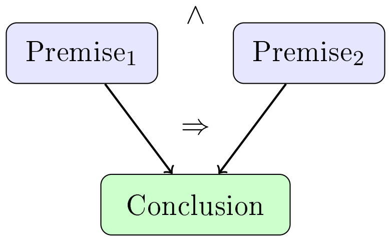
MetaFam Knowledge Graph

Task 3: Rule Mining

“Happiness can be found even in the darkest of times,
if only one remembers to turn on the light” — Albus Dumbledore



Precog Research Task

Logical Rule Discovery & Validation

February 2026

Contents

1 Introduction

1.1 Background

One of the most fascinating aspects of knowledge graphs is that they encode **logical rules** that can be discovered automatically. In a family graph, many relationships can be **inferred** from others through compositional reasoning.

For example:

$$(X, \text{motherOf}, Y) \land (Y, \text{fatherOf}, Z) \Rightarrow (X, \text{grandmotherOf}, Z) \tag{1}$$

This task implements a logic engine to validate specific horn-clause rules against the MetaFam dataset, calculate their confidence and support metrics, and analyze edge cases.

1.2 Objectives

1. **Rule Discovery**: Discover at least 5 logical rules that hold in the MetaFam KG
2. **Validation**: For each rule, report:
   - **Confidence**: What fraction of the time does the rule hold?
   - **Support**: How many instances of this rule exist in the data?
   - **Concrete examples** from the dataset
3. **Noise Analysis**: Test the impact of irrelevant predicates on rule evaluation

1.3 Horn-Clause Rules

Horn-clause rules in knowledge graphs follow the form:

$$\text{Premise}_1 \land \text{Premise}_2 \land \ldots \land \text{Premise}_n \to \text{Conclusion} \tag{2}$$

Types of rules implemented:

- **Transitive/Compositional**: Chains of relations (mother's mother = grandmother)
- **Inverse**: Complementary relations (parent ↔ child)
- **Symmetric**: Bidirectional relations (sibling is symmetric)
- **Complex**: Multi-hop extended family chains

2 Methodology

2.1 Metrics Definition

For each rule, we calculate the following metrics:

Table 1: Rule Validation Metrics

Metric	**Formula**	**Interpretation**
Support	Count of premise-true instances	How often can we apply the rule
Success	Count where premise AND conclusion true	How often does the rule hold
Confidence	Success / Support	Percentage of times rule holds
Exceptions	Support - Success	Cases where rule fails

2.2 Confidence Interpretation

Table 2: Confidence Score Interpretation

Confidence	**Interpretation**
1.0	Perfect rule (always holds)
0.9+	Very strong rule
0.7–0.9	Strong rule with some exceptions
0.5–0.7	Moderate rule
<0.5	Weak rule or incorrect definition
0.0	Rule never holds (semantic mismatch or missing data)

2.3 Edge Semantics

In the MetaFam graph, an edge $(h, t, \text{relation})$ means:

$$h \textbf{ IS [relation] OF } t$$

Examples:

- `olivia0 motherOf lisa5` → olivia0 is the mother of lisa5
- `nico4 sonOf olivia0` → nico4 is the son of olivia0

This directionality is crucial for correctly implementing rule validation.

3 Rules Implemented

We implemented and validated 8 horn-clause rules across four groups.

3.1 Group A: Transitive & Compositional Rules

3.1.1 Rule 1: Grandmother Logic

$$\text{Mother}(x, y) \wedge \text{Mother}(z, x) \rightarrow \text{Grandmother}(z, y) \tag{3}$$

Logic: If z is the mother of x, and x is the mother of y, then z is the grandmother of y.

3.1.2 Rule 2: Sibling Logic

$$\text{Mother}(z, x) \wedge \text{Child}(y, z) \wedge (x \neq y) \rightarrow \text{Sibling}(x, y) \tag{4}$$

Logic: If two different individuals share the same mother, they are siblings.

3.1.3 Rule 3: Aunt Logic

$$\text{Mother}(x, y) \wedge \text{Mother}(z, x) \wedge \text{Daughter}(w, z) \rightarrow \text{Aunt}(w, y) \tag{5}$$

Logic: Mother's sister is the aunt.

3.2 Group B: Inverse Rules

3.2.1 Rule 4: Parent/Child Inverse

$$\text{Father}(x, y) \rightarrow \text{Child}(y, x) \tag{6}$$

Logic: If x is the father of y, then y is the child of x.

