A Representative Use Cases

Team/Platform Focus	Specific Responsibilities	Code Analysis Capability and System Integration
R&D Efficiency	Code Metrics Analysis	Supports multi-language code metric analysis.
R&D Efficiency	Application Architecture Governance	Provides architectural metric analysis results.
Legal Compliance	Compliance Algorithm Monitoring	Analyzes code to detect if the algorithm supports personalized switches.
Code Change QA	Intelligent Change Analysis and Release Risk Assessment	Analyzes changed code for interface coverage and impact analysis.
Information Flow Analysis	Intelligent Analysis of Single System Information Flow	Maps all parameter relationships, up to upstream and downstream interfaces, which can be used as a base for rule comparison.
Gray-Scale Releasing	Automatic Generation of Gray Rules for Business Systems	Analyzes service interface parameter consumption and traceability relationships, aiding in the improvement of gray change lines and overall coverage.
App Size Reduction	Reducing the Size of iOS and Android Apps	Analyzes iOS and Android code to identify unused code resources.
Mutation Testing	Mutant Parameter Analysis and Static Code Analysis.	Analyzes code to supplement test cases and check test case coverage.
Application Security	Online Application Risk Prevention	Analyzes code to detect dangerous functions in the link.
Change Risk Assessment Platform	Technical Risk Analysis	Provides metadata information for files including java, xml, properties, etc.
QA Data Center for Cloud	Gathering QA-related Data for Cloud Platform	Extracts data models from source code and configuration files.
Intelligent Change Analysis	Intelligent Hosting	Evaluates the complexity of change code to judge if the change can be hosted.
Intelligent Testing	Code Risk Identification	Provides the source code of the corresponding interface or implementation class function.
Information Security Analysis	Data Security Code Interface Scan	Triggers a full-site code interface scan, collecting and summarizing all interfaces (http, tr) and corresponding interface parameters in the code repository.
Mini-program Privacy Compliance	Privacy Risk Detection for Mini Programs	Analyzes code to detect incidents that violate privacy rules.
Architecture Assets and Governance	Architecture Assets Management and Governance	Identifies the middleware framework information used in applications in batch.
Change Impact Analysis	Code Change Impact Analysis	Analyzes changed code to perform code change content analysis and change impact analysis.
Network Attack and Defense Exercise	Risk Point Injection Attack	Extracts code features such as classes, methods, variables, etc., to aid in the construction of the attack denominator.
Manual Test Case Recommendation	Intelligent Recommendation of Test Cases for Manual Testing	Analyzes changed code to perform code change link analysis and service interface information query.
Internal Coding Standards	Internal guidelines established to ensure consistency, readability, and maintainability of code.	Analyzes code by implementing the coding standard checkers.

