, per dataset. Note the log scale: the collapsed bars are one to three orders of magnitude smaller than the other two on Mushroom and Bank Marketing.

Two results follow, in opposite directions. First, the binary search costs more satisfiability calls than the ascending scan it replaced, on all three rulesets (400 against 360, 6144 against 5440, and 98304 against 92160) because every ruleset here resolves into only three or four distinct ranks, and a binary search commits to a fixed sequence of probes regardless of where the answer turns out to be, while an ascending scan pays only for the ranks it actually inspects. This is a measured non-win, not a benchmarking artefact: the asymptotic argument, $O(\log k)$ against $O(k)$, is not wrong, but it does not pay off at the $k = 3\text{--}4$ distinct ranks available here.

Second, the class-collapse optimisation removes nearly all of these calls. Grouping the 2^a patterns by which rules they make applicable reduces the number of RC evaluations needed from 64, 1024, and 16384 patterns to 10, 11, and 17 applicable-rule-set classes,

and cuts the shipped procedure’s satisfiability calls by 6-fold, 93-fold, and 964-fold on the same three rulesets (6-fold, 82-fold, and 904-fold against the naive scan). Wall-clock time falls with it: from 3.24 seconds under the shipped procedure to 0.10 seconds under the class collapse on Bank Marketing, the only ruleset large enough for the difference to be measured in anything but milliseconds. The grouped computation was checked pattern by pattern against the ungrouped one on all three exported rulesets’ pattern spaces (17472 patterns in total) with zero label mismatches; Section 3.5 reports the same check extended to the scholarship counter-example and its closed positive-control variant.

What follows interprets these figures. All three exported rulesets pass validation, and on all three, RC and MSW agree on every complete feature pattern, every realisable pattern, and every supplied instance, with zero disagreements and zero mutual abstentions; an independent implementation reaches the same MSW verdict on 13509 of 13509 instances and the same validator verdict on all nine condition checks. Had any of the three validated rulesets shown a disagreement, that would have been a serious finding, since it is exactly what the theorem’s hypotheses are supposed to rule out; none did. The scholarship ruleset, which fails validation, is the only one where RC and MSW disagree, and the single pattern where they disagree is exactly the pattern the validator’s strict-closure check flags. The theorem holding precisely where its hypotheses hold, and failing precisely where they do not, is evidence for the theorem. But it is evidence from three rulesets and one small counter-example, not a general proof. A ruleset that satisfies the three conditions and still disagrees somewhere has not been found here, but it has not been ruled out either.

Reporting satisfiability-call counts as the primary measure follows from Kaliski’s complexity result, that Rational Closure calls a classical entailment oracle a polynomial number of times rather than being harder than that oracle itself [12]; the roughly flat six calls per query observed across all three rulesets (Section 3.5) is consistent with that bound, and the class-collapse optimisation’s payoff comes entirely from reducing how many queries are made,

not from making each one cheaper. None of the abstentions or disagreements in this paper’s results trace back to Rational Closure’s known inheritance-blocking limitation (Section 2.4): the three validated rulesets have no abstentions or disagreements to explain, and the scholarship ruleset’s one disagreement is fully accounted for by its strict-closure violation. This is expected: sanity and strict global exception closure rule out the infinite-rank inconsistency the inheritance limitation depends on, though a validated ruleset that exposed the limitation anyway would point to either a validation gap or a case the theorem’s proof does not cover. None of the rulesets available produces an infinite rank in the first place, so this limitation was not directly exercised either way.

These figures are specific to the Rational Closure component of the wider project. Lexicographic Closure and System W, built by Mochoeneng [19] and Baker [2] on the same BaseRank ranking implemented here, are evaluated separately in their own papers; whether either method’s greater permissiveness recovers a conclusion Rational Closure’s ranking blocks is a question this paper does not answer, since neither implementation nor its results are part of this codebase.

Every figure above is reported as obtained on the three exported rulesets and the scholarship counter-example available at the time of writing. Where a result was negative, such as the binary search’s call-count regression, or partial, such as the pattern-enumeration limit this paper’s largest ruleset does not come close to testing, it is reported as such.