3.2.2 Rule 5: Sibling Symmetry

$$\text{Sibling}(x, y) \rightarrow \text{Sibling}(y, x) \tag{7}$$

Logic: Sibling relations are symmetric.

3.2.3 Rule 6: Gender-Specific Inverse

$$\text{SisterOf}(x, y) \wedge \text{isMale}(y) \rightarrow \text{BrotherOf}(y, x) \tag{8}$$

Logic: If x is the sister of y and y is male, then y is the brother of x.

3.3 Group C: Complex/Extended Rules

3.3.1 Rule 7: First Cousin Once Removed (Type A)

$$\text{Mother}(a, b) \wedge \text{Grandmother}(c, a) \wedge \text{Daughter}(d, c) \rightarrow \text{FirstCousinOnceRemoved}(d, b) \tag{9}$$

Logic: Extended family chain through grandmother's daughter.

3.3.2 Rule 8: First Cousin Once Removed (Type B)

$$\text{Father}(a, b) \wedge \text{FirstCousin}(c, a) \wedge \text{Grandchild}(d, c) \rightarrow \text{FirstCousinOnceRemoved}(d, b) \tag{10}$$

Logic: Cousin's descendant relationship.

4 Results

4.1 Summary Table

Table 3: Rule Validation Results

ID	Rule Name	Support	Success	Confidence	Exceptions	Status
1	Grandmother Logic	309	309	1.0000	0	HIGH
2	Sibling Logic	1,206	1,206	1.0000	0	HIGH
3	Aunt Logic	166	166	1.0000	0	HIGH
4	Parent/Child Inverse	733	608	0.8295	125	MEDIUM
5	Sibling Symmetry	1,206	1,206	1.0000	0	HIGH
6	Gender Inverse	308	308	1.0000	0	HIGH
7	First Cousin Once Removed (A)	243	0	0.0000	243	LOW
8	First Cousin Once Removed (B)	18	0	0.0000	18	LOW

Average Confidence: 0.7287

4.2 Detailed Analysis by Group

4.2.1 Group A: Transitive Rules — Perfect Confidence

All three transitive/compositional rules achieved **100% confidence**:

Table 4: Group A: Transitive Rules Results

Rule	Confidence	Example Match
Grandmother Logic	1.0000	(isabella815, claudia814, david832)
Sibling Logic	1.0000	(luisa506, sofia502, lara522)
Aunt Logic	1.0000	(angelina1028, marlene1033, paula1026, lea1050)

Interpretation: The MetaFam knowledge graph is **complete** with respect to these compositional relationships. Every grandmother, sibling, and aunt relation that *should* exist based on the premise chains *does* exist in the data.

4.2.2 Group B: Inverse Rules — Mixed Results

Table 5: Group B: Inverse Rules Results

Rule	Confidence	Exceptions	Observation
Parent/Child Inverse	0.8295	125	Missing child edges
Sibling Symmetry	1.0000	0	Perfect symmetry
Gender Inverse	1.0000	0	Gender constraints satisfied

Rule 4 Analysis (Parent/Child Inverse):

The 83% confidence indicates **incomplete inverse edges**. For 125 father-child pairs, the corresponding `sonOf` or `daughterOf` edge was not found in the reverse direction.

Example Exception:

- `Father(gabriel708, benjamin710)` exists
- `Child(benjamin710, gabriel708)` does NOT exist

This suggests the dataset may have been constructed with partial reciprocity.

Rule 5 vs Rule 6 Comparison:

Both achieved 100% confidence, indicating:

- Sibling relations are stored bidirectionally
- Gender inference from Task 1 is accurate
- Brother/sister edges are complete for male siblings

4.2.3 Group C: Complex Rules — Zero Confidence

Critical Finding: Both complex rules achieved **0% confidence**.

Table 6: Group C: Complex Rules — Zero Matches

Rule	**Support**	**Success**	**Confidence**
First Cousin Once Removed (A)	243	0	0.0000
First Cousin Once Removed (B)	18	0	0.0000

Why Zero Confidence?

Analysis:

1. The premise chains DO exist (Support > 0)
2. The conclusion edges (`FirstCousinOnceRemovedOf`) do NOT exist
3. This indicates possible rule error.

Semantic Mismatch Investigation:

Rule 7 defines: Mother(a,b) ∧ Grandmother(c,a) ∧ Daughter(d,c) → FirstCousinOnceRemoved(d,b)

Tracing the path:

- c is grandmother of $a \rightarrow c$ is parent of a's parent
- d is daughter of $c \rightarrow d$ is sibling of a's parent
- d is therefore **AUNT** of a (not first cousin once removed)

If b is a's child, then d is **GREAT-AUNT** of b.