Table 9: Representative Use Cases

B Query Script List

Category	Script Name	Programming Languages
Category 1: Code Measure	Q1 Code Comment Ratio Query	Java, Python, Js/Ts, Go
	Q2 Code Cyclomatic Complexity Query	Java, Python, Js/Ts, Go
	Q3 Code AST Query	Java, Python, Js/Ts, Go
	Q4 Code Reusability with Jar Query	Java
	Q5 Code Reusability with Http Api	Java
	Q6 Code Reusability with Rpc Api Query	Java, Xml
	Q7 Code Call Graph Query	Java
	Q8 Auto-generated Code Query	Java, Python, Js/Ts, Go
Category 2: Architecture Smell	Q9 Halstead Vocabulary Query	Java
	Q10 Fan-In/Fan-Out Query	Java
	Q11 Mutual Recursive Call Query	Java
	Q12 Depth of Inheritance Tree Query	Java
	Q13 Number of Cyclic Hierarchies Query	Java
	Q14 Find Duplicate Import Jar Query	Java
	Q15 Lack of Cohesion of Methods Query	Java
	Q16 Unused Import Query	Java
	Q17 Overlapping Interfaces Query	Java
	Q18 Overriding Methods Query	Java
	Q19 Call Chain by Given Method	Java
Category 3: Risk Analysis Meta Info Model	Q20 Call Graph with Root Query	Java
	Q21 Class Hierarchy Tree Query	Java
	Q22 Xml Dal Setting Query	Java, Xml
	Q23 Xml Pom Dependency Query	Xml
	Q24 Xml Sofa Reference Query	Xml
	Q25 Xml Sofa Consumer Query	Xml
	Q26 Xml Common Drm Config Query	Xml
	Q27 Xml Bean Query	Xml
	Q28 Xml Log Setting Query	Xml
	Q29 Properties Setting Query	Properties
Category 4: Change Risk Analysis Rule	Q30 Rpc Must Have Timeout Query	Java, Xml
	Q31 Find Set Before Update Query	Java
	Q32 Find Local Thread Pool Query	Java
	Q33 Find Inherited Class with the Same Name Query	Java
	Q34 Find Reference Assignment Query	Java, Xml
	Q35 Find Authenticate User Info Query	Java
	Q36 Find Cache Expiration Time Query	Java
	Q37 Find Interface Field Assigned Query	Java, Xml
	Q38 Find Jar Method Usage Query	Java
Category 5: Privacy Governance and Legal Compliance	Q39 Find Privacy Field In Interface Query	Java
	Q40 Find Exported Privacy Message Info Query	Java
	Q41 Find Depended Privacy Interface Query	Java
	Q42 Find Recommendation Algorithm Setting Query	Java, Xml
	Q43 Find Exported Privacy DB Info Query	Java, Xml
	Q44 Find Privacy DB Fields Lineage Query	SQL
	Q45 Find Privacy Data Lineage from Code to DB Field Query	Java, Xml, SQL
Category 6: Insurance Quality Testing	Q46 Find Adapter Setting Query	Java, Xml
	Q47 Find Insurance CV Model Mapping Query	Java, Xml, Python
	Q48 Find Configure Key Value Query	Java
Category 6: Security and AOP Governance	Q49 Find All Point Cut Values Query	Java
	Q50 Find All Point Cut Value Influences Query	Java
	Q51 Find Released Artifact Module Query	Java, Xml

Category	Script Name	Programming Languages
Category 7: Mini Program Security and Risk Governance	Q52 Find Chair Framework Api Query	Js/Ts
	Q53 Find Trade Bff Rpc Field Tracing Query	Js/Ts, Xml, Java
	Q54 Find Loop Pop Up Confrim Query	Js/Ts
	Q55 Find Loop Pop Up Onload Query	Js/Ts
	Q56 Find Loop Pop Up Redirect Query	Js/Ts
	Q57 Find Over Collection User Info Query	Js/Ts
Category 8: Android/IOS Package Size Governance	Q58 Find Unused Interface Query	Objective-C
	Q59 Find Class Dependency Query	Objective-C
	Q60 Find Resource Setting in Xml	Java, Xml
	Q61 Find Function/Class Declaration	Swift
	Q62 Find All statements and Ancestor Query	Objective-C
	Q63 Find All Declaration and Ancestor Query	Objective-C, Swift
Category 9: Middleware Governance	Q64 Check Value Type Write Query	Go
	Q65 Check Http Body Close Query	Go
	Q66 Check Unused Function Query	Go
	Q67 Check Error Setting Query	Go
	Q68 Check Set User Agent Definition Query	Go
	Q69 Check Set K8s User Structs Query	Go
	Q70 Find Control/Webhook Watches Query	Go
Category 10: Data Preprocessing for LLM Traning	Q71 Valid Function Comment Pair Query	Go
	Q72 Valid Function Comment Pair Query	Python
	Q73 Valid Callable/Class Documentation Pair Query	Java
	Q74 Valid Code Documentation Pair Query	Js/Ts
	Q75 Filter Oversized Code Block Query	Js/Ts, Java, Python, Go
	Q76 Filter Over Complicated Code Block Query	Js/Ts, Java, Python, Go
	Q77 Filter Auto-generated Files Query	Js/Ts, Java, Python, Go

Table 10: List of query scripts currently in use.

