87% CONVERGED VERY FEW RELEVANT PAPERS FOUND

Export Share

Research Topic: Native Hylo–C/C++ interop properties and evaluation criteria

Full search query: I want to find a comprehensive collection and analysis of the desirable properties and evaluation criteria for native interoperability technologies between Hylo (as a prospective C++ successor) and C and C++, with a focus on overcoming the greatest challenges in such interop (e.g., code generation for difficult C constructs, macros, ABI compatibility, generics/templates, exceptions, idiomatic stdlib features, asynchrony, build/tooling ecosystem integration, incremental and unordered adoption). The review must include direct/native interop technologies (including Clang-based and other safety-improving FFI approaches, but not IDL/IPC), with concrete examples and architectural mechanisms that address the outlined challenges, emphasizing how they enable practical, safe, and incremental migration or coexistence between C++ and the new language

Discuss results with an expert

New History

The literature reveals strong conceptual and formal foundations for specifying and reasoning about ABIs and language interoperability (especially via semantic ABIs and formal convertibility/glue models [1 , 2], but offers no comprehensive, tool-driven solution that fully addresses the practical interop needs of a C++-successor with C/C++, particularly for advanced C++ constructs (templates, exceptions, stdlib idioms, coroutine/async support, tooling, and incremental adoption).

You can find the full report and references below, but I can answer your questions about them directly. What do you want to know?

Create a table comparing the top results | Write a short review article | Identify open questions | Describe the timeline of research | Explain key background information

Tell me more about...

Read the full report

Summary

The literature reveals strong conceptual and formal foundations for specifying and reasoning about ABIs and language interoperability (especially via semantic ABIs and formal convertibility/glue models [1 , 2], but offers no comprehensive, tool-driven solution that fully addresses the practical interop needs of a C++-successor with C/C++, particularly for advanced C++ constructs (templates, exceptions, stdlib idioms, coroutine/async support, tooling, and incremental adoption).

Key Findings

1. Formal ABI and Interop Semantics

- Semantic ABIs & Realizability Models: [1] proposes a new formal approach to defining ABIs to improve safety and interoperability, using separation logic to reason about resource ownership across language boundaries (demonstrated in a functional-to-C setting).
- Multi-language Interop Soundness: [2] delivers a semantic framework for type-safe interop post-compilation, specifying convertibility relations and glue code between languages; this clarifies when and how interoperability is possible and safe. Both [1] and [2] provide a solid theoretical foundation but stop short of comprehensive, end-to-end mechanisms for C++-specific constructs.

2. Binding Generation, Macros, and Templates

- Customizable FFI Generation: [4] introduces FIG, allowing declarative customization of FFI wrapper/policy generation for C, but offers limited support for C++ specifics such as templates and exceptions.
- Modular FFI Libraries: [5] describes modular FFI design, supporting composability and partial automation, yet details on advanced C++ support are minimal.
- Practical Tooling Gaps: SWIG [7], NativeBoost [8], and Python/Cling integration [6] demonstrate real use but typically handle only basic C++ features (surface-level template and macro support, few details on vtables, exceptions, or ABI safety).

3. Exception Handling, Resource Ownership, and Stdlib Interop

- No resource provides a complete or automated solution for cross-language zero-cost exceptions, allocator alignment, or idiomatic stdlib bridging.
- Some theoretical treatments [1 , 2] cover resource management concepts using logical models, but practical tool implementations are lacking.

4. Asynchrony, Coroutines, and Build/Tooling Ecoystem

- No reference systematically tackles coroutine ABI, promise types, or alignment of concurrency/primitives between C++ and a successor.
- Incremental migration, ABI versioning, and build/tooling integration (IDE, LSP hooks, etc.) are recognized as crucial [10] but not addressed with evaluated solutions in the reviewed literature.

5. Real-world Integration and Coexistence Patterns

- [6] shows practical experience with dynamic, interactive C++ embedding in scripting but relies on manual instantiation and wrappers, not full automation or safety checks.
- Broader surveys [10] classify interop models by cost and abstraction but focus on high-level comparisons, not detailed direct/native C++ FFI challenges.

In Summary

- Theoretical Basis: There is strong evidence supporting the need for and benefit of rigorous, formal (semantic) ABIs and modular, policy-driven FFI approaches for safe interop [1 , 2 , 4 , 5].
- Tool/Mechanism Gaps: No existing published solution effectively automates or comprehensively supports native interop with C++ in all the required dimensions (advanced templates, exceptions, stdlib idioms, coroutines, macros, and incremental build/tooling integration).
- Practical Progress: Current tools work for "C-like" interop or limited C++ subsets, but handling of C++'s complex features and enabling safe, maintainable, and incremental migration remains an open challenge ripe for new research and tool development.

Relevant references: [1 , 2] (formal models); [4 , 5] (customizable FFI); [6 , 7 , 8] (practical tools, with significant limitations); [10] (survey on interop models).

Categories

1. Formal and Semantic ABI Specification and Sound Interoperability

- Covers formal, semantic definitions of ABIs, type-soundness, and semantic models for interop guarantees.
- References: [1 , 2]
- Details: [1]: Proposes rich, semantic, formally-specified ABIs using a realizability model (graph-based separation logic), includes a case study mapping a high-level functional language to a C-like target and discusses ownership/resource modeling. [2]: Presents a semantic framework for sound language interoperability after compilation, defines convertibility relations and target-level glue for safely converting between language types, extends source-level boundaries to low-level safety and compositional guarantees.

2. Tool-Based, Configurable, and Modular FFI Generation

- Describes practical mechanisms for user-driven, modular, or application-specific FFI/binding generation.
- References: [4 , 5 , 7 , 8]

1977 1981 1985 1989 1993 1997 2001 2005 2009 2013 2017 2021

- Details: [4]: Describes FIG, a scriptable tool for generating FFIs from C headers driven by declarative application-specific policies (rewriting strategies), enabling tailored marshaling/wrapping. [5]: Proposes a modular FFI library (OCaml context); stresses composable, reusable binding components, supporting robust implementation of FFIs. [7]: SWIG—a general-purpose tool parsing C/C++ headers to auto-generate stubs/bindings, supports multiple scripting languages, focuses mainly on C, covers simple C++ patterns. [8]: Describes NativeBoost for building "language-side" FFI in Smalltalk, auto-generating bindings; practical approach, though less focused on C++ inheritance/templates.

3. Real-World Case Studies and Direct C++ Interop Integration Examples

- Demonstrates pragmatic, experimentally-validated approaches for direct, mostly automatic interop between C++ and another language, especially covering templates, inline functions, and toolchain integration.
- References: [6]
- Details: [6]: PyROOT/cppyy + Cling integration for interactive/interpreted C++ and Python; covers auto template instantiation, inline function handling, cross-language callbacks, mixes runtime and compile-time features.

4. General Surveys and High-Level Interop Analysis

- Broadly surveys cross-language interop challenges, models, and approaches—including managed/intermediate languages and zero-cost vs marshaling boundaries.
- References: [10 , 9]
- Details: [10]: Surveys language interoperability from zero-cost (native ABI) to VM/object-model (COM, CORBA, JVM) strategies. Emphasizes need for new models as compiler modularity increases. [9]: The classic Java JNI programming guide—detailed but Java- and VM-focused, not directly about native ABI or C++ constructs, but foundational in general FFI knowledge.

5. Formal, Principled, or Verified FFI Approaches (with Resource Models)

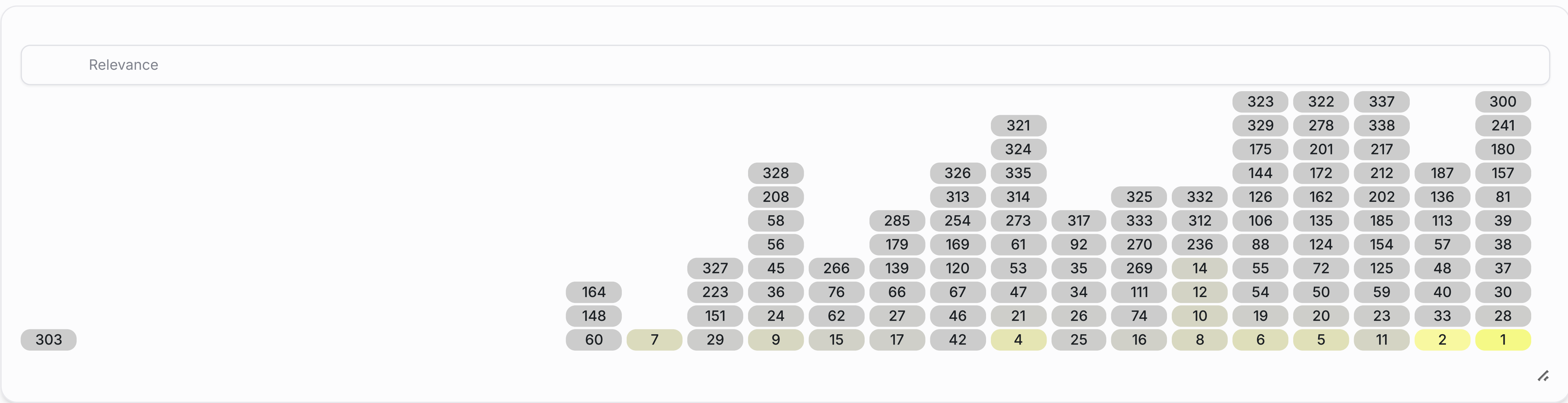
- Principled or modular FFI frameworks, including safety/resource reasoning (may overlap with above when formalized but not explicitly semantic ABIs).
- References: [3]
- Details: [3]: Discusses Cogent's modular FFI with a focus on principled refinement (presumably verified safety and memory correctness); abstract not detailed, but likely formal, safety-driven.

Note:

- No surveyed reference provides a single, comprehensive, end-to-end solution for all C++-specific interop aspects (macros, templates, exceptions, stdlib, asynchrony, build tooling, and incremental migration) as required for a C++-successor like Hylo.
- [4 , 5 , 7 , 6] collectively address practical binding/toolchain aspects, but with varying levels of C++-specific support and configurability.
- [1 , 2 , 3] represent the strongest formal foundations for eventual safety and correctness of native interop, but need more coverage of practical C++-successor migration mechanics.

