

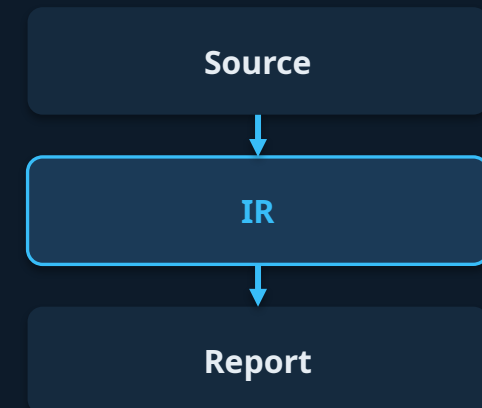
AuxFlow

Static Analysis for Multi-Pass Document Compilation

A compiler-based approach for analyzing dependency-driven compilation behaviour.

Name Upayan Mazumder
Reg. No. 24BDS0367
Course Compiler Design Lab

dependency graph



PROBLEM

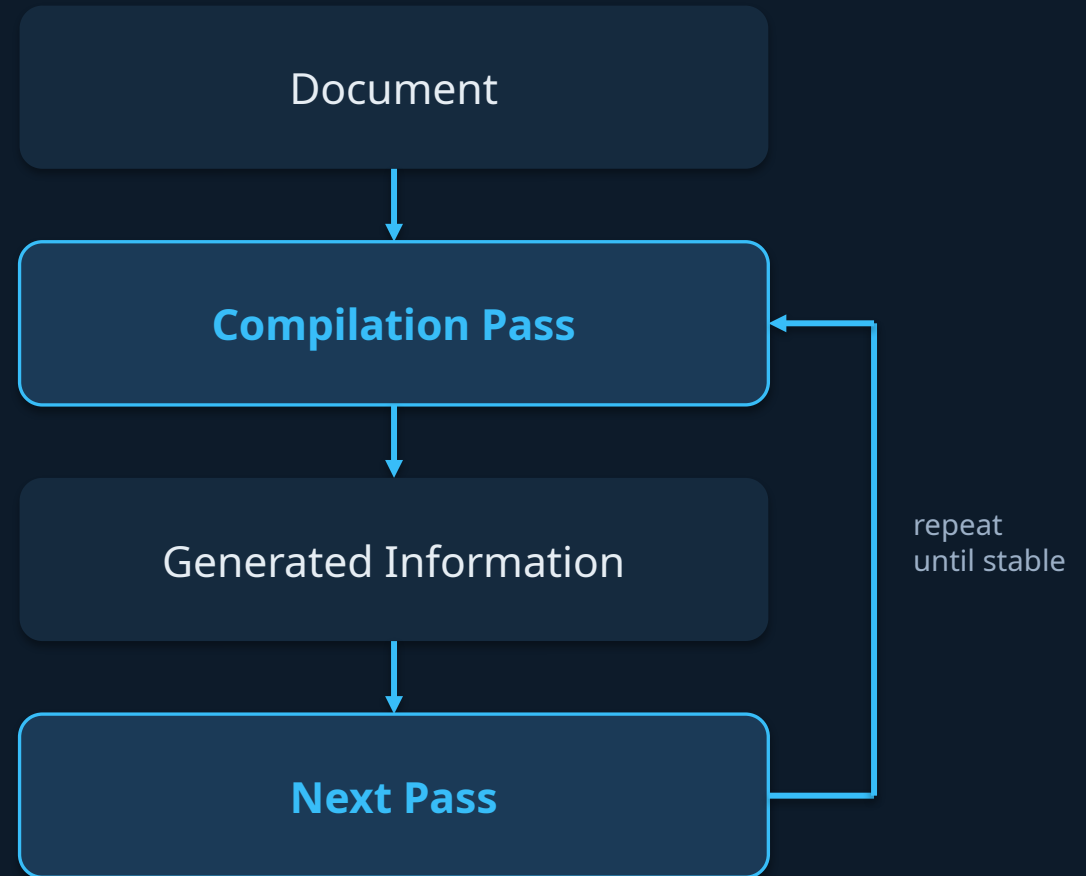
Why documents need multiple passes

The behaviour

Documents can compile in **several passes**.

Each pass writes information that later passes read back.

So passes repeat until the state stops changing — and this behaviour is hard to see from the source.



What the project sets out to do

Objectives

- ✓ Build a compiler pipeline for a restricted document subset.
- ✓ Extract meaningful cross-reference dependencies.
- ✓ Represent them as an intermediate dependency graph.
- ✓ Perform static analysis over that representation.
- ✓ Generate understandable, source-located reports.