C Open-source Repositories URLs in Evaluation

No.	Category	Repository Name	Repository URL
1	Python	Poetry	https://github.com/python-poetry/poetry.git
2	Python	Pytest	https://github.com/pytest-dev/pytest.git
3	Python	Faust	https://github.com/robinhood/faust.git
4	Python	Cirq	https://github.com/quantumlib/Cirq.git
5	Python	Request-HTML	https://github.com/psf/requests-html.git
6	Python	Bokeh	https://github.com/bokeh/bokeh.git
7	Python	Molten	https://github.com/Bogdanp/molten.git
8	Python	TermGraph	https://github.com/mkaz/termgraph.git
9	Python	Black	https://github.com/psf/black.git
10	Python	Bowler	https://github.com/facebookincubator/Bowler.git
11	Python	Transcrypt	https://github.com/TranscryptOrg/Transcrypt.git
12	Python	Langchain	https://github.com/langchain-ai/langchain.git
13	Python	AutoGPT	https://github.com/Significant-Gravitas/AutoGPT.git
14	Python	Flask	https://github.com/pallets/flask.git
15	Python	Chartify	https://github.com/spotify/chartify.git
16	Java (FRA)	Zipkin	https://github.com/openzipkin/zipkin
17	Java (FRA)	IoTDB	https://github.com/apache/iotdb
18	Java (FRA)	Dubbo	https://github.com/apache/dubbo
19	Java (FRA)	Kafka	https://github.com/apache/kafka.git
20	Java (FRA)	Camel	https://github.com/apache/camel.git
21	Java (FRA)	SkyWalking	https://github.com/apache/skywalking.git
22	Java (FRA)	RocketMQ	https://github.com/apache/rocketmq.git
23	Java (FRA)	Pulsar	https://github.com/apache/pulsar.git
24	Java (FRA)	HBase	https://github.com/apache/hbase.git
25	Java (FRA)	Hive	https://github.com/apache/hive.git
26	Java (FRA)	Storm	https://github.com/apache/storm.git
27	Java (FRA)	Iceberg	https://github.com/apache/iceberg.git
28	Java (FRA)	Logging-log4j2	https://github.com/apache/logging-log4j2
29	Java (DCA)	Hadoop	https://github.com/apache/hadoop
30	Java (DCA)	Druid	https://github.com/apache/druid
31	Java (DCA)	CAT	https://github.com/dianping/cat
32	Java (DCA)	Deeplearning4j	https://github.com/deeplearning4j/deeplearning4j
33	Java (DCA)	Realm-Java	https://github.com/realm/realm-java
34	Java (DCA)	Material Components Android	https://github.com/material-components/material-components-android
35	Java (DCA)	DoKit	https://github.com/didi/DoKit
36	Java (DCA)	Jedis	https://github.com/redis/jedis
37	Java (DCA)	Flink	https://github.com/apache/flink
38	Java (DCA)	Hystrix	https://github.com/Netflix/Hystrix
39	Java (DCA)	Apollo	https://github.com/apolloconfig/apollo
40	Java (DCA)	Tinker	https://github.com/Tencent/tinker
41	Java (DCA)	PhotoView	https://github.com/Baseflow/PhotoView
42	Java (DCA)	Fastjson	https://github.com/alibaba/fastjson
43	Java (DCA)	Servo	https://github.com/Netflix/servo
44	Java (DCA)	Eureka	https://github.com/Netflix/eureka
45	Java (DCA)	RxJava	https://github.com/ReactiveX/RxJava
46	Java (DCA)	Copybara	https://github.com/google/copybara
47	Java (DCA)	Guice	https://github.com/google/guice
48	Java (DCA)	Gson	https://github.com/google/gson
49	Java (DCA)	Guava	https://github.com/google/guava
50	Java (DCA)	Redisson	https://github.com/redisson/redisson