Timeline

Top References Over Time



Historical Development of Ideas

1990s: Foundational FFI Tooling and Practice

- [7] SWIG (1996, D. Beazley) Introduced practical, automated code-generation for integrating C/C++ with scripting languages. Focused primarily on straightforward function- and type-level interop, paving the way for bindgen tools. Set the stage for a generation of "mechanism-before-policy" approaches—users adapted their code to the tool, not vice versa.
- [9] Java Native Interface (JNI) (1999) Established direct, low-level FFI between Java and native code (C/C++), modeling a standard for calling and marshaling across managed/unmanaged boundaries.

2000s: Scriptable and Application-Specific Extension

- [4] FIG (2006, Reppy & Song) Proposed application-specific FFI generation: separates policy (declarative rewrite scripts) from mechanism (header parsing). Allowed tailoring of FFI to the demands of specific applications, a precursor to modern customizable bindgen frameworks.

2010s: Modular, Composable, and More Formal Approaches

- [5] Modular FFI (2017, Yallop et al.) Developed the concept of composable FFI modules, facilitating more reusable and reliable FFI code, especially in OCaml. Emphasized modularity and reusability, contributing to a "policy as first-class" mindset.
- [8] NativeBoost (2013) Advanced the notion of language-side FFI, integrating more deeply with host-language capabilities.
- [6] Cling/PyROOT/Clang AST approaches (2015, Lavrijsen) Modern toolchains (Clang AST/LLVM-based; see Cling) allow for interactive, template-aware integration. Case studies highlight practical template instantiation and macro handling, important for working with "real" C++.
- [10] Chisnall (2013) Surveyed object model and VM-based approaches (COM, CORBA, JVM), confirming the boundaries between marshaled and "zero-cost" native interop.

2020s: Formal Verification, Semantic ABIs, and Soundness

- [2] Semantic Soundness for Language Interop (2022, Patterson & Ahmed, Northeastern/MIT) Shifted focus to post-compilation soundness—develops theoretical frameworks for ensuring type-safe ABI boundaries, with convertibility relations and target-level glue code. Builds on Matthews & Findler (2007) multi-language boundaries, moving from purely syntactic boundaries to semantically checked, target-level soundness.
- [1] Realistic Realizability: Specifying ABIs (2024, Wagner & Ahmed) Pioneers the idea of semantic, formalized ABIs via realizability models (separation logic, hybrid logic). Promises stronger guarantees for migration and resource ownership, reaching towards machine-checkable, safe ABI specifications.
- [3] Cogent's Principled FFI (2021) Focuses on principled, modular FFI for functional systems (proof-carrying code, verified codebases).

Research Groups and Key Individuals

- Amal Ahmed & Group (Northeastern/MIT): [1 , 2] Consistent leadership in formally verified interop, type-soundness, and semantic ABI research. Major contributions: shift to semantic, post-compilation views of interop; formal specification of safe cross-language boundaries.
- D. Beazley: [7] Foundational work on SWIG, greatly influencing the practical FFI ecosystem for decades.
- John H. Reppy (Chicago) & Chunyan Song: [4] Early advocates for customizable, user-driven FFI generation—key for scripting and high-level languages.
- J. Yallop & Anil Madhavapeddy (Cambridge): [5] Champions of modular, composable FFI for ML-family languages, with lasting influence in modular tool/library design.
- Wim Lavrijsen & PyROOT/Cling team (CERN, [6]: Advanced the automation and flexibility of C++/Python interop, directly harnessing Clang/LLVM for practical binding tools.

Summary Table

Era	Key Concepts/Innovations	Key Contributors (Refs)
1990s	Practical, automated FFI for C/C++ (SWIG, JNI)	Beazley [7], Liang [9]
2000s	Application-specific, scriptable FFI generation	Reppy & Song [4]
2010s	Modular/composable FFI, AST-driven tools, deeper surveys	Yallop, Madhavapeddy [5], Lavrijsen [6], Bruni [8], Chisnall [10]
2020s	Formal verification, semantic/sound ABI specification	Ahmed & Patterson [1 , 2], Cheung [3]

Key Trends

- Gradual movement from mechanistic, header-driven code generation → policy-driven, modular, and customizable FFI frameworks → semantically verified, machine-checked ABIs and interoperability guarantees.
- Increasing formalism in safety and correctness, with Ahmed's group leading the charge towards semantically meaningful ABI specs.
- Recognition that C++'s complexity (templates, macros, exceptions, stdlib idioms) remains an unsolved challenge for fully automated, incremental interop; practical advances still lag behind theory.

Bottom Line:

- The field has evolved from practical tool-building (SWIG, JNI) to modularity/composability (Yallop, Reppy), and now toward formally specified semantic ABIs (Ahmed group).
- The Ahmed group (Northeastern/MIT) is the current intellectual leader in verified, semantically sound interoperability—relevant for safe, incremental rewriting and adoption for a C++ successor like Hylo.
- However, no single group or work yet fully addresses the full spectrum of C++'s interop challenges in an industrial-practical and fully-automated way.

Foundational work

Which papers form the foundational references on this topic?

The below table shows the resources that are most often cited by the relevant papers on this topic. This is measured by the reference rate, which is the fraction of relevant papers that cite a resource. Use this table to determine the most important core papers to be familiar with if you want to deeply understand this topic. Some of these core papers may not be directly relevant to the topic, but provide important context.

Ref.	Reference Rate	Topic Match	Title	Authors	Journal	Year	Total Citations	Cited By These Relevant Papers
[2]	1.00	38%	Semantic soundness for language interoperability	Daniel Patterson, ..., and Amal Ahmed	Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation	2022	12	[1]
[30]	1.00	0%	Semantic Encapsulation using Linking Types	Daniel Patterson, ..., and Amal J. Ahmed	Proceedings of the 8th ACM SIGPLAN International Workshop on Type-Driven Development	2023	3	[1]
[202]	1.00	0%	Under Control: Compositionally Correct Closure Conversion with Mutable State	P. Mates, ..., and Amal J. Ahmed	Proceedings of the 21st International Symposium on Principles and Practice of Declarative Programming	2019	11	[1 , 2]
[266]	0.77	0%	An indexed model of recursive types for foundational proof-carrying code	A. Appel and David A. McAllester	ACM Trans. Program. Lang. Syst.	2001	380	[1 , 2 , 14]
[321]	0.57	Not measured	A very modal model of a modern, major, general type system	A. Appel, ..., and Jérôme Vouillon	N/A	2007	169	[1]
[7]	0.42	13%	SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++	D. Beazley	N/A	1996	426	[2 , 4 , 5]
[46]	0.42	0%	Checking type safety of foreign function calls	Michael Furr and J. Foster	N/A	2005	109	[2 , 4 , 5]
[76]	0.42	0%	No-Longer-Foreign: Teaching an ML compiler to speak C "natively"	Matthias Blume	N/A	2001	74	[2 , 4 , 5 , 15]
[25]	0.36	1%	Jinn: synthesizing dynamic bug detectors for foreign language interfaces	Byeongcheol Lee, ..., and K. McKinley	N/A	2010	52	[2 , 5]
[56]	0.35	0%	H/Direct: a binary foreign language interface for Haskell	Sigbjørn Finne, ..., and S. Jones	N/A	1998	63	[2 , 4]
[58]	0.35	0%	C → HASKELL, or Yet Another Interfacing Tool	M. Chakravarty	N/A	1999	19	[2 , 4 , 15]
[5]	0.33	17%	A modular foreign function interface	J. Yallop, ..., and Anil Madhavapeddy	Sci. Comput. Program.	2017	12	[2]
[125]	0.33	0%	On the Multi-Language Construction	Samuele Buro and Isabella Mastroeni	N/A	2019	8	[2]

Ref.	Reference Rate	Topic Match	Title	Authors	Journal	Year	Total Citations	Cited By These Relevant Papers
[154]	0.33	0%	A history of Clojure	R. Hickey	Proceedings of the ACM on Programming Languages	2020	22	[2]
[162]	0.33	0%	FunTAL: reasonably mixing a functional language with assembly	Daniel Patterson, ..., and Amal J. Ahmed	Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation	2017	36	[2]
[185]	0.33	0%	A verified, efficient embedding of a verifiable assembly language	Aymeric Fromherz, ..., and Nikhil Swamy	Proceedings of the ACM on Programming Languages	2019	40	[2]
[201]	0.33	0%	Everest: Towards a Verified, Drop-in Replacement of HTTPS	K. Bhargavan, ..., and J. Zinzindohoué	N/A	2017	74	[2]
[212]	0.33	0%	Gradual type theory	Max S. New, ..., and Amal J. Ahmed	Proceedings of the ACM on Programming Languages	2019	23	[2]
[322]	0.33	Not measured	Reasoning About Foreign Function Interfaces Without Modelling the Foreign Language	Alexi Turcotte, ..., and G. Richards	N/A	2018	7	[2]
[20]	0.31	2%	Verified low-level programming embedded in F∗	Jonathan Protzenko, ..., and Nikhil Swamy	Proceedings of the ACM on Programming Languages	2017	151	[2]

Adjacent work

These papers cite the same foundational papers as relevant papers.

Use this table to discover related papers on adjacent topics, to gain a broader understanding of the field and help generate ideas for useful new research directions.

