PR #1890 完整报告

radixark/miles

Update the mechanical-refactor-verify skill to the latest sglang version

合并时间：2026-08-26 07:21

原文链接：http://prhub.com.cn/radixark/miles/pull/1890

执行摘要

- 一句话：升级 refactor 验证技能，新增 proof 生成与链式验证工具链
- 推荐动作：值得精读。核心看点是 mechanical_refactor_reproduction_utils.py 中 AST 定位 + 源码文本拼接的原子原语设计，以及 mechanical_refactor_proof_generator.py 中从 diff 反推 recipe 的启发式推断；这两者组合实现了「机器证明机械重构」的闭环。对计划做大规模代码移动或希望自动化重构 review 的团队有直接借鉴价值。建议重点关注 verify_mechanical_refactor 的 worktree 校验流程和 infer_recipe 对不支持场景（rename、语句级重排）的显式拒绝策略。

功能与动机

该 PR 是 #1837（refactoring and enhancements 跟踪 issue）的一部分，目标是让机械重构可被机器验证。PR 标题明确说明要同步最新 sglang 版本的 skill，commit message 也注明是 copied skill files 并经过 pre-commit 适配。升级后，mechanical-refactor-verify skill 不再只是简单封装 git 命令，而是能从 commit diff 推断 reproduce recipe、生成独立证明脚本、并验证整条机械重构链，从而支撑 #1837 下系列 launch scripts 重构 PR 的自动化审核。

实现拆解

整个升级分为四步：

1. 重构复现工具层：新增scripts/mechanical_refactor_reproduction_utils.py（约1359行），在旧 verify_mechanical_refactor 基础上提供 Repro 构建器，组合 move_symbol、extract_to_new_module、extract_function、lower_call_sites、requalify_call_sites、add_import、remove_imported_name 等 AST 定位、源码文本拼接的原子原语；并改进 exec_command 失败路径为抛 RuntimeError 而非直接 sys.exit，便于上层捕获。
2. 新增证明生成器：新增 scripts/mechanical_refactor_proof_generator.py（约 1401 行），通过 _per_file_diff 解析 commit 的逐文件增删行，用 _enclosing_function、_enclosing_class_of_def 等辅助函数在 base 状态 AST 上推断移动、调用点降级、导入重定向，生成 repro_scripts/<sha>.py 独立脚本并执行，diff 为空才 PASS。
3. 新增链式验证 CLI：新增 scripts/mechanical_refactor_reproduction_cli.py（约 484 行），提供 verify_chain 与 CommitVerdict、ChainResult 等数据模型，按 mechanical_provable / non_mechanical_provable 关键字分类链上每个 commit，并行运行 proof 并用 --skip-passed 缓存结果，输出 chain_report.md，exit code 反映整链

verdict。

4. 删除旧入口并补齐测试：删除旧的 mechanical_refactor_verify_utils.py（80 行），其能力由新 scripts 目录取代；同时新增 scripts/tests/ 下 31 个测试文件，覆盖 proof generator 的 recipe 推断、reproduction utils 的符号移动与导入原语、CLI 的链验证语义，测试通过 sys.path.insert 直接引用被测模块，未接入 CI 而由开发者手工运行。