Scope

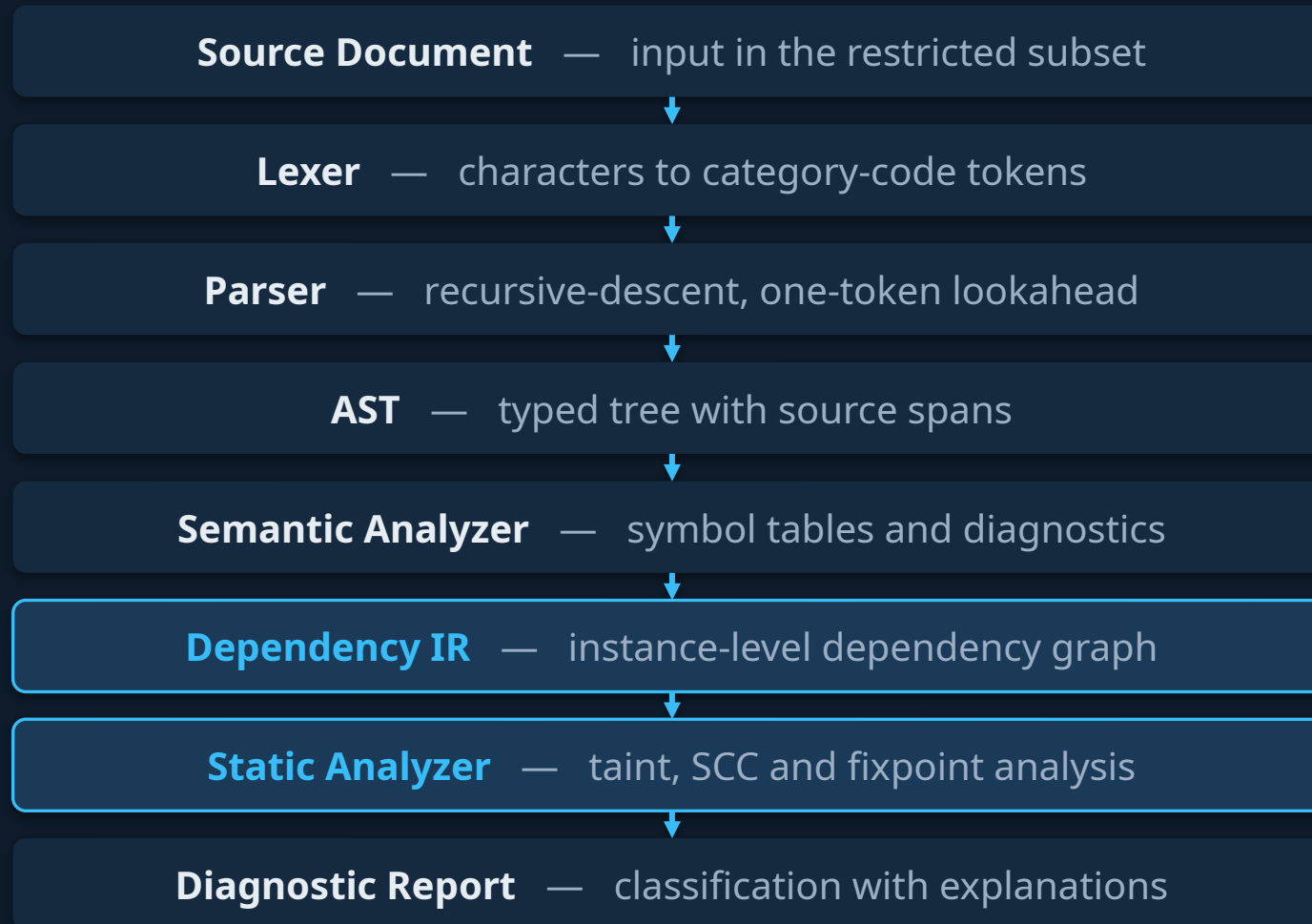
- Restricted set of document features.
- Dependency-focused analysis only.
- Conservative results — flag or defer, never guess.

Anything outside the subset is marked **opaque** and degrades safely.