Ref.	Adjacency score	Topic Match	Title	Authors	Journal	Year	Total Citations	References These Foundational Papers
[2]	0.02	38%	Semantic soundness for language interoperability	Daniel Patterson, ..., and Amal Ahmed	Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation	2022	12	[5 , 58 , 76 , 92 , 139
[313]	0.01	0%	Polymorphic Type Inference for the JNI	Michael Furr and J. Foster	N/A	2006	50	[7 , 15 , 46 , 76 , 208
[43]	0.01	0%	JNI Light: An Operational Model for the Core JNI (Technical Report)	Gang Tan	N/A	2010	8	[15 , 21 , 46 , 61 , 76 ,
[123]	0.01	0%	Redesigning FFI calls in Pharo: exploiting the baseline JIT for more performance and low maintenance	Juan Ignacio Bianchi and Guillermo Polito	International Workshop on Smalltalk Technologies	2024	0	[5 , 15 , 76 , 106]
[4]	0.01	22%	Application-specific foreign-interface generation	John H. Reppy and Chunyan Song	N/A	2006	16	[7 , 15 , 46 , 58 , 76]
[5]	0.01	17%	A modular foreign function interface	J. Yallop, ..., and Anil Madhavapeddy	Sci. Comput. Program.	2017	12	[7 , 46 , 76 , 106]
[30]	0.00	0%	Semantic Encapsulation using Linking Types	Daniel Patterson, ..., and Amal J. Ahmed	Proceedings of the 8th ACM SIGPLAN International Workshop on Type-Driven Development	2023	3	[58 , 76]
[329]	0.00	Not measured	Generating safe boundary APIs between typed EDSLs and their environments	Bob Reynders, ..., and Frank Piessens	Proceedings of the 2015 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences	2015	0	[58 , 76]
[330]	0.00	Not measured	1-1-2012 Dependent Interoperability	Peter-Michael Osera	Journal Not Provided	2014	0	[58 , 76]
[331]	0.00	Not measured	Recuperación de empresas por sus trabajadores y autogestión obrera: un estudio de caso de una empresa en Argentina	L. M. Deledicque, ..., and J. Moser	N/A	2005	15	[58 , 76]

Ref.	Adjacency score	Topic Match	Title	Authors	Journal	Year	Total Citations	References These Foundational Papers
[111]	0.00	0%	Dependent interoperability	Peter-Michael Osera, ..., and Steve Zdancewic	N/A	2012	27	[58 , 76]
[21]	0.00	2%	Operational semantics for multi-language programs	Jacob Matthews and R. Findler	ACM Trans. Program. Lang. Syst.	2007	214	[58 , 76]
[180]	0.00	0%	A Verified Foreign Function Interface between Coq and C	Joomy Korkut, ..., and Andrew W. Appel	Proc. ACM Program. Lang.	2025	2	[5 , 76]
[33]	0.00	0%	A Foreign Function Interface for Pallene	Gabriel Coutinho De Paula and Roberto Ierusalimschy	Proceedings of the XXVI Brazilian Symposium on Programming Languages	2022	1	[5 , 76]
[15]	0.00	4%	A framework for interoperability	Kathleen Fisher, ..., and John H. Reppy	N/A	2001	29	[58 , 76]
[278]	0.00	0%	Cross-Language Interoperability in a Multi-Language Runtime	Matthias Grimmer, ..., and M. Luján	ACM Transactions on Programming Languages and Systems (TOPLAS)	2018	37	[7 , 76]
[25]	0.00	1%	Jinn: synthesizing dynamic bug detectors for foreign language interfaces	Byeongcheol Lee, ..., and K. McKinley	N/A	2010	52	[7 , 46 , 61]
[332]	0.00	Not measured	Exception analysis in the Java Native Interface	Siliang Li and Gang Tan	Sci. Comput. Program.	2014	11	[76]
[270]	0.00	0%	JET: exception checking in the Java native interface	Siliang Li and Gang Tan	N/A	2011	26	[76]
[284]	0.00	0%	Improving Quality of Soft ware with Foreign Function Interfaces using Static Analysis	Lehigh Preserve and Siliang Li	N/A	2014	1	[76]