D Representative ER/Class diagram

Figure 4: COREF for Java ER Diagram

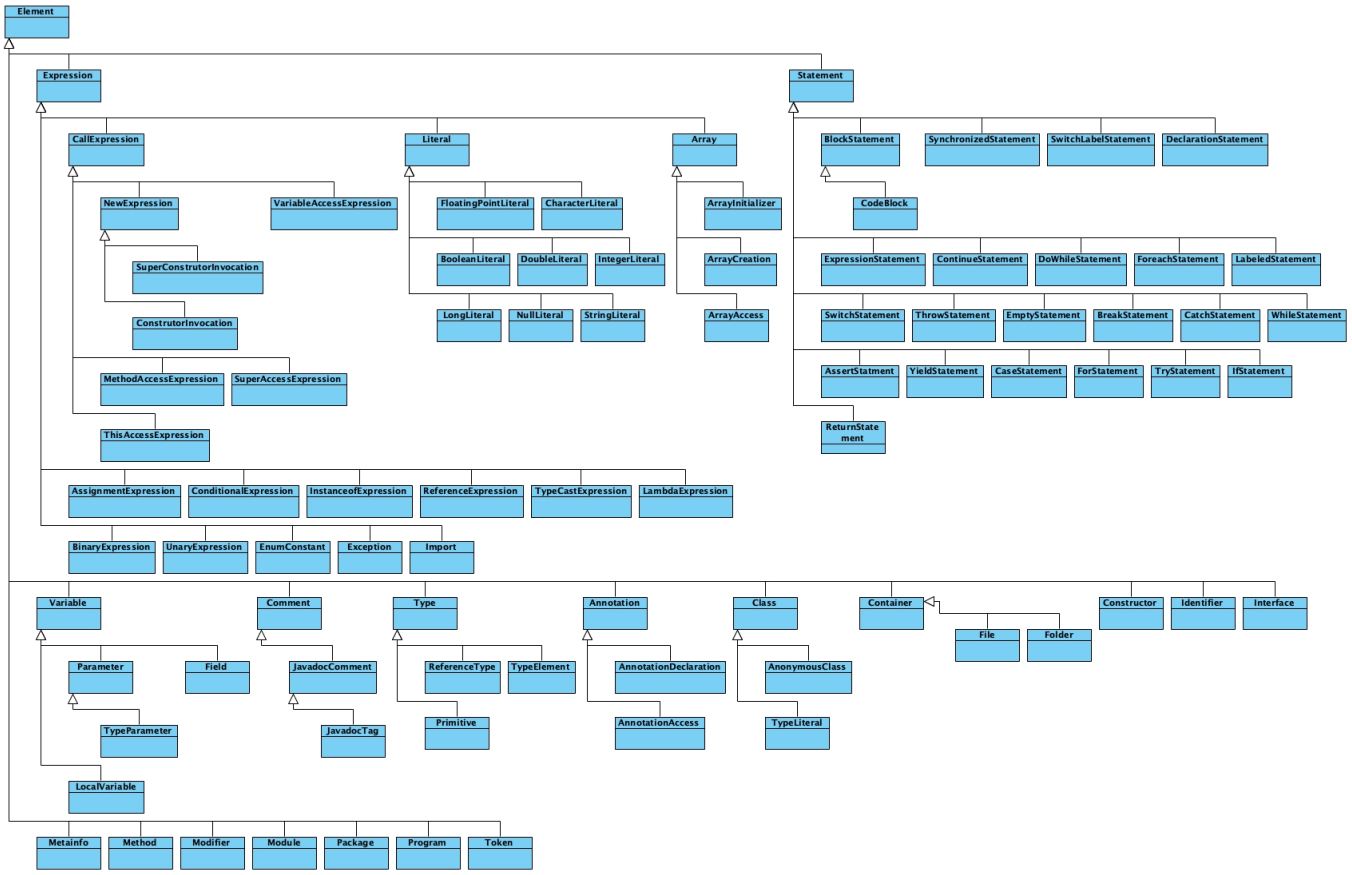

Figure 5: COREF for Java Class Diagram

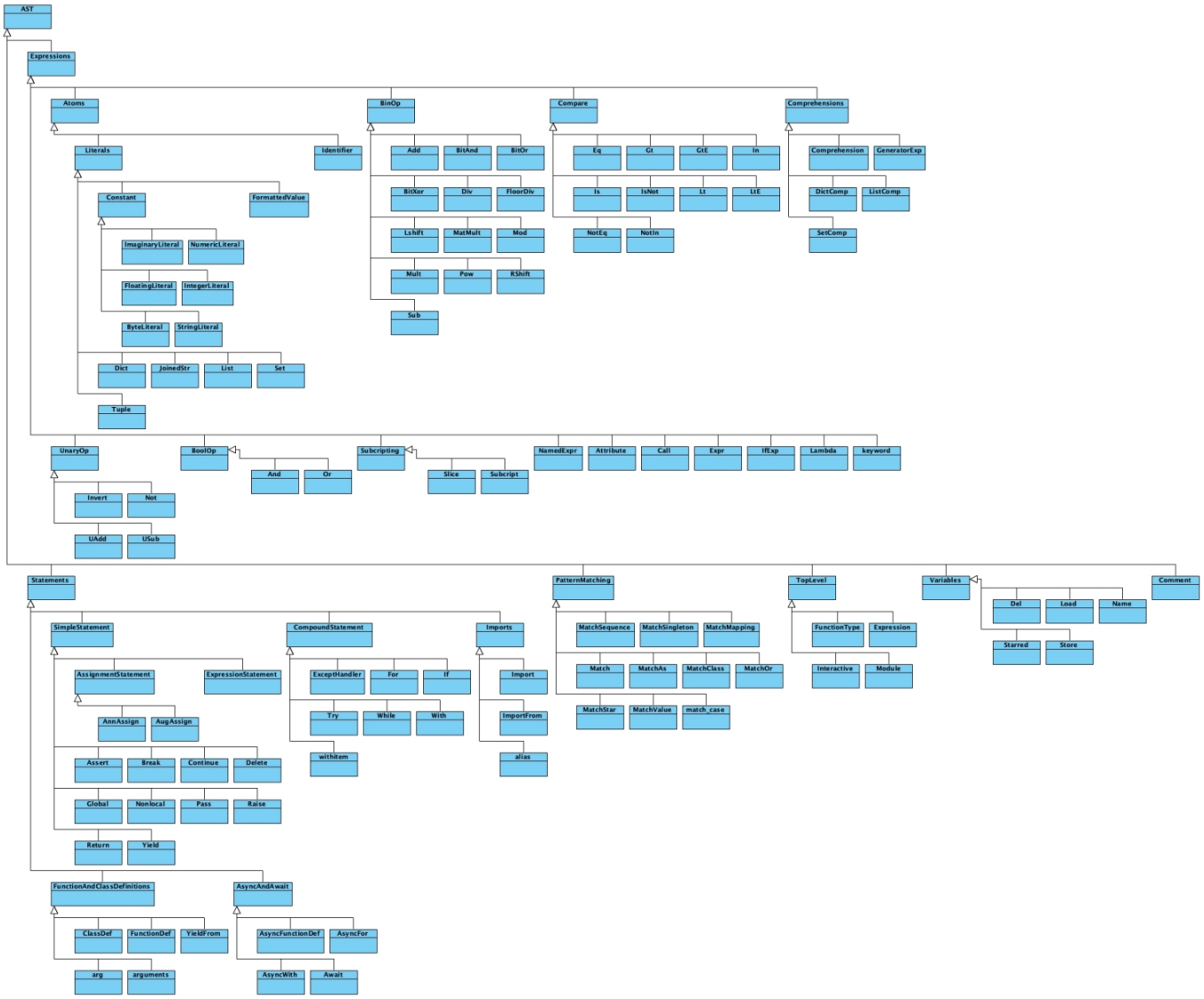

Figure 6: COREF for Python Class Diagram

E Statistics of Distinct Queries for Whole Version Tasks Over One Week.

Code Version	Day 1	Day 2	Day 3	Day 4	Day 5
Total Query Tasks	119510	126231	132537	133344	110990
Distinct Queries	1751	2630	2641	2694	2558
Distinct Queries (Normalized)	111	98	101	105	100

Table 11: Statistics of Distinct Queries for Whole Version Tasks Over One Week.

Script ID	Frequency
Q1	24293
Q2	22694
Q3	22631
Q4	1813
Q5	1802
Q6	343
Q7	38
Q8	35
Q9	35
Q10	22

(a) Most frequent queries for slice tasks

Script ID	Frequency
Q1	5393
Q2	5390
Q3	5389
Q4	5388
Q5	5387
Q6	5387
Q7	5387
Q8	5385
Q9	5381
Q10	5381

(b) Most frequent queries for whole version tasks

Script ID	Frequency
Q1	46
Q2	45
Q3	45
Q4	31
Q5	31
Q6	31
Q7	19
Q8	19
Q9	19
Q10	18

(c) Most frequent queries for slice tasks with the same code

Script ID	Frequency
Q1	54
Q2	38
Q3	35
Q4	35
Q5	35
Q6	34
Q7	33
Q8	33
Q9	33
Q10	33

(d) Most frequent queries for whole version tasks with the same code

Table 12: Top 10 most frequent queries per day for four scenarios.