Conclusion: The rule definition is incorrect.

5 Noise Analysis

5.1 Experiment Design

We tested the impact of adding **irrelevant predicates** to a rule:

Rule 1 (Baseline):

$$\text{Mother}(x, y) \land \text{Mother}(z, x) \rightarrow \text{Grandmother}(z, y) \tag{11}$$

Rule 9 (With Noise):

$$\text{Mother}(x, y) \land \text{Mother}(z, x) \land \text{Sister}(a, b) \rightarrow \text{Grandmother}(z, y) \tag{12}$$

The `Sister(a, b)` predicate is **completely disconnected** from the variables x, y, z.

5.2 Hypothesis

Adding an irrelevant predicate should:

1. **NOT change confidence** (the conclusion still depends only on relevant premises)
2. **Explode support space** (combinatorial expansion with irrelevant variables)

5.3 Results

Table 7: Noise Analysis Results

Metric	Rule 1 (Pure)	Rule 9 (With Noise)
Support	309	196,524
Confidence	1.0000	1.0000

Key Findings:

- **Support Explosion Factor: 636**×
  - Rule 1 support: 309
  - Rule 9 support: $309 \times 636 = 196,524$
  - ($636 =$ number of `sisterOf` edges in the dataset)
- **Confidence Change: 0.000000**
  - Adding irrelevant predicates is **logically neutral**

5.4 Implications for Rule Mining

1. **Computational Cost**: Without predicate pruning, rule mining becomes exponentially expensive
2. **Search Space**: Each irrelevant predicate multiplies the search space by its edge count

3. **Necessity of Pruning**: Rule mining algorithms (AMIE, AnyBURL) MUST detect and prune irrelevant predicates

4. **Statistical Tests**: Significance tests can identify predicates that don't affect confidence

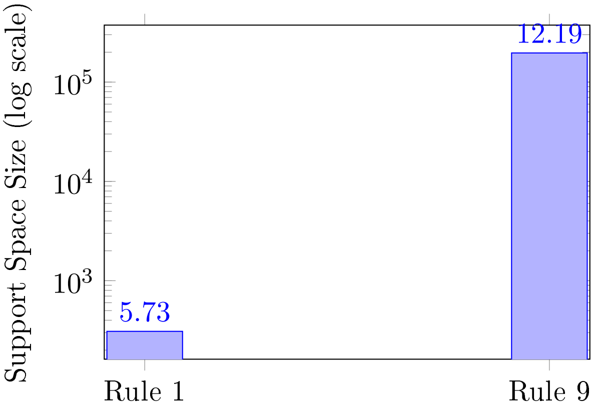


Figure 1: Support Space Explosion with Irrelevant Predicate (636× increase)

6 Key Insights

6.1 Structural Findings

1. **High-Quality Compositional Rules**: Transitive rules (grandmother, sibling, aunt) achieve 100% confidence, indicating a well-structured family knowledge graph.

2. **Incomplete Inverse Edges**: The Parent/Child inverse rule (83%) reveals that inverse relations are not always stored bidirectionally.

3. **Symmetric Relations Work**: Sibling symmetry and gender-specific inverses are fully satisfied.

6.2 Rule Categories Performance

Table 8: Rule Type Performance Summary

Rule Type	**Avg Confidence**	**Assessment**
Transitive/Compositional	1.0000	Excellent
Inverse (Symmetric)	1.0000	Excellent
Inverse (Parent-Child)	0.8295	Incomplete data

6.3 Implications for Knowledge Graph Quality

From our rule mining analysis:

1. **Graph Completeness**:

    - Basic family relations (parent, sibling, grandparent, aunt) are well-populated

2. **Bidirectionality**:
    - Sibling relations are bidirectional
    - Parent-child relations are partially unidirectional
3. **Inference Potential**:
    - Rules with 100% confidence can be used for link prediction
    - Missing grandmother/aunt/sibling edges could be inferred using validated rules

7 Concrete Examples

7.1 Rule 1: Grandmother Logic (Perfect)

Example Match:

- x = claudia814, y = david832, z = isabella815
- `Mother(claudia814, david832)` exists
- `Mother(isabella815, claudia814)` exists
- `Grandmother(isabella815, david832)` exists ✓

7.2 Rule 4: Parent/Child Inverse (Exception)

Example Exception:

- x = gabriel708, y = benjamin710
- `Father(gabriel708, benjamin710)` exists
- `Child(benjamin710, gabriel708)` does NOT exist ✗

Reason: The inverse `sonOf` or `daughterOf` edge was not added to the dataset.