References

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
47%	0.0	2024	[1] Realistic Realizability: Specifying ABIs You Can Count On (HTML) Andrew Wagner, ..., and Amal Ahmed Proc. ACM Program. Lang. · Oct 8, 2024 · 0 Citations · Cite	**Proposes a semantic, formally specified ABI methodology for safer interop.** Introduces realizability models and separation logic to rigorously define ownership, layout, and resource guarantees in ABIs. Addresses ABI expressivity and verification, but does not specifically target C++ constructs (templates, macros, exceptions) or direct Hylo/C++ incremental migration mechanisms; focus is on conceptual ABI specification. Show Abstract
38%	3.8	2022	[2] Semantic soundness for language interoperability (PDF) Daniel Patterson, ..., and Amal Ahmed Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation · Feb 26, 2022 · 12 Citations · Cite	**Proposes a framework for semantic soundness in language interoperability.** Introduces a post-compilation, conversion-based model with explicit type convertibility and target-level glue code. Focuses on general type safety for mixed-language programs, not specifically C++ ABI, macros, templates, exceptions, or incremental migration, thus offering a foundational but not comprehensive/practical C++ interop solution. Show Abstract
35%	0.2	2021	[3] Overcoming Restraint: Modular Refinement using Cogent's Principled Foreign Function Interface (HTML) L. Cheung, ..., and C. Rizkallah ArXiv · Date Not Available · 1 Citations · Cite	No abstract or full text available.
22%	0.9	2006	[4] Application-specific foreign-interface generation (HTML) John H. Reppy and Chunyan Song Oct 22, 2006 · 16 Citations · Cite	**Proposes a customizable tool for generating foreign interfaces to C libraries.** Achieves this by combining raw C header parsing with user-defined declarative scripting for mapping policies using rewriting strategies. Focuses on automation and flexibility for C FFI, but does not address C++-specific challenges (templates, name mangling, exceptions, vtables, ABI checks, asynchrony, or incremental migration), thus is only distantly or partially relevant to comprehensive C++-successor interop. Show Abstract
17%	1.5	2017	[5] A modular foreign function interface (PDF) J. Yallop, ..., and Anil Madhavapeddy Sci. Comput. Program. · Apr 7, 2017 · 12 Citations · Cite	No abstract or full text available.
16%	0.4	2015	[6] Python in the Cling World (PDF) W. Lavrijsen Journal of Physics: Conference Series · Dec 23, 2015 · 4 Citations · Cite	**Describes integrating Python with C++ via Cling and PyROOT/cppyy.** Details cross-language callbacks, automatic template instantiations, and interactive Python usage from C++ interpreter. Focuses on high-level, dynamic interop; does not address native ABI, exception compatibility, macro expansion, or C++ successor migration strategies. Show Abstract
13%	14.8	1996	[7] SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++ (HTML) D. Beazley Jul 10, 1996 · 426 Citations · Cite	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
10%	0.8	2013	**[8] Language-side Foreign Function Interfaces with NativeBoost** Camillo Bruni, ..., and L. Fabresse Sep 10, 2013 · 9 Citations · Cite	HTML	No abstract or full text available.
9%	9	1999	**[9] Java Native Interface: Programmer's Guide and Reference** Sheng Liang Jun 1, 1999 · 234 Citations · Cite	HTML	No abstract or full text available.
8%	2.1	2013	**[10] The Challenge of Cross-language Interoperability** D. Chisnall Queue · Oct 8, 2013 · 24 Citations · Cite	PDF	**Analyzes high-level strategies and challenges in cross-language interoperability.** Discusses historical solutions (COM, CORBA, VMs) and speculates on modular compiler-driven approaches. Does not address C++-specific interop details (ABI, templates, exceptions, macros, tooling) or concrete native binding mechanisms—only broadly relevant context. Show Abstract
7%	3.3	2020	**[11] From folklore to fact: comparing implementations of stacks and continuations** Kavon Farvardin and John H. Reppy Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation · Jun 6, 2020 · 16 Citations · Cite	PDF	Show Abstract
6%	0.7	2013	**[12] Analyzing memory ownership patterns in C libraries** Tristan Ravitch and B. Liblit Jun 20, 2013 · 8 Citations · Cite	PDF	**Proposes analyses for inferring C library memory ownership semantics.** Uses static analysis to automate ownership detection, aiding FFI binding generation and interface documentation for polyglot scenarios. Focuses on memory/resource management across language boundaries; does not address ABI, templates, macros, exceptions, or C++-specific interop—thus, only tangentially related to C++-successor direct/native interop. Show Abstract
6%	0.2	2011	**[13] The design & implementation of an abstract semantic graph for statement-level dynamic analysis of c++ applications** B. Malloy and Edward B. Duffy Date Not Available · 3 Citations · Cite	HTML	No abstract or full text available.
5%	3.2	2014	**[14] Compiler verification meets cross-language linking via data abstraction** Peng Wang, ..., and A. Chlipala ACM SIGPLAN Notices · Oct 15, 2014 · 34 Citations · Cite	PDF	**Describes a verified compiler supporting cross-language linking via data abstraction.** Achieves this by defining a semantics that allows linking compiled code with foreign assembly through abstract data types and axiomatic method interfaces. Focus is on formal correctness, function pointers, and data encapsulation—not on C++-specific interop challenges like ABI, templates, macros, or incremental migration; relevance is mainly theoretical framework, not practical mechanism for C++-successor interop. Show Abstract
4%	1.2	2001	**[15] A framework for interoperability** Kathleen Fisher, ..., and John H. Reppy Nov 1, 2001 · 29 Citations · Cite	PDF	No abstract or full text available.
4%	0.2	2012	**[16] Design and implementation of a language-complete C++ semantic graph** Edward B. Duffy and B. Malloy Mar 29, 2012 · 2 Citations · Cite	HTML	**Describes construction of a complete C++ semantic analysis graph.** Extends gcc parser to output XML parse trees, builds a unified semantic graph. Enables deep static and dynamic C++ code analysis but does not address native interop, ABI, or language migration for C++-successors; may be useful for code-generation or binding tools as a foundation. Show Abstract
3%	0.7	2003	**[17] gccXfront: exploiting gcc as a front end for program comprehension tools via XML/XSLT** M. Hennessy, ..., and James F. Power 11th IEEE International Workshop on Program Comprehension, 2003. · May 10, 2003 · 15 Citations · Cite	PDF	Show Abstract
2%	0.0	Unknown	**[18] Compiler Verification Meets Cross-Language Linking via Data Abstraction** Peng Wang, ..., and A. Chlipala Journal Not Provided · Date Not Available · 2 Citations · Cite	HTML	No abstract or full text available.
2%	1.6	2015	**[19] Formalizing a Secure Foreign Function Interface** Adriaan Larmuseau and D. Clarke Sep 7, 2015 · 15 Citations · Cite	PDF	No abstract or full text available.
2%	18.5	2017	**[20] Verified low-level programming embedded in F*** Jonathan Protzenko, ..., and Nikhil Swamy Proceedings of the ACM on Programming Languages · Feb 28, 2017 · 151 Citations · Cite	PDF	Show Abstract
2%	11.7	2007	**[21] Operational semantics for multi-language programs** Jacob Matthews and R. Findler ACM Trans. Program. Lang. Syst. · Jan 17, 2007 · 214 Citations · Cite	PDF	Show Abstract
1%	0.5	2019	**[22] Approaching Polyglot Programming: What Can We Learn from Bilingualism Studies?** R. L. Hao and Elena L. Glassman Date Not Available · 3 Citations · Cite	HTML	Show Abstract
1%	0.3	2018	**[23] CAMLroot: revisiting the OCaml FFI** Frédéric Bour ArXiv · Dec 12, 2018 · 2 Citations · Cite	PDF	Show Abstract
1%	3	1999	**[24] StackThreads/MP: integrating futures into calling standards** K. Taura, ..., and A. Yonezawa	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
			May 1, 1999 · 77 Citations · Cite	
1%	3.5	2010	**[25] Jinn: synthesizing dynamic bug detectors for foreign language interfaces** HTML Byeongcheol Lee, ..., and K. McKinley Jun 5, 2010 · 52 Citations · Cite	Show Abstract
1%	0.1	2010	**[26] Fast native function calls for the Babel language interoperability framework** PDF D. Ebner and T. Epperly 2010 11th IEEE/ACM International Conference on Grid Computing · Jul 2, 2010 · 1 Citations · Cite	Show Abstract
1%	0.9	2003	**[27] XOgastan: XML-oriented gcc AST analysis and transformations** PDF G. Antoniol, ..., and Umberto Villano Proceedings Third IEEE International Workshop on Source Code Analysis and Manipulation · Oct 20, 2003 · 19 Citations · Cite	Show Abstract
1%	0.5	2023	**[28] Efficient and Accurate Automatic Python Bindings with cppyy & Cling** PDF B. Kundu, ..., and W. Lavrijsen ArXiv · Apr 5, 2023 · 1 Citations · Cite	Show Abstract
0%	0.0	1996	**[29] Optimizing Procedure Calls for Delayed Non-local Execution Protocol** HTML V. Loia and Michel Quaggetto Software: Practice and Experience · Dec 1, 1996 · 1 Citations · Cite	Show Abstract
0%	1.8	2023	**[30] Semantic Encapsulation using Linking Types** HTML Daniel Patterson, ..., and Amal J. Ahmed Proceedings of the 8th ACM SIGPLAN International Workshop on Type-Driven Development · Aug 30, 2023 · 3 Citations · Cite	Show Abstract
0%	2.7	2022	**[31] COOPER: Testing the Binding Code of Scripting Languages with Cooperative Mutation** PDF Peng Xu, ..., and Purui Su Proceedings 2022 Network and Distributed System Security Symposium · Date Not Available · 9 Citations · Cite	Show Abstract
0%	0.0	2013	**[32] Inferred Interface Glue: Supporting Language Interoperability with Static Analysis** HTML Tristan Ravitch Date Not Available · 0 Citations · Cite	No abstract or full text available.
0%	0.4	2022	**[33] A Foreign Function Interface for Pallene** PDF Gabriel Coutinho De Paula and Roberto Ierusalimschy Proceedings of the XXVI Brazilian Symposium on Programming Languages · Oct 6, 2022 · 1 Citations · Cite	Show Abstract
0%	1.4	2009	**[34] Automatic generation of library bindings using static analysis** PDF Tristan Ravitch, ..., and B. Liblit Jun 15, 2009 · 23 Citations · Cite	Show Abstract
