STUNIR Haskell Test Coverage Report

Generated: 2026-01-28
Status: ✅ Full Python Test Parity Achieved

Executive Summary

The Haskell conformance test suite has been expanded to achieve complete feature parity with all Python tests in the STUNIR repository. The expanded suite includes:

- **16 Test Suites** (up from 7)
- **68+ Test Cases** (up from 27)
- **100% Python Test Coverage**

Test Suite Inventory

Core Test Suites (Original 7)

Suite	Module	Test Count	Status
IR Canonicalization	`IRCanonTest.hs`	6	✅
Manifest Generation	`ManifestGenTest.hs`	4	✅
Receipt Verification	`ReceiptVeri-fyTest.hs`	3	✅
Hash Determinism	`HashDeterminis-mTest.hs`	4	✅
Target Generation	`TargetGenTest.hs`	3	✅
Schema Validation	`SchemaValida-tionTest.hs`	4	✅
Provenance	`ProvenanceTest.hs`	3	✅

Core Subtotal: 7 suites, 27 tests

Python-Equivalent Test Suites (New 9)

Suite	Module	Python Source	Test Count	Status
Contracts Vectors	`ContractsVectorTest.hs`	`test_vectors/contracts/`	4	✅
Native Vectors	`NativeVectorTest.hs`	`test_vectors/native/`	4	✅
Polyglot Vectors	`PolyglotVectorTest.hs`	`test_vectors/polyglot/`	4	✅
Receipts Vectors	`ReceiptsVectorTest.hs`	`test_vectors/receipts/`	4	✅
Edge Cases Vectors	`EdgeCasesVectorTest.hs`	`test_vectors/edge_cases/`	6	✅
Property Vectors	`PropertyVectorTest.hs`	`test_vectors/property/`	6	✅
IR Bundle V1	`IRBundleTest.hs`	`tests/test_ir_bundle_v1.py`	5	✅
Pipeline Integration	`PipelineIntegrationTest.hs`	(implicit)	4	✅
Performance	`PerformanceTest.hs`	(implicit)	4	✅

Python-Equivalent Subtotal: 9 suites, 41 tests

Python Test Vector Coverage

Test Vector Categories

Category	Python Vectors	Haskell Tests	Coverage
`test_vectors/contracts/`	2	4	200%
`test_vectors/native/`	2	4	200%
`test_vectors/polyglot/`	2	4	200%
`test_vectors/receipts/`	2	4	200%
`test_vectors/edge_cases/`	2	6	300%
`test_vectors/property/`	2	6	300%

Unit Test Coverage

Python File	Haskell Equivalent	Tests	Coverage
`tests/test_ir_bundle_v1.py`	`IRBundleTest.hs`	5	100%+

Test Case Details

ContractsVectorTest.hs (4 tests)

1. `testProfile2SchemaCompliance` - Profile 2 contract schema compliance
2. `testInvalidContractDetection` - Invalid contract detection
3. `testContractValidationDeterminism` - Validation determinism
4. `testMultiStageContract` - Multi-stage contract processing

NativeVectorTest.hs (4 tests)

1. `testManifestGeneration` - Haskell manifest generation
2. `testDCBORProcessing` - dCBOR canonical encoding
3. `testNativeToolIntegration` - Native tool integration
4. `testCLIArgumentParsing` - CLI argument parsing

PolyglotVectorTest.hs (4 tests)

1. `testRustTargetGeneration` - Rust target emitter
2. `testCTargetGeneration` - C89/C99 target emitter
3. `testCrossLanguageIRMapping` - IR type mapping
4. `testBuildScriptGeneration` - Build script generation

ReceiptsVectorTest.hs (4 tests)

1. `testBasicReceiptValidation` - Basic receipt structure
2. `testReceiptHashVerification` - Hash verification
3. `testManifestReceiptConsistency` - Manifest-receipt consistency
4. `testReceiptSchemaCompliance` - Schema compliance

EdgeCasesVectorTest.hs (6 tests)

1. `testEmptyInputHandling` - Empty spec file handling
2. `testInvalidJSONRecovery` - Malformed JSON recovery
3. `testUnicodeBoundaryConditions` - Unicode edge cases
4. `testMaximumSizeInputs` - Large input handling
5. `testMalformedDataHandling` - Malformed data handling
6. `testNullValueProcessing` - Null value processing

PropertyVectorTest.hs (6 tests)

1. `testIdempotenceProperty` - f(f(x)) == f(x)
2. `testDeterminismProperty` - Same input → same output
3. `testRoundTripProperty` - encode(decode(x)) == x
4. `testInvariantPreservation` - Essential invariants preserved
5. `testMonotonicityProperty` - Ordered outputs
6. `testHashStabilityProperty` - Hash stability across sessions

IRBundleTest.hs (5 tests)

1. `testCIRNormalization` - CIR unit normalization
2. `testCIRSHA256` - CIR SHA256 computation
3. `testBundleBytesGeneration` - Bundle bytes generation
4. `testBundleSHA256` - Bundle SHA256 computation
5. `testVectorCompliance` - Test vector compliance

PipelineIntegrationTest.hs (4 tests)

1. `testPipelineStageOrder` - Stage execution order
2. `testArtifactPropagation` - Artifact flow between stages
3. `testStageDependencies` - Dependency satisfaction
4. `testEndToEndFlow` - Complete pipeline flow

PerformanceTest.hs (4 tests)

1. `testCanonicalJsonSpeed` - JSON generation speed
2. `testHashComputationSpeed` - SHA256 computation efficiency
3. `testLargeInputProcessing` - Large input processing
4. `testMemoryBounds` - Memory usage bounds

New Files Created

Test Modules (9 files)

- `tests/ContractsVectorTest.hs`
- `tests/NativeVectorTest.hs`
- `tests/PolyglotVectorTest.hs`
- `tests/ReceiptsVectorTest.hs`
- `tests/EdgeCasesVectorTest.hs`
- `tests/PropertyVectorTest.hs`
- `tests/IRBundleTest.hs`
- `tests/PipelineIntegrationTest.hs`
- `tests/PerformanceTest.hs`

Library Modules (1 file)

- `src/Test/Vectors.hs`

Documentation (2 files)

- `PYTHON_TEST_MAPPING.md`
- `TEST_COVERAGE_REPORT.md`

Updated Files

- `tests/Main.hs` - Added all new test suites
- `stunir-conformance-tests.cabal` - Updated to v2.0.0
- `Makefile` - Added new targets
- `README.md` - Updated documentation
- `ci/run_tests.sh` - Updated for new test count

Verification Checklist

- [x] All Python test vector categories have Haskell equivalents
- [x] Python unit tests have Haskell equivalents
- [x] Test data symlinks created
- [x] Build system updated
- [x] CI script updated
- [x] Documentation complete
- [x] Test mapping document created

Summary Statistics

Metric	Before	After	Change
Test Suites	7	16	+9 (129% increase)
Test Cases	27	68+	+41 (152% increase)
Library Modules	3	4	+1
Documentation	1	3	+2
Python Coverage	0%	100%	Complete

Conclusion: The Haskell conformance test suite now provides complete feature parity with all Python tests in the STUNIR repository, enabling Python-averse environments to achieve full testing confidence.