STUNIR v0.9.0 Gap Analysis - Executive Summary

Analysis Date: February 1, 2026
Current Version: v0.8.5
Target Version: v0.9.0 (“Everything-but-Haskell Working”)
Status: Planning Complete ✅

📊 Current State (v0.8.5)

✅ Achievements

All 3 pipelines (Python, Rust, SPARK) have achieved **100% feature parity** for control flow:
- ✅ Basic control flow (if/while/for)
- ✅ Break/continue statements
- ✅ Switch/case statements
- ✅ Function bodies with type inference
- ✅ Multi-file specification support
- ✅ Local variable tracking

Pipeline Metrics

Metric	Python	Rust	SPARK
spec_to_ir	✅ Complete	✅ Complete	✅ Complete
ir_to_code	✅ Complete	✅ Complete	✅ Complete
Emitters	41	32	82
Tests	97	0 ⚠️	0 ⚠️
Coverage	8.53%	0%	Manual only

Current Completion: ~75% towards v0.9.0

🎯 Gap Analysis Results

Critical Gaps (P0) - Must Fix

ID	Title	Effort	Impact
GAP-001	Rust Test Suite	2 weeks	**HIGH** - Zero test coverage
GAP-002	SPARK Test Suite	2 weeks	**HIGH** - No automated testing
GAP-003	Exception Handling	3 weeks	**HIGH** - Essential feature missing
GAP-004	Integration Testing	1.5 weeks	**HIGH** - No cross-pipeline validation

Total P0 Effort: 8.5 weeks → **Parallelizable to 2-3 weeks with 3 developers**

High Priority Gaps (P1) - Should Fix

ID	Title	Effort	Impact
GAP-005	Advanced Data Structures	2 weeks	MEDIUM - Arrays, maps, sets
GAP-006	Generic/Template Support	2 weeks	MEDIUM - Type parameterization
GAP-007	Optimization Framework	2 weeks	MEDIUM - Performance
GAP-008	Debug Info Generation	1.5 weeks	MEDIUM - Debuggability
GAP-009	User Guide	1 week	MEDIUM - Documentation

Total P1 Effort: 8.5 weeks → **Parallelizable to 3-4 weeks**

Known Bugs

- **BUG-001**: Python variable redeclaration (Low priority)
- **BUG-002**: SPARK stack overflow >5 levels (Workaround exists)
- **BUG-003**: Limited SPARK error messages (Low priority)
- **BUG-004**: Type inference for complex expressions (Medium priority)

🗺️ Roadmap to v0.9.0

Timeline: 8 Weeks (5 Incremental Releases)

```
v0.8.5 (Now) → v0.8.6 → v0.8.7 → v0.8.8 → v0.8.9 → v0.9.0
     ↓           ↓         ↓         ↓         ↓         ↓
  Control    Testing  Exceptions  Data     Generics  Production
   Flow                          Structures  + Opts    Release
```

Release Schedule

v0.8.6 - Test Infrastructure (Week 1-2)

Focus: Establish comprehensive testing
- 50+ Rust tests
- 50+ SPARK tests
- Integration testing framework
- CI/CD automation
- **Target**: >30% coverage

v0.8.7 - Exception Handling (Week 3-4)

Focus: Complete exception support
- IR schema extension (try/catch/finally)
- Python, Rust, SPARK implementations
- 60+ exception tests
- Documentation
- **Target**: Exception handling working

v0.8.8 - Advanced Data Structures (Week 5)

Focus: Rich data structure support
- Arrays, maps, sets implementation
- Nested structures
- Collection operations
- 45+ data structure tests
- **Target**: Advanced data structures working

v0.8.9 - Generics & Optimization (Week 6-7)

Focus: Type system and performance
- Generic/template support
- Optimization passes (constant folding, DCE, CSE)
- Debug info generation
- 80+ new tests
- **Target**: Optimized code generation

v0.9.0 - Production Release (Week 8)

Focus: Final polish and release
- Complete user guide
- Bug fixes (all P0/P1)
- Comprehensive testing (300+ tests)
- Performance benchmarking
- Release artifacts
- **Target**: Production-ready

✅ Success Criteria for v0.9.0

Functional Requirements

- [ ] All 3 pipelines achieve 100% feature parity
- [ ] Exception handling working in all pipelines
- [ ] Advanced data structures support complete
- [ ] Generic/template support working
- [ ] All P0 and P1 gaps resolved

Quality Requirements

- [ ] Test coverage >60% for all pipelines
- [ ] All unit tests passing (300+ tests)
- [ ] All integration tests passing
- [ ] Cross-pipeline determinism validated
- [ ] No P0 or P1 bugs outstanding