5 Conclusions

On all three rulesets validated and tested for this project (exported for the Breast Cancer Wisconsin, Bank Marketing, and Mushroom datasets), Rational Closure and Most Specific Wins agree exactly: every complete feature pattern each ruleset covers receives the same label from both operators, 64 patterns for Breast Cancer Wisconsin, 1024 for Mushroom, and 16384 for Bank Marketing, with no abstentions on either side. This is consistent with the alignment theorem on the rulesets tested; three rulesets, however clean the result, establish only that it holds where we have checked it, not that it holds universally. The theorem itself is recent and unpublished, and this project is, as far as we are aware, the only one with access to it, so the honest claim to priority is a narrow one: this is the first time it has been checked against rulesets exported from classifiers learned on real data rather than a small worked example, not that it has been tested more broadly than that.

The optimised procedure the theorem’s practicality depends on is further along than it was, though not every measurement went the way we expected going in. The applicable-rule-set collapse cuts the satisfiability calls the agreement analysis needs by up to 964-fold on Bank Marketing (904-fold against the naive scan), differentially tested against the ungrouped computation on all five rulesets used in this paper. The binary search shipped in its place has the better asymptotic bound, $O(\log k)$ against $O(k)$, but measured, costs more calls than the ascending scan it replaced on all three rulesets tested. A Horn-clause fast path was designed but not built, since its likely payoff was small next to the class collapse’s. The class collapse still stops at the same 20-rule-atom enumeration limit the naive

procedure had; only a ruleset with more rule atoms than rules would make the rejected rule-subset search (Section 3.5) worth revisiting.

The two-sided explanation trace does what it was designed to do on the cases examined by hand: the scholarship worked example’s default, exception, and exception-to-exception chain on the ECCRS side lines up, rule for rule, with the rank comparison Rational Closure reports on the other side. Whether an outside reader would find it legible, not just correct, is untested outside the project.

The clearest limitation now is not breadth of testing but scale. We have validated and tested three rulesets, not a broader sample, and none approaches the 34-attribute Ionosphere ruleset originally targeted before it, Breast Cancer Wisconsin, and German Credit were replaced with the three rulesets the ECCRS side actually exported (Section 3.7). Bank Marketing is the largest available, with 21 rules over 14 rule atoms against Breast Cancer Wisconsin’s six, and it is where the class collapse’s headline reduction was measured; but fourteen rule atoms is still well inside the 20-atom enumeration limit, so this paper cannot say what happens as that limit is approached, only that none of the three rulesets comes close to testing it.

Two things follow. A ruleset with more rule atoms than rules, or one that approaches the enumeration limit the class collapse does not lift, would be the natural next test of how far Section 3.5’s approach generalises; none of the three rulesets here puts either question under real pressure. And, outside this paper’s scope, connecting these results to the Lexicographic Closure and System W components once they reach the same point would let where the two-sided trace holds, and where the stronger methods diverge from Rational Closure, be reported together rather than compared after the fact across three papers.

AI Usage Declaration

Generative AI tools were used during the preparation of this work in the following ways:

- Reviewing drafts of the manuscript for clarity, structure, and internal consistency (e.g., flagging duplicated content across sections, inconsistent figures, and unclear terminology).
- Checking the paper against the intended contribution and framing (e.g., whether claims in the abstract, results, and conclusion were mutually consistent).
- Identifying factual or attribution errors for correction.
- Debugging and checking code used in the implementation and evaluation.
- Critiquing design and architecture decisions during development, including surfacing alternative approaches when making significant design choices.

All substantive edits, code changes, and design decisions made in response to AI feedback were authored and verified by the author.