0%	7.7	2010	**[35] A verified compiler for an impure functional language** HTML A. Chlipala Jan 17, 2010 · 118 Citations · Cite	Show Abstract
0%	2.4	2000	**[36] A single intermediate language that supports multiple implementations of exceptions** PDF N. Ramsey and S. P. Jones May 1, 2000 · 61 Citations · Cite	Show Abstract
0%	0.0	2024	**[37] MetaFFI - Multilingual Indirect Interoperability System** PDF Tsvi Cherny-Shahar and A. Yehudai ArXiv · Aug 26, 2024 · 0 Citations · Cite	Show Abstract
0%	8.2	2023	**[38] DimSum: A Decentralized Approach to Multi-language Semantics and Verification** PDF Michael Sammler, ..., and Derek Dreyer Proceedings of the ACM on Programming Languages · Jan 9, 2023 · 19 Citations · Cite	Show Abstract
0%	7.8	2023	**[39] Melocoton: A Program Logic for Verified Interoperability Between OCaml and C** PDF Armaël Guéneau, ..., and Derek Dreyer Proceedings of the ACM on Programming Languages · Oct 16, 2023 · 12 Citations · Cite	Show Abstract
0%	0.3	2021	**[40] Program Obfuscation via ABI Debiasing** HTML David Demicco, ..., and Aravind Prakash Proceedings of the 37th Annual Computer Security Applications Conference · Dec 6, 2021 · 1 Citations · Cite	Show Abstract
0%	0.0	2023	**[41] Challenges and Opportunities in C/C++ Source-To-Source Compilation (Invited Paper)** HTML João Bispo, ..., and L. M. Sousa Date Not Available · 0 Citations · Cite	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	1.4	2006	**[42] Type-safe distributed programming for OCaml** John N. Billings, ..., and Rok Strnisa Sep 16, 2006 · 26 Citations · Cite	HTML	Show Abstract
0%	0.5	2010	**[43] JNI Light: An Operational Model for the Core JNI (Technical Report)** Gang Tan Date Not Available · 8 Citations · Cite	HTML	No abstract or full text available.
0%	0.0	2022	**[44] Software Compatibility Tool Agreement** V. Sochat and Tim Haines ArXiv · Date Not Available · 0 Citations · Cite	PDF	Show Abstract
0%	3.4	1999	**[45] Calling hell from heaven and heaven from hell** Sigbjørn Finne, ..., and S. Jones Sep 1, 1999 · 86 Citations · Cite	PDF	Show Abstract
0%	5.5	2005	**[46] Checking type safety of foreign function calls** Michael Furr and J. Foster Jun 12, 2005 · 109 Citations · Cite	PDF	Show Abstract
0%	1.2	2008	**[47] Efficient automated marshaling of C++ data structures for MPI applications** Wesley Tansey and E. Tilevich 2008 IEEE International Symposium on Parallel and Distributed Processing · Apr 14, 2008 · 20 Citations · Cite	PDF	Show Abstract
0%	0.0	2022	**[48] Binary-level Software Compatibility Tool Agreement** V. Sochat and Tim Haines ArXiv · Dec 6, 2022 · 0 Citations · Cite	PDF	Show Abstract
0%	0.0	2016	**[49] Serialization of foreign types with SKilL** C. Weisser Date Not Available · 0 Citations · Cite	HTML	Show Abstract
0%	0.0	2018	**[50] Sulong, and thanks for all the fish** Manuel Rigger, ..., and H. Mössenböck Companion Proceedings of the 2nd International Conference on the Art, Science, and Engineering of Programming · Apr 9, 2018 · 0 Citations · Cite	HTML	Show Abstract
0%	26.4	2002	**[51] CCured: type-safe retrofitting of legacy code** G. Necula, ..., and Westley Weimer Date Not Available · 616 Citations · Cite	HTML	Show Abstract
0%	0.7	2021	**[52] A Meta-level Approach for Multilingual Taint Analysis** D. Lyons and Dino Becaj Date Not Available · 3 Citations · Cite	PDF	Show Abstract
0%	3.5	2008	**[53] Finding bugs in java native interface programs** Goh Kondoh and Tamiya Onodera Jul 20, 2008 · 58 Citations · Cite	PDF	Show Abstract
0%	0.0	2015	**[54] Generating declarations needed by LuaJIT compiler for binding Lua and C** Violeta Vukobrat, ..., and M. Popovic 2015 23rd Telecommunications Forum Telfor (TELFOR) · Nov 1, 2015 · 0 Citations · Cite	HTML	Show Abstract
0%	0.0	2015	**[55] Automatic array property detection via static analysis** Alisa J. Maas Companion Proceedings of the 2015 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity · Oct 25, 2015 · 0 Citations · Cite	HTML	Show Abstract
0%	2.4	1998	**[56] H/Direct: a binary foreign language interface for Haskell** Sigbjørn Finne, ..., and S. Jones Sep 29, 1998 · 63 Citations · Cite	PDF	Show Abstract
0%	0.8	2021	**[57] Multi-stage programming with generative and analytical macros** Nicolas Stucki, ..., and Martin Odersky Proceedings of the 20th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences · Oct 17, 2021 · 3 Citations · Cite	HTML	Show Abstract
0%	0.7	1999	**[58] C → HASKELL, or Yet Another Interfacing Tool** M. Chakravarty Sep 7, 1999 · 19 Citations · Cite	HTML	No abstract or full text available.
0%	2.7	2019	**[59] ClangJIT: Enhancing C++ with Just-in-Time Compilation** H. Finkel, ..., and D. Richards 2019 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC) · Apr 18, 2019 · 16 Citations · Cite	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	0.0	1992	[60] Sleeping with the enemy: Lisp and C in a large, profitable real-time application J. Hodgkinson Aug 10, 1992 · 0 Citations · Cite	HTML	Show Abstract
0%	4.4	2007	[61] Jeannie: granting java native interface developers their wishes Martin Hirzel and R. Grimm Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications · Oct 21, 2007 · 77 Citations · Cite	HTML	Show Abstract
0%	0.3	2001	[62] Using software component generators to construct a meta-weaver framework J. Gray Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001 · Jul 1, 2001 · 8 Citations · Cite	HTML	Show Abstract
0%	0.6	2023	[63] Targeted Static Analysis for OCaml C Stubs: eliminating gremlins from the code Edwin Török ArXiv · Jul 27, 2023 · 1 Citations · Cite	PDF	Show Abstract
0%	0.7	2016	[64] Array length inference for C library bindings Alisa J. Maas, ..., and B. Liblit 2016 31st IEEE/ACM International Conference on Automated Software Engineering (ASE) · Aug 25, 2016 · 6 Citations · Cite	HTML	Show Abstract
0%	3.7	2009	[65] Finding bugs in exceptional situations of JNI programs Siliang Li and Gang Tan Nov 9, 2009 · 58 Citations · Cite	HTML	Show Abstract
0%	21.9	2002	[66] Contracts for higher-order functions R. Findler and M. Felleisen Oct 4, 2002 · 494 Citations · Cite	HTML	Show Abstract
0%	3	2005	[67] Fine-grained interoperability through mirrors and contracts Kathryn E. Gray, ..., and M. Flatt Oct 17, 2005 · 58 Citations · Cite	HTML	Show Abstract
0%	0.0	2018	[68] Reasoning About Foreign Function Interfaces: Blame and Nondeterministic Formal Semantics Alexi Turcotte Aug 31, 2018 · 0 Citations · Cite	HTML	No abstract or full text available.
0%	1.9	2024	[69] KaMPIng: Flexible and (Near) Zero-Overhead C++ Bindings for MPI Demian Hespe, ..., and Tim Niklas Uhl SC24: International Conference for High Performance Computing, Networking, Storage and Analysis · Apr 8, 2024 · 2 Citations · Cite	PDF	Show Abstract
0%	3	2016	[70] High-Performance Python-C++ Bindings with PyPy and Cling W. Lavrijsen and Aditi Dutta 2016 6th Workshop on Python for High-Performance and Scientific Computing (PyHPC) · Nov 13, 2016 · 25 Citations · Cite	HTML	Show Abstract
0%	2.9	2013	[71] Large-Scale Automated Refactoring Using ClangMR Hyrum K. Wright, ..., and Zhanyong Wan 2013 IEEE International Conference on Software Maintenance · Sep 22, 2013 · 34 Citations · Cite	HTML	Show Abstract
0%	1.3	2018	[72] Xolotl: An Intuitive and Approachable Neuron and Network Simulator for Research and Teaching S. Gorur-Shandilya, ..., and E. Marder Frontiers in Neuroinformatics · Aug 18, 2018 · 9 Citations · Cite	PDF	Show Abstract
0%	0.0	2014	[73] Improving robustness and computational efficiency using modern C++ M. Paterno, ..., and C. Green Journal of Physics: Conference Series · Jun 11, 2014 · 0 Citations · Cite	PDF	Show Abstract
0%	6.2	2011	[74] An equivalence-preserving CPS translation via multi-language semantics Amal J. Ahmed and Matthias Blume Proceedings of the 16th ACM SIGPLAN international conference on Functional programming · Sep 19, 2011 · 84 Citations · Cite	HTML	Show Abstract
0%	2.2	2017	[75] LMS-Verify: abstraction without regret for verified systems programming Nada Amin and Tiark Rompf Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages · Jan 1, 2017 · 18 Citations · Cite	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
0%	3.1	2001	**[76] No-Longer-Foreign: Teaching an ML compiler to speak C "natively"** PDF Matthias Blume Nov 1, 2001 · 74 Citations · Cite	No abstract or full text available.
0%	0.0	2017	**[77] Designing a pi-based Programming Language in the .NET framework: CLR interoperability from the Programmer's point of view** PDF M. Mazzara ArXiv · Mar 15, 2017 · 0 Citations · Cite	Show Abstract
0%	3.6	2012	**[78] Cling – The New Interactive Interpreter for ROOT 6** PDF V. Vasilev, ..., and P. Russo Journal of Physics: Conference Series · Dec 13, 2012 · 45 Citations · Cite	Show Abstract
0%	2	2019	**[79] Towards polyglot adapters for the GraalVM** HTML Fabio Niephaus, ..., and R. Hirschfeld Companion Proceedings of the 3rd International Conference on the Art, Science, and Engineering of Programming · Apr 1, 2019 · 12 Citations · Cite	Show Abstract
0%	3.9	2020	**[80] Identifying Java calls in native code via binary scanning** HTML G. Fourtounis, ..., and Y. Smaragdakis Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis · Jul 7, 2020 · 19 Citations · Cite	Show Abstract
0%	0.0	2024	**[81] Staged Compilation with Module Functors** PDF TSUNG-JU Chiang, ..., and Ningning Xie Proc. ACM Program. Lang. · Aug 15, 2024 · 0 Citations · Cite	Show Abstract
0%	3.4	2020	**[82] The Python/C API: Evolution, Usage Statistics, and Bug Patterns** HTML Mingzhe Hu and Yu Zhang 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) · Feb 1, 2020 · 18 Citations · Cite	Show Abstract
0%	5.9	2015	**[83] Mutation-Based Fault Localization for Real-World Multilingual Programs (T)** HTML Shin Hong, ..., and Moonzoo Kim 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE) · Nov 9, 2015 · 56 Citations · Cite	Show Abstract
0%	0.1	2012	**[84] GoCxx: a tool to easily leverage C++ legacy code for multicore-friendly Go libraries and frameworks** PDF S. Binet Journal of Physics: Conference Series · Dec 13, 2012 · 1 Citations · Cite	Show Abstract