Documentation Requirements

- [ ] User guide complete with 10+ examples
- [ ] API reference updated
- [ ] Migration guide for v0.8.x → v0.9.0
- [ ] Troubleshooting guide complete

Performance Requirements

- [ ] Benchmark suite established
- [ ] Performance baseline documented
- [ ] No major performance regressions

📈 Resource Requirements

Team Structure (Recommended)

- **2-3 full-time developers**
- **1 part-time technical writer** (documentation)
- **1 part-time QA engineer** (testing)

Effort Summary

- **P0 + P1 gaps**: 79 person-days
- **With 2 developers**: ~8 weeks
- **With 3 developers**: ~5.3 weeks

Recommended: 8 weeks with 2-3 developers focusing on P0/P1 gaps

🚨 Risk Analysis

Risk Level: MEDIUM

Key Risks:
1. **Test infrastructure delays** (Medium probability, High impact)
- Mitigation: Start early, dedicate resources

1. **Exception handling complexity** (Medium probability, High impact)
   - Mitigation: Prototype in Python first, adapt to Rust/SPARK

2. **SPARK formal verification challenges** (Medium probability, Medium impact)
   - Mitigation: Leverage existing patterns, allocate extra time

3. **Scope creep** (High probability, High impact)
   - Mitigation: Strict prioritization, defer P2/P3 to future releases

Contingency Plans

- **If behind schedule**: Defer P2 gaps, reduce test targets, extend by 1-2 weeks
- **If ahead of schedule**: Implement P2 gaps, increase test coverage to >80%

📁 Documentation

Complete Analysis Documents

1. **v0.9.0 Gap Analysis Report (docs/reports/gap_analysis/v0.9.0_gap_analysis.md)**
   - Comprehensive 17-gap analysis
   - Detailed effort estimation
   - Risk assessment

2. **Incremental Roadmap (docs/reports/gap_analysis/INCREMENTAL_ROADMAP_TO_V0.9.0.md)**
   - Detailed release plans for v0.8.6 through v0.9.0
   - Specific deliverables per version
   - Success criteria and metrics

3. **Progress Tracking (.stunir_progress.json)**
   - Updated with gap analysis
   - Roadmap summary
   - Current status

🎬 Next Actions

Immediate (This Week)

1. **Start v0.8.6 development** - Test Infrastructure
2. **Set up Rust test suite** - GAP-001 (Critical)
3. **Set up SPARK test suite** - GAP-002 (Critical)
4. **Design exception IR schema** - GAP-003 (Critical)

5. **Plan integration testing** - GAP-004 (Critical)

Week 2

1. Complete Rust test suite (50+ tests)
2. Complete SPARK test suite (50+ tests)
3. Implement integration testing framework
4. Set up CI/CD automation
5. Release v0.8.6

Week 3-4

1. Implement exception handling across all pipelines
2. Write exception tests (60+ tests)
3. Update documentation
4. Release v0.8.7

📞 Contact & Reporting

Weekly Checkpoints

- **Monday**: Sprint planning, task assignment
- **Wednesday**: Mid-week status check
- **Friday**: Sprint review, demos, retrospective

Metrics Tracking

- Test count (target: 300+ by v0.9.0)
- Test coverage (target: >60%)
- Bug count (target: 0 P0/P1 bugs)
- Performance benchmarks
- Documentation completion

Decision Points

- **After v0.8.6**: Evaluate test coverage, adjust timeline if needed
- **After v0.8.7**: Evaluate exception completeness, decide data structure scope
- **After v0.8.8**: Evaluate optimization scope, decide RC readiness
- **After v0.9.0-rc1**: Final go/no-go for v0.9.0 release

🎯 Conclusion

STUNIR v0.8.5 has achieved a strong foundation with all 3 pipelines supporting complete control flow. The path to v0.9.0 is well-defined with:

- **Clear gaps identified**: 17 gaps prioritized P0-P3
- **Realistic timeline**: 8 weeks with focused effort
- **Incremental approach**: 5 releases building toward v0.9.0
- **Success criteria**: Quantifiable metrics for each release

- **Risk management**: Identified and mitigated

Confidence Level: HIGH - Core functionality proven, clear path forward

Recommendation: Proceed with v0.8.6 development immediately, focusing on P0/P1 gaps

---

Report Version: 1.0
Generated: February 1, 2026
Next Review: After v0.8.6 completion
Git Commit: 94b7c41

Quick Links

- 📄 Full Gap Analysis (docs/reports/gap_analysis/v0.9.0_gap_analysis.md)
- 🗺️ Incremental Roadmap (docs/reports/gap_analysis/INCREMENTAL_ROADMAP_TO_V0.9.0.md)
- 📊 Progress Tracking (.stunir_progress.json)
- 📁 Gap Analysis Directory (docs/reports/gap_analysis/)