关键文件：

- .claude/skills/mechanical-refactor-verify/scripts/mechanical_refactor_reproduction_utils.py（模块 复现工具；类别 source；类型 dependency-wiring；符号 exec_command, git_add_and_commit, dedent, _split_keepends）：新增约 1359 行，是整个工具链的核心：提供 AST 定位的原子移动原语（move_symbol、extract_to_new_module、extract_function 等）和 verify_mechanical_refactor 的 worktree 逐字节校验流程。
- .claude/skills/mechanical-refactor-verify/scripts/mechanical_refactor_proof_generator.py（模块 证明生成；类别 source；类型 dependency-wiring；符号 _git_output, _repo_root, _removed_symbol_names, _def_indent）：新增约 1401 行，负责从 commit diff 与 base 状态 AST 推断 reproduce recipe，并生成独立可运行的证明脚本，是 skill 从「手动写 transform」到「自动生成证明」的关键升级。
- .claude/skills/mechanical-refactor-verify/scripts/mechanical_refactor_reproduction_cli.py（模块 链式验证；类别 source；类型 dependency-wiring；符号 ChainVerificationError, CommitVerdict, ok, _PendingProof）：新增约 484 行，提供整条机械重构链的验证入口：按关键字分类 commit、并发运行 proof、输出 markdown 报告并映射 exit code。
- .claude/skills/mechanical-refactor-verify/mechanical_refactor_verify_utils.py（模块 旧版工具；类别 source；类型 deletion；符号 exec_command, git_add_and_commit, dedent, verify_mechanical_refactor）：删除的旧单文件工具模块（80 行），其能力被 scripts/ 目录下的新实现取代；删除动作需要确认仓库内无残留引用。
- .claude/skills/mechanical-refactor-verify/scripts/tests/proof_generator/test_infer_moves.py（模块 证明生成；类别 test；类型 test-coverage；符号 test_infer_recipe_method_onto_class, test_infer_recipe_move_before_typechecking_uses_after_anchor, test_infer_recipe_free_function_move_uses_requalify, test_infer_recipe_excludes_the_moved_bodys_own_call）：代表性测试文件（525 行），覆盖 recipe 推断的核心场景：方法移入类、移动到模块级自由函数、TYPE_CHECKING 锚点选择、非 Python 文件存在等。
- .claude/skills/mechanical-refactor-verify/scripts/tests/reproduction_utils/test_move_symbol.py（模块 符号移动；类别 test；类型 test-coverage；符号 test_move_symbol_drops_self_annotation_into_class, test_move_symbol_into_class_drops_decorator_and_appends, test_move_symbol_to_module_level_with_dedent, test_move_symbol_before_inserts_above_named_sibling）：代表性测试文件（337 行），验证 move_symbol 原语在移入类、移到模块级、保留装饰器、async 函数、同文件移动等场景下的忠实性。

关键符号：infer_recipe, recipe_to_script, generate_range, verify_mechanical_refactor, verify_chain, move_symbol, extract_to_new_module, extract_function, lower_call_sites, requalify_call_sites, exec_command, main

关键源码片段

.claude/skills/mechanical-refactor-verify/scripts/mechanical_refactor_reproduction_utils.py

新增约 1359 行，是整个工具链的核心：提供 AST 定位的原子移动原语（move_symbol、extract_to_new_module、extract_function 等）和 verify_mechanical_refactor 的 worktree 逐字节校验流程。

```
def verify_mechanical_refactor(
    base_commit: str,
    target_commit: str,
    transform: "Callable[[Path], None]",
) -> None:
    """在一次性 worktree 中复现机械重构，并与目标 commit 做逐字节比对。

    流程：基于 base_commit 创建临时 worktree -> 运行 transform 还原变更 ->
    对变更文件跑 pre-commit -> 与 target_commit 做字节级 diff，diff 为空才算 PASS。
    """
    repo_root = exec_command("git rev-parse --show-toplevel")
    worktree_dir = tempfile.mkdtemp(prefix="verify-mechanical-")
    branch_name = f"verify-mechanical-{base_commit[:8]}"

    try:
        print(f"[1/4] Creating worktree at {base_commit[:8]}...")
        # 从 base_commit 拉出独立 worktree，避免污染当前开发环境
        exec_command(
            f"git worktree add -b {branch_name} {worktree_dir} {base_commit}",
            cwd=repo_root,
        )

        print("[2/4] Running transformation...")
        # transform 由调用方提供，内部组合各个原子移动原语
        transform(Path(worktree_dir))

        print("[3/4] Running pre-commit...")
        # 只对变更文件跑 pre-commit，避免全仓格式化引入无关噪音
        exec_command("git add -A", cwd=worktree_dir)
        changed = exec_command(
            f"git diff --cached --name-only --diff-filter=ACMR {base_commit}",
            cwd=worktree_dir,
        ).split()
        if changed:
            files = " ".join(shlex.quote(path) for path in changed)
            exec_command(f"pre-commit run --files {files}", cwd=worktree_dir, check=False)
```

```
        if exec_command("git status --porcelain", cwd=worktree_dir):
            git_add_and_commit("pre-commit fixes", cwd=worktree_dir)

        print(f"[4/4] Diffing against {target_commit[:8]}...")
        # 字节级 diff：任何差异都说明 transform 没有忠实复现原 commit
        diff = exec_command(
            f"git diff {target_commit} -- .",
            cwd=worktree_dir,
            check=False,
        )

        if diff:
            print(f"\nFAIL: diff is non-empty:\n{diff}")
            sys.exit(1)
        print("\nPASS: transform reproduces the commit exactly.")

    finally:
        # 故意保留 worktree 供人工复核，并打印清理命令
        print(f"\nWorktree left at: {worktree_dir}")
        print(f"Branch: {branch_name}")
        print("To clean up manually:")
        print(f"  git worktree remove {worktree_dir} && git branch -D {branch_name}")
```

