Seal5 stage sequence

Fresh run with PatternGen; follow the numbered boxes left to right, then the next row.

Preparation

01 reset (dashed)
fresh-run reset

→ **02 initialize**
LLVM checkout
+ metadata

→ **03 load**
CoreDSL, YAML
and tests

→ **04 setup**
dependencies
+ support patches

→ **05 patch**
phase 0
bootstrap support

→ **06 build**
bootstrap LLVM

Construction

07 transform
model passes

→ **08 generate**
structural artifacts

→ **09 patch**
phases 1–2

→ **10 build**
instruction-aware LLVM
PatternGen + llc

→ **11 generate**
selection patterns

→ **12 patch**
phases 3–5

→ **13 build**
final LLVM

Validation

14 test
imported + generated
lit tests

Delivery

15 install (dashed)
compiler installation

→ **16 deploy** (dashed)
LLVM source archive

→ **17 export** (dashed)
metadata + patches

→ **18 clean** (dashed)
final cleanup

Dashed boxes: optional reset / delivery. Other operations can also be disabled through lifecycle flags.

The construction build groups the instruction-aware build and extraction-tool targets; see the full-flow figure for details.

Source: seal5/wrapper.py · seal5/flow.py. LLVM patch-phase definitions are provided in the separate TeX table.