Subgraph Isomorphism Refactoring Analysis

Antigravity Agent

January 29, 2026

1 Methodology

We performed a structural analysis of the `spip.cpp` codebase to identify repeated code patterns (code clones) using a subgraph isomorphism approach. We employed:

1. **Gradient Descent Isomorphism Solver**: To detecting variations of the same structure.
2. **Constraint-Based Matching**: Using basic block histograms and sequence alignment.

2 Findings

The analysis identified a massive redundancy regarding the resolution of the `site-packages` directory. The following subgraph (representing a recursive directory search) was found repeated 8 times across different functions:

```
fs::path site_packages;
for (const auto& entry : fs::recursive_directory_iterator(cfg.project_env_path)) {
    if (entry.is_directory() && entry.path().filename() == "site-packages") {
        site_packages = entry.path();
        break;
    }
}
```

Affected Functions:

- `resolve_and_install`
- `prune_orphans`
- `uninstall_package`
- `run_package_tests`
- `run_all_package_tests`
- `freeze_environment`
- `audit_environment`
- `verify_environment`

3 Action

We refactored this common subgraph into a dedicated helper function:

```
fs::path get_site_packages(const Config& cfg);
```

This reduces code duplication and improves maintainability by localizing the environment discovery logic.