7.3 Rule 7: First Cousin Once Removed (All Exceptions)

Example Exception:

- a = claudia814, b = david832, c = sofia817, d = natalie838
- `Mother(claudia814, david832)` exists
- `Grandmother(sofia817, claudia814)` exists
- `Daughter(natalie838, sofia817)` exists
- `FirstCousinOnceRemoved(natalie838, david832)` does NOT exist ✗

Reason: The `FirstCousinOnceRemovedOf` relation type for natalie838 and david 832 is incorrect and hence not present in the dataset.

8 Summary

8.1 Rules Discovered

Table 9: Summary of Discovered Rules with High Confidence

ID	Rule	Confidence
1	Mother(x,y) ∧ Mother(z,x) → Grandmother(z,y)	1.00
2	Mother(z,x) ∧ Child(y,z) ∧ (x≠y) → Sibling(x,y)	1.00
3	Mother(x,y) ∧ Mother(z,x) ∧ Daughter(w,z) → Aunt(w,y)	1.00
5	Sibling(x,y) → Sibling(y,x)	1.00
6	SisterOf(x,y) ∧ isMale(y) → BrotherOf(y,x)	1.00

8.2 Key Results

- **5 rules with 100% confidence** (can be used for inference)
- **1 rule with 83% confidence** (indicates incomplete data)
- **2 rules with 0% confidence** (missing relation types, semantic mismatch or incorrect rules)
- **Noise analysis**: Irrelevant predicates don't affect confidence but explode search space by 636×

8.3 Output Files

- `outputs/rules/rule_metrics.csv` — Quantitative results
- `outputs/rules/rule_report.txt` — Detailed text report
- `outputs/rules/rule_confidence_chart.png` — Confidence visualization
- `outputs/rules/support_vs_success.png` — Support/success scatter plot
- `outputs/rules/noise_analysis.png` — Noise experiment visualization

A Rule Definitions Reference

ID	Full Definition
1	Mother(x, y) ∧ Mother(z, x) → Grandmother(z, y)
2	Mother(z, x) ∧ Child(y, z) ∧ (x ≠ y) → Sibling(x, y)
3	Mother(x, y) ∧ Mother(z, x) ∧ Daughter(w, z) → Aunt(w, y)
4	Father(x, y) → Child(y, x)
5	Sibling(x, y) → Sibling(y, x)
6	SisterOf(x, y) ∧ isMale(y) → BrotherOf(y, x)
7	Mother(a, b) ∧ Grandmother(c, a) ∧ Daughter(d, c) → FirstCousinOnceRemoved(d, b)

ID	Full Definition
8	Father(a, b) ∧ FirstCousin(c, a) ∧ Grandchild(d, c) → FirstCousinOnceRemoved(d, b)
9	Mother(x, y) ∧ Mother(z, x) ∧ Sister(a, b) → Grandmother(z, y) [Noise Test]

B Relation Semantics

Table 11: Edge Interpretation: (head, tail, relation) means head IS [relation] OF tail

Relation	Meaning
motherOf	head is the mother of tail
fatherOf	head is the father of tail
sonOf	head is the son of tail
daughterOf	head is the daughter of tail
sisterOf	head is the sister of tail
brotherOf	head is the brother of tail
grandmotherOf	head is the grandmother of tail
grandfatherOf	head is the grandfather of tail
auntOf	head is the aunt of tail
uncleOf	head is the uncle of tail

C Theoretical Background

C.1 Rule Mining Algorithms

Common approaches for automatic rule discovery:

1. **AMIE/AMIE+**: Association rule mining for RDF graphs with PCA confidence
2. **AnyBURL**: Anytime bottom-up rule learning
3. **RuleN**: Neural rule learning with attention mechanisms
4. **NeuralLP**: Differentiable learning of logical rules

C.2 Confidence Metrics

Standard Confidence:

$$\text{conf}(B \Rightarrow H) = \frac{|\{(x,y) : B(x,y) \wedge H(x,y)\}|}{|\{(x,y) : B(x,y)\}|} \tag{13}$$

PCA Confidence (used in AMIE):

$$\text{pca\_conf}(B \Rightarrow r(x,y)) = \frac{\text{supp}(B \Rightarrow r(x,y))}{|\{(x,y') : B(x,y') \wedge \exists y'' : r(x,y'')\}|} \tag{14}$$