.claude/skills/mechanical-refactor-verify/scripts/mechanical_refactor_proof_generator.py

新增约 1401 行，负责从 commit diff 与 base 状态 AST 推断 reproduce recipe，并生成独立可运行的证明脚本，是 skill 从「手动写 transform」到「自动生成证明」的关键升级。

```
def _per_file_diff(commit: str, root: str) -> dict[str, dict]:
    """逐文件解析 commit 的删除/新增内容行（保留空白），并标记是否新文件。

    返回结构供后续 AST 推断使用：removed 与 added 都按行保留原样，
    这样 ast 的行号能和文件真实行号对齐，保证定位准确。
    """
    out = _git_output(["show", commit, "--format=", "--no-color", "--no-ext-diff"], root)
    files: dict[str, dict] = {}
    path: str | None = None
    in_hunk = False
    for line in out.splitlines():
        header = re.match(r"diff --git a/(.*) b/(.+)$", line)
        if header:
            # 每个文件的 diff 起始行，初始化独立的增删缓冲区
            path = header.group(2)
            files[path] = {"removed": [], "added": [], "new": False, "deleted": False}
            in_hunk = False
        elif line.startswith("new file"):
            files[path]["new"] = True
        elif line.startswith("deleted file"):
            files[path]["deleted"] = True
```

```
        elif line.startswith("@@"):
            # 进入 hunk 后才开始收集 + / - 行，diff 头部的元信息不参与
            in_hunk = True
        elif in_hunk and line.startswith("+"):
            files[path]["added"].append(line[1:])
        elif in_hunk and line.startswith("-"):
            files[path]["removed"].append(line[1:])
    return files
```

评论区精华

PR 无实质性技术争论，review_comments 为 0。唯一一条审核评论来自 guapisolo 的 APPROVED：

> treasure, better merge to main first as standalone PR

即认可该变更价值，并建议作为独立 PR 先合入 main，而不是积压在功能分支里。该建议已被合并者采纳。Issue 侧仅有一条 gemini-code-assist 的弃用声明，与本次变更无关。

- 独立 PR 合并建议 (other): 合并者采纳建议，将该 skill 升级作为独立 PR 合入 main。

风险与影响

- 风险：
  1. 大体积外部导入：约 9000 行新增且来源为 sglang 仓库，虽然 commit message 提到已跑 pre-commit 自动修复，但跨仓库复制的代码可能与本仓库 AST 版本、pre-commit hook 集合存在隐性差异，需关注后续运行反馈。
  2. 删除旧模块的兼容性风险：mechanical_refactor_verify_utils.py 被整体删除，旧 skill 使用者或外部脚本若仍 import 该模块会直接失败，需确认仓库内无残留引用。
  3. shell 执行面：exec_command 使用 shell=True 拼接命令字符串，参数来自 skill 内部受控代码，但若未来允许用户传入任意 commit/ 分支名，存在注入面，建议保持参数白名单。
  4. 测试运行方式不明确：31 个测试文件挂在 .claude/skills 下，未在现有 CI 工作流中发现对应接入点，存在测试被遗漏执行的风险。 - 影响：影响集中在开发工具链而非运行时产品：skill 升级后，Claude Code 等 AI 助手在仓库内执行机械重构时，可获得 recipe 推断、逐字节复现和整链验证能力，显著降低 #1837 系列重构 PR 的审核负担。对团队而言，这是一套可复用的「重构可证明」方法论沉淀；对用户和系统无运行时影响。由于是独立 skill 目录，风险隔离良好，但后续所有机械重构 PR 都可能依赖该工具链的可靠性。 - 风险标记：大体积外部导入代码，旧模块删除兼容性，shell 命令拼接执行，测试接入 CI 待确认

关联脉络

- PR #1896 Add a shell launch script test harness for future protection: 同属 #1837 重构跟踪下的子任务，测试基座与本 PR 的证明工具链共同构成重构可信度保障。

- PR #1901 Snapshot the commands and generated configs of every python launch script: 同一跟踪 issue 下的快照测试体系，与本 PR 的逐字节复现证明互为补充。
- PR #2279 Run the launch script snapshot tests by hand instead of in CI: 明确 snapshot 测试改人工运行，与本 PR 中 skill 测试未接入 CI 的运行方式一致，反映同一演进方向。