F Example Query Scripts

Listing 2: Query Example 1

```
// script
use coref::java::*

fn default_java_db() -> JavaDB {
    return JavaDB::load("coref_java_src.db")
}

// find unused methods
fn unused_method(unused: string) -> bool {
    for(c in Callable(default_java_db()), method in Callable(default_java_db()), caller in
        method.getCaller()) {
        if (c != caller && unused = method.getSignature()) {
            return true
        }
    }
}

fn main() {
    output(unused_method())
}
```

Listing 3: Query Example 2

```
// script
use coref::javascript::*

fn default_db() -> JavascriptDB {
    return JavascriptDB::load("coref_javascript_src.db")
}

fn getACallerFunction(function: FunctionLikeDeclaration, callerFunction: FunctionLikeDecla
    ration) -> bool {
    for (mayInvokeExpression in MayInvokeExpression(default_db())) {
        if (mayInvokeExpression in function.getACallSite() &&
            callerFunction = mayInvokeExpression.getEnclosingFunction()) {
            return true
        }
    }
}

fn getAnEffectedFunction(function: FunctionLikeDeclaration, effectedFunction:
    FunctionLikeDeclaration) -> bool {
    if (getACallerFunction(function, effectedFunction)) {
        return true
    }
    for (callerFunction in FunctionLikeDeclaration(default_db())) {
        if (getACallerFunction(function, callerFunction) &&
            getAnEffectedFunction(callerFunction, effectedFunction)) {
            return true
        }
    }
}

/**
```

```
 * Query the effected functions according to the changed lines.
 *
 * @param function              the changed function id
 * @param signature             the changed function signature
 * @param functionPath          the changed function file path
 * @param startLine             the changed function start line
 * @param endLine               the changed function end line
 * @param effectedFunction      the effected function id
 * @param effectedSignature     the effected function signature
 * @param effectedFunctionPath  the effected function file path
 * @param effectedStartLine     the effected function start line
 * @param effectedEndLine       the effected function end line
 */
fn out(
    function: FunctionLikeDeclaration,
    signature: string,
    functionPath: string,
    startLine: int,
    endLine: int,
    effectedFunction: FunctionLikeDeclaration,
    effectedSignature: string,
    effectedFunctionPath: string,
    effectedStartLine: int,
    effectedEndLine: int
) -> bool {
    if (getAnEffectedFunction(function, effectedFunction)) {
        let (symbol = function.getSymbol(),
            effectedSymbol = effectedFunction.getSymbol(),
            location = function.getLocation(),
            effectedLocation = effectedFunction.getLocation()) {
            if (signature = symbol.getDescription() &&
                effectedSignature = effectedSymbol.getDescription() &&
                functionPath = location.getRelativePath() &&
                startLine = location.getStartLineNumber() &&
                endLine = location.getEndLineNumber() &&
                effectedFunctionPath = effectedLocation.getRelativePath() &&
                effectedStartLine = effectedLocation.getStartLineNumber() &&
                effectedEndLine = effectedLocation.getEndLineNumber()) {
                return true
            }
        }
    }
}

fn main() {
    output(out())
}
```