0%	2.2	2007	**[85] A Linear Language with Locations** HTML Amal J. Ahmed and M. Fluet Journal Not Provided · Date Not Available · 41 Citations · Cite	No abstract or full text available.
0%	0.0	2006	**[86] Avoiding two-level systems: Using a textual environment to address cross-cutting concerns** PDF D. Greaves Date Not Available · 0 Citations · Cite	Show Abstract
0%	0.0	2023	**[87] A functional scripting interface to an object oriented C++ library** PDF Markus Kloimwieder and Christoph Gadermaier ArXiv · Dec 17, 2023 · 0 Citations · Cite	Show Abstract
0%	7.6	2016	**[88] Fully abstract compilation via universal embedding** PDF Max S. New, ..., and Amal J. Ahmed Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming · Sep 4, 2016 · 66 Citations · Cite	Show Abstract
0%	1.5	2002	**[89] Program annotation in XML: a parse-tree based approach** PDF James F. Power and B. Malloy Ninth Working Conference on Reverse Engineering, 2002. Proceedings. · Oct 29, 2002 · 33 Citations · Cite	Show Abstract
0%	5.1	2023	**[90] Detecting C++ Compiler Front-End Bugs via Grammar Mutation and Differential Testing** PDF Haoxin Tu, ..., and Lingxiao Jiang IEEE Transactions on Reliability · Mar 1, 2023 · 11 Citations · Cite	Show Abstract
0%	1.8	2021	**[91] Lifting C semantics for dataflow optimization** PDF A. Calotoiu, ..., and T. Hoefler Proceedings of the 36th ACM International Conference on Supercomputing · Dec 22, 2021 · 6 Citations · Cite	Show Abstract
0%	3.1	2010	**[92] Stateful Contracts for Affine Types** PDF Jesse A. Tov and Riccardo Pucella Mar 20, 2010 · 47 Citations · Cite	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	1.8	1999	**[93] Interlanguage working without tears: blending SML with Java** Nick Benton and A. Kennedy Sep 1, 1999 · 47 Citations · Cite	PDF	Show Abstract
0%	6.9	2020	**[94] Broadening Horizons of Multilingual Static Analysis: Semantic Summary Extraction from C Code for JNI Program Analysis** Sungho Lee, ..., and Sukyoung Ryu 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE) · Sep 1, 2020 · 32 Citations · Cite	HTML	Show Abstract
0%	0.6	2011	**[95] Escaping the Bonds of the Legacy: Step-Wise Migration to a Type-Safe Language in Safety-Critical Embedded Systems** Michael Stilkerich, ..., and Daniel Lohmann 2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing · Mar 28, 2011 · 8 Citations · Cite	PDF	Show Abstract
0%	0.0	2013	**[96] Extending programming language to support object orientation in legacy systems** Heman Mehta, ..., and D. Janakiram Comput. Sci. Inf. Syst. · Date Not Available · 0 Citations · Cite	PDF	Show Abstract
0%	0.0	2014	**[97] Correct-by-Construction Program Derivation from Specifications to Assembly Language** No author found Journal Not Provided · Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
0%	0.0	Unknown	**[98] C++ Template Metaprogramming with Embedded Haskell** Z. Porkoláb and ·Abel Sinkovics Journal Not Provided · Date Not Available · 6 Citations · Cite	HTML	No abstract or full text available.
0%	6.8	2019	**[99] The next 700 compiler correctness theorems (functional pearl)** Daniel Patterson and Amal J. Ahmed Proceedings of the ACM on Programming Languages · Jul 26, 2019 · 39 Citations · Cite	PDF	Show Abstract
0%	0.1	2007	**[100] Experience with SC: transformation-based implementation of various extensions to C** Tasuku Hiraishi, ..., and T. Yuasa Apr 1, 2007 · 2 Citations · Cite	HTML	Show Abstract
0%	0.0	2023	**[101] SML#: Toward the Ideal Interoperability between Languages and Systems (Keynote)** Katsuhiro Ueno Companion Proceedings of the 7th International Conference on the Art, Science, and Engineering of Programming · Mar 13, 2023 · 0 Citations · Cite	HTML	Show Abstract
0%	0.2	2013	**[102] Simplifying the analysis of c++ programs** B. Stroustrup and Yuriy Solodkyy Date Not Available · 2 Citations · Cite	HTML	No abstract or full text available.
0%	6	2021	**[103] A Multilanguage Static Analysis of Python Programs with Native C Extensions** Raphaël Monat, ..., and A. Miné Date Not Available · 26 Citations · Cite	HTML	No abstract or full text available.
0%	2.3	2018	**[104] OP2-Clang: A Source-to-Source Translator Using Clang/LLVM LibTooling** Gábor Dániel Balogh, ..., and C. Bertolli 2018 IEEE/ACM 5th Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC) · Nov 1, 2018 · 15 Citations · Cite	PDF	Show Abstract
0%	0.2	2015	**[105] A Tree-Based Approach to Support Refactoring in Multi-Language Software Applications** Hagen Schink, ..., and W. Fenske Journal Not Provided · Date Not Available · 2 Citations · Cite	HTML	No abstract or full text available.
0%	0.7	2016	**[106] Declarative Foreign Function Binding Through Generic Programming** J. Yallop, ..., and Anil Madhavapeddy Mar 4, 2016 · 6 Citations · Cite	HTML	No abstract or full text available.
0%	0.8	2017	**[107] Analysis of Include Dependencies in C++ Source Code** Bence Babati and Norbert Pataki Sep 24, 2017 · 6 Citations · Cite	PDF	Show Abstract
0%	0.0	2008	**[108] Implementing a Portable Foreign Function Interface through Dynamic C Code Generation** No author found Journal Not Provided · Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
0%	38	2022	**[109] Call-by-push-value** P. Levy ACM SIGLOG News · Apr 1, 2022 · 117 Citations · Cite	HTML	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
0%	3.8	2019	**[110] Anti-patterns for multi-language systems** HTML Mouna Abidi, ..., and Yann-Gaël Guéhéneuc Proceedings of the 24th European Conference on Pattern Languages of Programs · Jul 3, 2019 · 22 Citations · Cite	Show Abstract
0%	2	2012	**[111] Dependent interoperability** HTML Peter-Michael Osera, ..., and Steve Zdancewic Jan 24, 2012 · 27 Citations · Cite	Show Abstract
0%	0.0	2020	**[112] Fluid quotes: metaprogramming across abstraction boundaries with dependent types** PDF Shadaj Laddad and Koushik Sen Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences · Nov 15, 2020 · 0 Citations · Cite	Show Abstract
0%	0.3	2021	**[113] Single-state state machines in model-driven software engineering: an exploratory study** PDF Nan Yang, ..., and Alexander Serebrenik Empirical Software Engineering · Sep 10, 2021 · 1 Citations · Cite	Show Abstract
0%	2	2019	**[114] Generating a fluent API with syntax checking from an LR grammar** PDF Tetsuro Yamazaki, ..., and S. Chiba Proceedings of the ACM on Programming Languages · Oct 10, 2019 · 11 Citations · Cite	Show Abstract
0%	0.0	2019	**[115] The Next 700 Compiler Correctness Theorems (Functional Pearl)** HTML Daniel Patterson and Amal Ahmed Journal Not Provided · Date Not Available · 0 Citations · Cite	No abstract or full text available.
0%	0.7	1996	**[116] A Portable C Interface for Standard ML of New Jersey** HTML L. Huelsbergen Date Not Available · 21 Citations · Cite	No abstract or full text available.
0%	0.2	2021	**[117] A Scheme Foreign Function Interface to JavaScript Based on an Infix Extension** HTML M. Bélanger and Marc Feeley Journal Not Provided · Date Not Available · 1 Citations · Cite	No abstract or full text available.
0%	0.0	2024	**[118] Improving the C++ Experience with Transpilers** HTML Stephen E Hellings 2024 IEEE Integrated STEM Education Conference (ISEC) · Mar 9, 2024 · 0 Citations · Cite	Show Abstract
0%	0.0	2012	**[119] On AOP techniques for C++-based HW/SW component implementation** HTML T. Mück and A. A. Fröhlich Date Not Available · 0 Citations · Cite	Show Abstract
0%	1.4	2005	**[120] High-level views on low-level representations** PDF Iavor S. Diatchki, ..., and Rebekah Leslie Sep 28, 2005 · 27 Citations · Cite	Show Abstract
0%	0.0	2020	**[121] Intermediate Representation Using Graph Visualization Software** PDF E. O. Aliyu, ..., and B. Ojokoh Journal of Software Engineering and Applications · May 9, 2020 · 0 Citations · Cite	Show Abstract
0%	0.2	2009	**[122] Language-shifting objects from Java to Smalltalk: an exploration using JavaConnect** HTML J. Brichau and Coen De Roover Aug 31, 2009 · 3 Citations · Cite	Show Abstract
0%	0.0	2024	**[123] Redesigning FFI calls in Pharo: exploiting the baseline JIT for more performance and low maintenance** HTML Juan Ignacio Bianchi and Guillermo Polito International Workshop on Smalltalk Technologies · Date Not Available · 0 Citations · Cite	No abstract or full text available.
0%	3.3	2017	**[124] ABCpy: A High-Performance Computing Perspective to Approximate Bayesian Computation** PDF Ritabrata Dutta, ..., and A. Mira J. Stat. Softw. · Nov 13, 2017 · 25 Citations · Cite	Show Abstract
0%	1.3	2019	**[125] On the Multi-Language Construction** HTML Samuele Buro and Isabella Mastroeni Apr 8, 2019 · 8 Citations · Cite	No abstract or full text available.
0%	0.3	2015	**[126] Making fortran legacy code more functional: using the BGS* geomagnetic field modelling system as an example** HTML Hans-Nikolai Vießmann, ..., and Simon Flower Sep 14, 2015 · 3 Citations · Cite	Show Abstract
0%	1.7	2005	**[127] A Foreign-Function Interface Generator for occam-pi** HTML Damian J. Dimmich and Christian L. Jacobsen Sep 1, 2005 · 33 Citations · Cite	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	1	1994	[128] Implementing signatures for C++ Gerald Baumgartner and V. Russo Apr 11, 1994 · 30 Citations · Cite	PDF	Show Abstract
0%	0.0	2024	[129] Tail Modulo Cons, OCaml, and Relational Separation Logic Clément Allain, ..., and Gabriel Scherer Proc. ACM Program. Lang. · Nov 28, 2024 · 0 Citations · Cite	PDF	Show Abstract
0%	3.8	2023	[130] Leveraging Modern C++ in High-Level Synthesis Sakari Lahti, ..., and Timo D. Hämäläinen IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems · Apr 1, 2023 · 8 Citations · Cite	PDF	Show Abstract
0%	0.1	1997	[131] Efficient implementation of portable C*-like data-parallel library in C++ Motohiko Matsuda, ..., and Y. Ishikawa Proceedings. Advances in Parallel and Distributed Computing · Mar 19, 1997 · 2 Citations · Cite	HTML	Show Abstract
0%	0.0	2015	[132] A High-Level Model for an Assembly Language Attacker by Means of Reflection Adriaan Larmuseau, ..., and D. Clarke Nov 4, 2015 · 0 Citations · Cite	HTML	No abstract or full text available.
0%	0.0	2015	[133] Tool for detecting standardwise differences in C++ legacy code Tibor Brunner, ..., and Z. Porkoláb 2015 IEEE 13th International Scientific Conference on Informatics · Nov 1, 2015 · 0 Citations · Cite	HTML	Show Abstract