References

- [1] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. 2018. Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research* 18, 234 (2018), 1–78. <http://jmlr.org/papers/v18/17-716.html>
- [2] Antony Baker. 2026. System W Honours Paper. (2026). Unpublished honours thesis, University of Cape Town.
- [3] Giovanni Casini, Umberto Straccia, and Thomas Meyer. 2019. A Polynomial Time Subsumption Algorithm for Nominal Safe $\text{ELO}_{\perp}$ under Rational Closure. *Information Sciences* 501 (2019), 588–620. Also available as arXiv:1802.08201.
- [4] Victoria Chama. 2020. Explanation for defeasible entailment. (2020). Master's Thesis, University of Cape Town. <http://hdl.handle.net/11427/32206>
- [5] V. Chama, S. Wang, T. Meyer, and G. Casini. 2024. Defeasible justification for KLM-style logic. In *CEUR WORKSHOP PROCEEDINGS*, vol. 3739. Bergen, Norway, 18-21/06/2024. CEUR. <https://ceur-ws.org/Vol-3739/paper-3.pdf>
- [6] Adnan Darwiche and Auguste Hirth. 2023. On the (complete) reasons behind decisions. *Journal of Logic, Language and Information* 32, 1 (2023), 63–88. doi:10.1007/s10849-022-09377-8
- [7] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. arXiv:1702.08608 [stat.ML] <https://arxiv.org/abs/1702.08608>
- [8] Niklas Eén and Niklas Sörensson. 2003. Temporal Induction by Incremental SAT Solving. *Electronic Notes in Theoretical Computer Science* 89, 4 (2003), 543–560. doi:10.1016/S1571-0661(05)82542-3
- [9] Andreas Fuster, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther. 2022. Predictably Unequal? The Effects of Machine Learning on Credit Markets. *The Journal of Finance* 77, 1 (2022), 5–47. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/jofi.13090> doi:10.1111/jofi.13090
- [10] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 80–89. <https://arxiv.org/abs/1806.00069>
- [11] Reijo Jaakkola, Tomi Janhunen, Antti Kuusisto, Masood Feyzbakhsh Rankooh, and Miikka Vilander. 2023. Short Boolean Formulas as Explanations in Practice. In *Logics in Artificial Intelligence*, Sarah Gaggl, Maria Vanina Martinez, and Magdalena Ortiz (Eds.). Springer Nature Switzerland, Cham, 90–105. doi:10.1007/978-3-031-43619-2_7
- [12] Adam Kaliski. 2020. An overview of KLM-style defeasible entailment. <http://hdl.handle.net/11427/32743>
- [13] Christian Komo and Christoph Beierle. 2022. Nonmonotonic reasoning from conditional knowledge bases with system W. *Annals of Mathematics and Artificial Intelligence* 90, 1 (2022), 107–144. doi:10.1007/s10472-021-09777-9
- [14] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44, 1 (1990), 167–207. doi:10.1016/0004-3702(90)90101-5
- [15] Daniel Lehmann. 1995. Another perspective on default reasoning. *Annals of Mathematics and Artificial Intelligence* 15 (1995), 61–82. doi:10.1007/BF01535841
- [16] Daniel Lehmann and Menachem Magidor. 1992. What does a conditional knowledge base entail? *Artificial Intelligence* 55, 1 (1992), 1–60. doi:10.1016/0004-3702(92)90041-U
- [17] Ruvarashe Madzime. 2026. ECCRS in a Nutshell for the Honours Project: Semantics, Most-Specific-Wins Operator, and the Rational Closure Alignment. (2026). Unpublished technical note, University of Cape Town.
- [18] Ruvarashe Shalom Madzime, Thomas Meyer, and Louise Leenen. 2025. An Override-Aware Classifier for Transparent AI. In *Proceedings of the Southern African Conference for Artificial Intelligence Research* (Cape Town, South Africa), Aurna Gerber and Anban Pillay (Eds.). <https://2025.sacair.org.za/online-proceedings/proceedings.html>
- [19] Augustine Mochoeneng. 2026. Lexicographic Closure Honours Paper. (2026). Unpublished honours thesis, University of Cape Town.
- [20] Fernando E. B. Otero and Alex A. Freitas. 2013. Improving the Interpretability of Classification Rules Discovered by an Ant Colony Algorithm. In *Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation (GECCO '13)*. ACM, 73–80. doi:10.1145/2463372.2463382
- [21] Raymond Reiter. 1980. A logic for default reasoning. *Artificial Intelligence* 13, 1-2 (1980), 81–132. doi:10.1016/0004-3702(80)90014-4
- [22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1135–1144. doi:10.1145/2939672.2939778
- [23] Tjitze Rienstra. 2023. Explaining Boolean classifiers with non-monotonic background theories. In *Benelux Conference on Artificial Intelligence*. Springer, 174–188. doi:10.1007/978-3-031-74650-5_10
- [24] Cynthia Rudin. 2019. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. arXiv:1811.10154 [stat.ML] <https://arxiv.org/abs/1811.10154>