Listing 4: Query Example 3

```
// script
use coref::xml::*

schema DependencyElement extends XmlElement {}

impl DependencyElement {
    @data_constraint
    pub fn __all__(db: XmlDB) -> *DependencyElement {
        for(e in XmlElement(db)) {
            if (e.getElementName() = "dependency") {
                yield DependencyElement {
                    id: e.id,
                    location_id: e.location_id,
                    parent_id: e.parent_id,
                    index_order: e.index_order
                }
            }
        }
    }
}

schema GroupElement extends XmlElement {}

impl GroupElement {
    @data_constraint
    pub fn __all__(db: XmlDB) -> *GroupElement {
        for(e in XmlElement(db)) {
            if (e.getElementName() = "groupId") {
                yield GroupElement {
                    id: e.id,
                    location_id: e.location_id,
                    parent_id: e.parent_id,
                    index_order: e.index_order
                }
            }
        }
    }
}

schema VersionElement extends XmlElement {}

impl VersionElement {
    @data_constraint
    pub fn __all__(db: XmlDB) -> *VersionElement {
        for(e in XmlElement(db)) {
            if (e.getElementName() = "version") {
                yield VersionElement {
                    id: e.id,
                    location_id: e.location_id,
                    parent_id: e.parent_id,
                    index_order: e.index_order
                }
            }
        }
    }
```

```
}

schema ArtifactElement extends XmlElement {}

impl ArtifactElement {
    @data_constraint
    pub fn __all__(db: XmlDB) -> *ArtifactElement {
        for(e in XmlElement(db)) {
            if (e.getElementName() = "artifactId") {
                yield ArtifactElement {
                    id: e.id,
                    location_id: e.location_id,
                    parent_id: e.parent_id,
                    index_order: e.index_order
                }
            }
        }
    }
}

schema PomFile extends XmlFile {}

impl PomFile {
    @data_constraint
    pub fn __all__(db: XmlDB) -> *PomFile {
        for(f in XmlFile(db)) {
            if (f.getFileName() = "pom.xml") {
                yield PomFile {
                    id: f.id,
                    file_name: f.file_name,
                    relative_path: f.relative_path
                }
            }
        }
    }
}

// output relative path of the file, referenced jar name and version
fn out(fileName: string, m1: string, m2: string, m3: string) -> bool {
    let (db = XmlDB::load("coref_xml_src.db")) {
        for (f in PomFile(db),
            e1 in GroupElement(db),
            e2 in VersionElement(db),
            e3 in ArtifactElement(db),
            c1 in XmlCharacter(db),
            c2 in XmlCharacter(db),
            c3 in XmlCharacter(db),
            p in DependencyElement(db)) {
            if (f.key_eq(p.getLocation().getFile()) &&
                fileName = f.getRelativePath() &&
                p.key_eq(e1.getParent()) &&
                e1.key_eq(c1.getBelongedElement()) &&
                m1 = c1.getText() &&
                p.key_eq(e2.getParent()) &&
                e2.key_eq(c2.getBelongedElement()) &&
                m2 = c2.getText() &&
                p.key_eq(e3.getParent()) &&
```

```
                e3.key_eq(c3.getBelongedElement()) &&
                m3 = c3.getText()) {
                return true
            }
        }
    }
}

fn main() {
    output(out())
}
```

G Comparative Results of Querying Performance

Language	Query Name	CoQuery		CodeQL	
		Time(s)	Mem(MB)	Time(s)	Mem(MB)
Java	Q1. Afferent Coupling	18.5	294.7	7.2	1018.1
Java	Q2. Efferent Coupling	19.7	292.3	71.5	3045.9
Java	Q3. Cyclomatic Complexity	64.4	1391.6	6.5	1003.8
Java	Q4. Call Graph	19.8	297.3	5.8	785.9
Java	Q5. Class Hierarchy	7.0	151.1	4.9	674.2
Java	Q6. Find All Class	12.2	328.7	4.5	718.8
	Avg.	23.6	459.3	16.7	1207.8
Python	Q1. Cyclomatic Complexity	11.9	169.5	27.6	2431.3
Python	Q2. Class Hierarchy	9.6	178.7	17.1	1964.3
Python	Q3. Find Redundant If Statement	6.1	145.4	2.7	382.9
	Avg.	9.2	164.5	15.8	1592.8

Table 13: Comparative Results of Querying Performance

H Code Model Statistics

Language	**Status**	**Nodes (T1)**	**Nodes (T2)**
Java	Mature	157	482
XML	Mature	12	27
Js/Ts	Mature	392	574
Objective-C	Beta	53	109
Go	Beta	38	263
Python	Beta	55	120
Swift	Beta	248	679
SQL	Beta	750	2552
Properties	Beta	9	11

Table 14: Code Model Statistics