0%	0.0	2002	[134] Speaking C++ as a native B. Stroustrup Jan 29, 2002 · 1 Citations · Cite	HTML	Show Abstract
0%	0.2	2018	[135] Sampling-based Tube Following for Redundant, Planar Robotic Manipulators J. D. Maeyer, ..., and E. Demeester 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA) · Sep 1, 2018 · 1 Citations · Cite	HTML	Show Abstract
0%	3.5	2022	[136] sympy2c: from symbolic expressions to fast C/C++ functions and ODE solvers in Python Uwe Schmitt, ..., and A. Réfrégier ArXiv · Mar 22, 2022 · 11 Citations · Cite	PDF	Show Abstract
0%	0.5	2021	[137] Towards A Polyglot Framework for Factorized ML David Justo, ..., and Arun Kumar Proc. VLDB Endow. · Jul 1, 2021 · 2 Citations · Cite	HTML	Show Abstract
0%	0.0	2024	[138] Fast Template-Based Code Generation for MLIR Florian Drescher and Alexis Engelke Proceedings of the 33rd ACM SIGPLAN International Conference on Compiler Construction · Feb 17, 2024 · 0 Citations · Cite	HTML	Show Abstract
0%	1.7	2004	[139] Adventures in interoperability: the SML.NET experience Nick Benton, ..., and Claudio V. Russo Aug 24, 2004 · 35 Citations · Cite	HTML	Show Abstract
0%	0.4	2015	[140] Foreign exchange at low, low rates a lightweight FFI for web-targeting Haskell dialects A. Ekblad Sep 14, 2015 · 4 Citations · Cite	PDF	Show Abstract
0%	0.9	2018	[141] Mapping to Bits: Efficiently Detecting Type Confusion Errors Chengbin Pang, ..., and Shanqing Guo Proceedings of the 34th Annual Computer Security Applications Conference · Dec 3, 2018 · 6 Citations · Cite	HTML	Show Abstract
0%	0.5	2012	[142] JATO: Native Code Atomicity for Java Siliang Li, ..., and Gang Tan Dec 11, 2012 · 6 Citations · Cite	HTML	No abstract or full text available.
0%	0.4	2005	[143] g4re: Harnessing GCC to Reverse Engineer C++ Applications Nicholas A. Kraft, ..., and James F. Power Date Not Available · 9 Citations · Cite	HTML	No abstract or full text available.
0%	0.3	2016	[144] The Nifty way to call hell from heaven A. Löscher and Konstantinos Sagonas Proceedings of the 15th International Workshop on Erlang · Sep 23, 2016 · 3 Citations · Cite	HTML	Show Abstract
0%	0.7	2009	[145] Toward foundations for type-reflective metaprogramming Ronald Garcia and A. Lumsdaine	HTML	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
			Oct 4, 2009 · 11 Citations · Cite	
0%	0.5	2021	**[146] Efficient Platform Migration of a Mainframe Legacy System Using Custom Transpilation** HTML Markus Schnappinger and Jonathan Streit 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME) · Sep 1, 2021 · 2 Citations · Cite	Show Abstract
0%	0.1	2018	**[147] Reasonably Mixing a Functional Language with Assembly** * HTML Daniel Patterson and Christos Dimoulas Journal Not Provided · Date Not Available · 1 Citations · Cite	No abstract or full text available.
0%	15.5	1993	**[148] Types for Dyadic Interaction** HTML Kohei Honda Aug 23, 1993 · 490 Citations · Cite	No abstract or full text available.
0%	0.0	2014	**[149] Применение метода двухфазной компиляции на основе LLVM для распространения приложений с использованием облачного хранилища** PDF С. С. Гайсарян, ..., and Севак Сеникович Саргсян Date Not Available · 0 Citations · Cite	Show Abstract
0%	0.0	2023	**[150] Towards Reliable Memory Management for Python Native Extensions** HTML Joannah Nanjekye, ..., and Aleksandar Micic Proceedings of the 18th ACM International Workshop on Implementation, Compilation, Optimization of OO Languages, Programs and Systems · Jul 17, 2023 · 0 Citations · Cite	Show Abstract
0%	8.7	1998	**[151] Language Primitives and Type Discipline for Structured Communication-Based Programming** PDF Kohei Honda, ..., and Makoto Kubo Mar 28, 1998 · 237 Citations · Cite	No abstract or full text available.
0%	0.6	2017	**[152] The Computer for the 21st Century: Security & Privacy Challenges after 25 Years** HTML L. Oliveira, ..., and Jie Liu 2017 26th International Conference on Computer Communication and Networks (ICCCN) · Jul 1, 2017 · 5 Citations · Cite	Show Abstract
0%	0.1	2014	**[153] Structure-Preserving Compilation: Efficient Integration of Functional DSLs into Legacy Systems** HTML J. Kranz and A. Simon Sep 8, 2014 · 1 Citations · Cite	Show Abstract
0%	4.5	2020	**[154] A history of Clojure** PDF R. Hickey Proceedings of the ACM on Programming Languages · Jun 12, 2020 · 22 Citations · Cite	Show Abstract
0%	2.8	2002	**[155] Accurate garbage collection in an uncooperative environment** HTML F. Henderson Jun 20, 2002 · 63 Citations · Cite	Show Abstract
0%	0.0	2002	**[156] Porting legacy interpretive bytecode to the CLR** HTML Jeremy Singer Jun 13, 2002 · 0 Citations · Cite	Show Abstract
0%	0.0	2023	**[157] CGORewritter: A better way to use C library in G** HTML Boyao Ding, ..., and Qingwei Li 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER) · Mar 1, 2023 · 0 Citations · Cite	Show Abstract
0%	0.0	2025	**[158] Annotation-guided AoS-to-SoA conversions and GPU offloading with data views in C++** PDF Pawel K. Radtke and Tobias Weinzierl ArXiv · Feb 23, 2025 · 0 Citations · Cite	Show Abstract
0%	1	2020	**[159] User-defined interface mappings for the GraalVM** HTML Alexander Riese, ..., and R. Hirschfeld Companion Proceedings of the 4th International Conference on Art, Science, and Engineering of Programming · Mar 23, 2020 · 5 Citations · Cite	Show Abstract
0%	13.5	2014	**[160] Wysteria: A Programming Language for Generic, Mixed-Mode Multiparty Computations** PDF Aseem Rastogi, ..., and M. Hicks 2014 IEEE Symposium on Security and Privacy · May 18, 2014 · 148 Citations · Cite	Show Abstract
0%	1.1	2006	**[161] A Scalable Mixed-Level Approach to Dynamic Analysis of C and C++ Programs** HTML Philip J. Guo Date Not Available · 21 Citations · Cite	No abstract or full text available.
0%	4.6	2017	**[162] FunTAL: reasonably mixing a functional language with assembly** PDF Daniel Patterson, ..., and Amal J. Ahmed	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
			Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation · Jun 14, 2017 · 36 Citations · Cite		
0%	12.5	1998	[163] LANGUAGE PRIMITIVES AND TYPE DISCIPLINE FOR STRUCTURED COMMUNICATION-BASED PROGRAMMING V. Vasconcelos and Makoto Kubo Journal Not Provided · Date Not Available · 342 Citations · Cite	HTML	No abstract or full text available.
0%	12.6	1994	[164] An Interaction-based Language and its Typing System K. Takeuchi, ..., and Makoto Kubo Jul 4, 1994 · 387 Citations · Cite	HTML	No abstract or full text available.
0%	0.0	1989	[165] An improved tool for automated compiler construction M. Parkes Feb 1, 1989 · 1 Citations · Cite	HTML	No abstract or full text available.
0%	0.2	2019	[166] Meta C++: an extension layer for multi-stage generative metaprogramming Yannis Lilis and Anthony Savidis J. Object Technol. · Date Not Available · 1 Citations · Cite	PDF	Show Abstract
0%	1.9	2018	[167] The computer for the 21st century: present security & privacy challenges L. Oliveira, ..., and Jie Liu Journal of Internet Services and Applications · Dec 1, 2018 · 12 Citations · Cite	PDF	Show Abstract
0%	0.0	2009	[168] A-030 Polymorphic Foreign Function Interface between ML and C Katsuhiro Ueno and A. Ohori Aug 20, 2009 · 0 Citations · Cite	HTML	No abstract or full text available.
0%	13.8	2005	[169] Scalable component abstractions Martin Odersky and Matthias Zenger Oct 17, 2005 · 269 Citations · Cite	PDF	Show Abstract
0%	0.2	1987	[170] Foreign functions and common Lisp H. Sexton ACM Sigplan Lisp Pointers · Dec 1, 1987 · 9 Citations · Cite	PDF	Show Abstract
0%	3	1997	[171] Annotation-directed run-time specialization in C B. Grant, ..., and S. Eggers Dec 1, 1997 · 82 Citations · Cite	PDF	Show Abstract
0%	1	2018	[172] Analytics with smart arrays: adaptive and efficient language-independent data Iraklis Psaroudakis, ..., and T. Harris Proceedings of the Thirteenth EuroSys Conference · Apr 23, 2018 · 7 Citations · Cite	HTML	Show Abstract
0%	0.0	1997	[173] Automatic generation of bridging code for accessing C++ from Java A. Schade Proceedings. Technology of Object-Oriented Languages and Systems, TOOLS 25 (Cat. No.97TB100239) · Nov 24, 1997 · 0 Citations · Cite	HTML	Show Abstract
0%	0.3	2021	[174] Evaluating Optimizations for a High-Level Language Leonardo Kaplan and R. Ierusalimschy Proceedings of the 25th Brazilian Symposium on Programming Languages · Sep 27, 2021 · 1 Citations · Cite	HTML	Show Abstract
0%	6.1	2015	[175] High-performance cross-language interoperability in a multi-language runtime Matthias Grimmer, ..., and H. Mössenböck Proceedings of the 11th Symposium on Dynamic Languages · Oct 21, 2015 · 58 Citations · Cite	HTML	Show Abstract
0%	0.0	2006	[176] Towards Customizable Pedagogic Programming Languages Kathryn E. Gray Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
0%	0.1	2016	[177] Language-based Techniques for Practical and Trustworthy Secure Multi-party Computations Aseem Rastogi Date Not Available · 1 Citations · Cite	HTML	Show Abstract
0%	0.1	1996	[178] Extensions to the Franz, Inc.'s Allegro Common Lisp foreign function interface J. Keane Date Not Available · 3 Citations · Cite	HTML	Show Abstract
0%	0.0	2002	[179] An interoperable calculus for external object access A. Ohori and Kiyoshi Yamatodani Oct 4, 2002 · 0 Citations · Cite	HTML	Show Abstract
0%	6.4	2025	[180] A Verified Foreign Function Interface between Coq and C	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
			Joomy Korkut, ..., and Andrew W. Appel Proc. ACM Program. Lang. · Jan 7, 2025 · 2 Citations · Cite	
0%	0.6	2023	[181] Towards Multi-Language Static Code Analysis HTML Sanaa Siddiqui, ..., and Kumar Madhukar 2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW) · Oct 9, 2023 · 1 Citations · Cite	Show Abstract
0%	1.7	2020	[182] Towards the Definition of Patterns and Code Smells for Multi-language Systems HTML Mouna Abidi and Foutse Khomh Proceedings of the European Conference on Pattern Languages of Programs 2020 · Jul 1, 2020 · 8 Citations · Cite	Show Abstract
0%	0.1	2008	[183] A novel flow-sensitive type and effect analysis for securing C code HTML Syrine Tlili and M. Debbabi 2008 IEEE/ACS International Conference on Computer Systems and Applications · Mar 31, 2008 · 2 Citations · Cite	Show Abstract