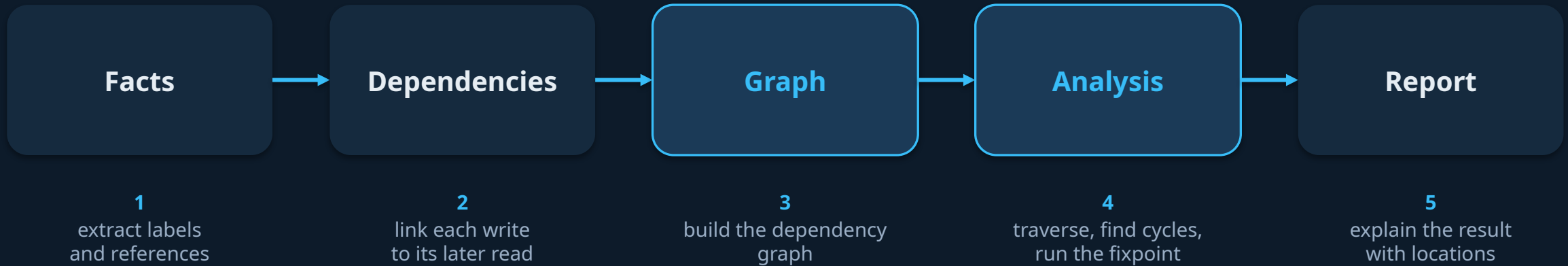
Every stage is a standard compiler concept

Compiler Phase	AuxFlow Component
Lexical Analysis	Token extraction (category-code scanner)
Syntax Analysis	AST generation (recursive-descent parser)
Semantic Analysis	Dependency extraction (symbol tables)
Intermediate Representation	Dependency graph (Auxiliary Reference Graph)
Static Analysis	Graph evaluation (dataflow / fixpoint)
Graph Algorithms	Cycle detection (Tarjan SCC)

A single-direction compiler pipeline



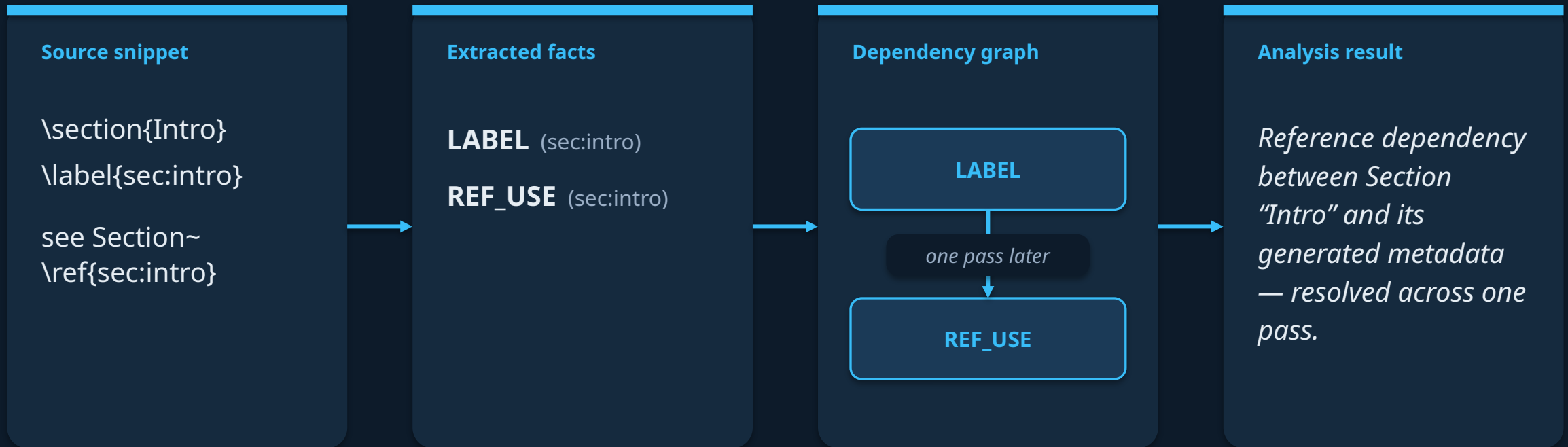
From source facts to an explanation



Techniques: graph traversal · cycle detection (strongly connected components) · dataflow / fixpoint analysis

EXAMPLE

A single cross-reference, end to end



A compiler project with an analysis target

01

Compiler concepts applied

Lexer, parser, AST, symbol tables, an IR, and dataflow and graph analysis — a complete front-end.

02

Dependency-based static analysis

Cross-reference dependencies modelled as a graph and analysed statically, without running any engine.

03

Explainable, conservative results

A readable, source-located report that flags risky feedback and defers safely when unsure.

Source

Front-End

IR

Analysis

Report