Common Code Patterns (Alpha-Equivalence)

Antigravity Agent

January 29, 2026

1 Overview

This report shows actual code patterns that appear multiple times in `spip.cpp`, detected using alpha-equivalence (variable name normalization).

2 Common Patterns

Pattern 1: Function-Call (57 occurrences)

Pattern ID: `call:6c66ec86`

Example Instance:

```
create_directories(scripts_dir);
```

Normalized Form (Alpha-Equivalent):

```
VAR(VAR);
```

Other Occurrences: 56 more instance(s) with the same structure

Pattern 2: Assignment (14 occurrences)

Pattern ID: `assign:3816d22f`

Example Instance:

```
extraction_helper = cfg.spip_root / "scripts" / "safe_extract.py"
```

Normalized Form (Alpha-Equivalent):

```
VAR = VAR.VAR / "STR" / "STR"
```

Other Occurrences: 13 more instance(s) with the same structure

Pattern 3: Assignment (12 occurrences)

Pattern ID: `assign:53cd0d06`

Example Instance:

```
out = get_exec_output(cmd)
```

Normalized Form (Alpha-Equivalent):

```
VAR = VAR(VAR)
```

Other Occurrences: 11 more instance(s) with the same structure

Pattern 4: Assignment (12 occurrences)

Pattern ID: `assign:0a6c11e2`

Example Instance:

```
pkg = q.front()
```

Normalized Form (Alpha-Equivalent):

```
VAR = VAR.VAR()
```

Other Occurrences: 11 more instance(s) with the same structure

Pattern 5: Assignment (11 occurrences)

Pattern ID: `assign:460ef88e`

Example Instance:

```
RESET = "\033[0m"
```

Normalized Form (Alpha-Equivalent):

```
VAR = "STR"
```

Other Occurrences: 10 more instance(s) with the same structure

Pattern 6: Function-Call (10 occurrences)

Pattern ID: `call:58c6c839`

Example Instance:

```
create_base_version(cfg, version);
```

Normalized Form (Alpha-Equivalent):

```
VAR(VAR, VAR);
```

Other Occurrences: 9 more instance(s) with the same structure

Pattern 7: Assignment (9 occurrences)

Pattern ID: `assign:f8ced854`

Example Instance:

```
spip_root = cfg.home_dir / ".spip"
```

Normalized Form (Alpha-Equivalent):

```
VAR = VAR.VAR / "STR"
```

Other Occurrences: 8 more instance(s) with the same structure

Pattern 8: Assignment (8 occurrences)

Pattern ID: `assign:918b7b68`

Example Instance:

```
command = = "prune") {
        setup_project_env(cfg)
```

Normalized Form (Alpha-Equivalent):

```
VAR = = "STR") { VAR(VAR)
```

Other Occurrences: 7 more instance(s) with the same structure

Pattern 9: If-Statement (7 occurrences)

Pattern ID: `if:a6ff1bda`

Example Instance:

```
if (command == "bundle") { if (args.size() < 2) {
            std::cout << "Usage:␣spip␣bundle␣<folder>" << }
```

Normalized Form (Alpha-Equivalent):

```
VAR == "STR" -> VAR (VAR.VAR() < NUM) { VAR::VAR << "STR" << VAR::VAR; VAR;
```

Other Occurrences: 6 more instance(s) with the same structure

Pattern 10: Assignment (7 occurrences)

Pattern ID: `assign:a561ba01`

Example Instance:

```
name = entry.path().filename().string()
```

Normalized Form (Alpha-Equivalent):

```
VAR = VAR.VAR().VAR().VAR()
```

Other Occurrences: 6 more instance(s) with the same structure

Pattern 11: Function-Call (6 occurrences)

Pattern ID: `call:a2b04579`

Example Instance:

```
for (size_t i = 0; i < cpp_files.size();
```

Normalized Form (Alpha-Equivalent):

```
VAR (VAR VAR = NUM; VAR < VAR.VAR();
```

Other Occurrences: 5 more instance(s) with the same structure

Pattern 12: Assignment (6 occurrences)

Pattern ID: `assign:6a27acac`

Example Instance:

```
ret = std::system(venv_cmd.c_str())
```

Normalized Form (Alpha-Equivalent):

```
VAR = VAR::VAR(VAR.VAR())
```

Other Occurrences: 5 more instance(s) with the same structure

Pattern 13: Function-Call (5 occurrences)

Pattern ID: `call:4ef80dae`

Example Instance:

```
format("{:x}", h);
```

Normalized Form (Alpha-Equivalent):

```
VAR("STR", VAR);
```

Other Occurrences: 4 more instance(s) with the same structure

Pattern 14: Function-Call (5 occurrences)

Pattern ID: `call:52c6a25d`

Example Instance:

```
bundle_package(cfg, args[1]);
```

Normalized Form (Alpha-Equivalent):

```
VAR(VAR, VAR[NUM]);
```

Other Occurrences: 4 more instance(s) with the same structure

Pattern 15: If-Statement (4 occurrences)

Pattern ID: `if:c4d85ad6`

Example Instance:

```
if (command == "prune") { setup_project_env(cfg);
        prune_orphans(cfg); }
```

Normalized Form (Alpha-Equivalent):

```
VAR == "STR" -> VAR(VAR); VAR(VAR);
```

Other Occurrences: 3 more instance(s) with the same structure

3 Interpretation

These patterns represent opportunities for:

- **Common Subexpression Elimination (CSE)**: Extract repeated logic into helper functions
- **Code Refactoring**: Reduce duplication and improve maintainability
- **Template/Macro Creation**: For frequently used patterns

The normalization process converts variable names to placeholders, allowing detection of structurally equivalent code even when identifiers differ.