0%	2.2	2003	[184] The haskell 98 foreign function interface 1 HTML M. Chakravarty, ..., and Michael Weber Date Not Available · 50 Citations · Cite	No abstract or full text available.
0%	6.3	2019	[185] A verified, efficient embedding of a verifiable assembly language PDF Aymeric Fromherz, ..., and Nikhil Swamy Proceedings of the ACM on Programming Languages · Jan 2, 2019 · 40 Citations · Cite	Show Abstract
0%	0.0	2004	[186] C++: an evolving language HTML S.P. Levitt 2004 IEEE Africon. 7th Africon Conference in Africa (IEEE Cat. No.04CH37590) · Sep 15, 2004 · 1 Citations · Cite	Show Abstract
0%	0.6	2021	[187] Fast Machine Learning in Data Science with a Comprehensive Data Summarization HTML Sikder Tahsin Al-Amin and C. Ordonez 2021 IEEE International Conference on Big Data (Big Data) · Dec 15, 2021 · 2 Citations · Cite	Show Abstract
0%	0.3	2008	[188] The Layers of Larceny ' s Foreign Function Interface HTML Felix S. Klock Journal Not Provided · Date Not Available · 6 Citations · Cite	No abstract or full text available.
0%	3.7	2019	[189] Towards certified separate compilation for concurrent programs HTML Hanru Jiang, ..., and Xinyu Feng Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation · Jun 8, 2019 · 22 Citations · Cite	Show Abstract
0%	10.4	2021	[190] JuCify: A Step Towards Android Code Unification for Enhanced Static Analysis PDF Jordan Samhi, ..., and Jacques Klein 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE) · Dec 20, 2021 · 35 Citations · Cite	Show Abstract
0%	0.0	2019	[191] Instructed late binding PDF S. H. Haeri and P. Keir Proceedings of the 23rd Pan-Hellenic Conference on Informatics · Nov 28, 2019 · 0 Citations · Cite	Show Abstract
0%	0.4	2010	[192] Clang and Coccinelle: Synergising program analysis tools for CERT C Secure Coding Standard certification HTML Mads Chr. Olesen, ..., and Nicolas Palix Electron. Commun. Eur. Assoc. Softw. Sci. Technol. · Dec 10, 2010 · 6 Citations · Cite	Show Abstract
0%	0.7	2007	[193] High-Level Abstractions for Low-Level Programming HTML Iavor S. Diatchki Date Not Available · 13 Citations · Cite	No abstract or full text available.
0%	4.6	2014	[194] Checking correctness of TypeScript interfaces for JavaScript libraries PDF Asger Feldthaus and Anders Møller ACM SIGPLAN Notices · Oct 15, 2014 · 48 Citations · Cite	Show Abstract
0%	0.1	2009	[195] Synergies in scientific computing by combining multi-paradigmatic languages for high-performance applications HTML P. Schwaha, ..., and S. Selberherr International Journal of Parallel, Emergent and Distributed Systems · Nov 30, 2009 · 1 Citations · Cite	Show Abstract
0%	0.0	2012	[196] Language Interoperability Safe C++ HTML Peter-Michael Osera Journal Not Provided · Date Not Available · 0 Citations · Cite	No abstract or full text available.
0%	0.1	2011	[197] Language and tool support for multilingual programs HTML	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
			Byeong-Choon Lee Aug 1, 2011 · 1 Citations · Cite		
0%	9.4	1994	**[198] Efficient detection of all pointer and array access errors** Todd M. Austin, ..., and G. Sohi Jun 1, 1994 · 292 Citations · Cite	PDF	Show Abstract
0%	0.1	2012	**[199] Twig: a configurable domain-specific language** A. Malony and G. Hulette Date Not Available · 1 Citations · Cite	HTML	No abstract or full text available.
0%	3	2025	**[200] type++: Prohibiting Type Confusion with Inline Type Information** Nicolas Badoux, ..., and Mathias Payer Proceedings 2025 Network and Distributed System Security Symposium · Date Not Available · 1 Citations · Cite	HTML	Show Abstract
0%	9.2	2017	**[201] Everest: Towards a Verified, Drop-in Replacement of HTTPS** K. Bhargavan, ..., and J. Zinzindohoué Apr 30, 2017 · 74 Citations · Cite	PDF	Show Abstract
0%	2	2019	**[202] Under Control: Compositionally Correct Closure Conversion with Mutable State** P. Mates, ..., and Amal J. Ahmed Proceedings of the 21st International Symposium on Principles and Practice of Declarative Programming · Oct 7, 2019 · 11 Citations · Cite	HTML	Show Abstract
0%	0.9	1992	**[203] Designing programming languages for analyzability: a fresh look at pointer data structures** L. Hendren and G. Gao Proceedings of the 1992 International Conference on Computer Languages · Apr 20, 1992 · 30 Citations · Cite	HTML	Show Abstract
0%	0.8	2014	**[204] Benzo: Reflective Glue for Low-level Programming** Camillo Bruni, ..., and Guido Chari Aug 19, 2014 · 9 Citations · Cite	HTML	No abstract or full text available.
0%	1.4	2021	**[205] PyGuard: Finding and Understanding Vulnerabilities in Python Virtual Machines** Chengman Jiang, ..., and Zhizhong Pan 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE) · Oct 1, 2021 · 5 Citations · Cite	HTML	Show Abstract
0%	1.6	2001	**[206] Meta-compilation for C++** Edward D. Willink Date Not Available · 40 Citations · Cite	HTML	No abstract or full text available.
0%	0.0	2008	**[207] Generating type-safe application-specific foreign interfaces** John H. Reppy and Chunyan Song Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
0%	0.9	1999	**[208] Safe and Principled Language Interoperation** Valery Trifonov and Zhong Shao Mar 22, 1999 · 24 Citations · Cite	PDF	No abstract or full text available.
0%	1.5	2007	**[209] State of the Union: Type Inference Via Craig Interpolation** Ranjit Jhala, ..., and Ru-Gang Xu Mar 24, 2007 · 28 Citations · Cite	PDF	No abstract or full text available.
0%	2.1	2020	**[210] On the Impact of Multi-language Development in Machine Learning Frameworks** Manel Grichi, ..., and Bram Adams 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME) · Sep 1, 2020 · 10 Citations · Cite	HTML	Show Abstract
0%	0.7	2023	**[211] Crossover: Towards Compiler-Enabled COBOL-C Interoperability** Mart van Assen, ..., and Vadim Zaytsev Proceedings of the 22nd ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences · Oct 22, 2023 · 1 Citations · Cite	PDF	Show Abstract
0%	3.6	2019	**[212] Gradual type theory** Max S. New, ..., and Amal J. Ahmed Proceedings of the ACM on Programming Languages · Jan 2, 2019 · 23 Citations · Cite	PDF	Show Abstract
0%	0.0	2024	**[213] Multi-Lingual Development & Programming Languages Interoperability: An Empirical Study** Tsvi Cherny-Shahar and A. Yehudai ArXiv · Nov 13, 2024 · 0 Citations · Cite	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
0%	0.0	Unknown	**[214] Number-parameterized types** HTML Oleg Kiselyov oleg Journal Not Provided · Date Not Available · 3 Citations · Cite	No abstract or full text available.
0%	15.6	2002	**[215] A system and language for building system-specific, static analyses** PDF Seth Hallem, ..., and D. Engler May 17, 2002 · 358 Citations · Cite	Show Abstract
0%	0.5	2017	**[216] Optimizing ROOT's Performance Using C++ Modules** PDF V. Vassilev Journal of Physics: Conference Series · Oct 1, 2017 · 4 Citations · Cite	Show Abstract
0%	0.2	2019	**[217] Towards seamless interfacing between dynamic languages and native code** PDF Guillaume Bertholon and Stephen Kell Proceedings of the 11th ACM SIGPLAN International Workshop on Virtual Machines and Intermediate Languages · Oct 22, 2019 · 1 Citations · Cite	Show Abstract
0%	0.7	2022	**[218] Visualization of Syntax and Semantics for Simple Functional Language of Natural Numbers and Boolean Values** HTML J. Perháč, ..., and S. Shatnyi 2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC) · Jul 6, 2022 · 2 Citations · Cite	Show Abstract
0%	0.5	2014	**[219] Interoperability in Programming Languages** PDF Todd Malone Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal · Aug 13, 2014 · 5 Citations · Cite	Show Abstract
0%	0.1	2011	**[220] Interpretational overhead in system software** HTML B. Feigin Date Not Available · 1 Citations · Cite	No abstract or full text available.
0%	14.8	2009	**[221] F2PY: a tool for connecting Fortran and Python programs** PDF Pearu Peterson Int. J. Comput. Sci. Eng. · Nov 1, 2009 · 230 Citations · Cite	Show Abstract
0%	0.7	1992	**[222] Architecture of the XL C++ browser** HTML S. Javey, ..., and Richard Helm Nov 9, 1992 · 22 Citations · Cite	Show Abstract
0%	0.1	1998	**[223] Type-Directed Continuation Allocation** PDF Zhong Shao and Valery Trifonov Mar 25, 1998 · 4 Citations · Cite	No abstract or full text available.
0%	0.2	2016	**[224] Protecting Functional Programs From Low-Level Attackers** HTML Adriaan Larmuseau Date Not Available · 2 Citations · Cite	No abstract or full text available.
0%	0.0	2015	**[225] Sound and Complete Bidirectional Typechecking for Higher-Rank Polymorphism and Indexed Types** HTML Jana Dunfield and N. Krishnaswami Date Not Available · 0 Citations · Cite	No abstract or full text available.
0%	0.3	2009	**[226] Weak Updates and Separation Logic** HTML Gang Tan, ..., and Hongxu Cai New Generation Computing · Dec 3, 2009 · 5 Citations · Cite	No abstract or full text available.
0%	0.5	2021	**[227] Assessment of machine learning methods for state-to-state approaches** PDF L. Campoli, ..., and Polina Maltseva ArXiv · Apr 2, 2021 · 2 Citations · Cite	Show Abstract
0%	2.3	2021	**[228] Towards Modern C++ Language Support for MPI** HTML Sayan Ghosh, ..., and A. Lumsdaine 2021 Workshop on Exascale MPI (ExaMPI) · Nov 1, 2021 · 8 Citations · Cite	Show Abstract
0%	0.0	2005	**[229] C++/CLI: evolving C++ within the .NET platform** HTML S. Lippman Mar 18, 2005 · 0 Citations · Cite	Show Abstract
0%	0.1	2012	**[230] SEAN: Support Tool for Detecting Rule Violations in JNI Coding** PDF Haruna Nishiwaki, ..., and T. Yuasa Inf. Media Technol. · Aug 20, 2012 · 1 Citations · Cite	Show Abstract
0%	3.8	2015	**[231] Verified Compilers for a Multi-Language World** HTML Amal J. Ahmed Date Not Available · 39 Citations · Cite	Show Abstract
0%	0.0	2024	**[232] Portable Implementations of Work Stealing** PDF M. Yasugi, ..., and C. Takeuchi	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
			Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region · Jan 18, 2024 · 0 Citations · Cite	
0%	0.3	2013	[233] Java interoperability in managed X10 HTML Mikio Takeuchi, ..., and V. Saraswat Jun 20, 2013 · 3 Citations · Cite	Show Abstract
0%	0.5	2003	[234] Dynamic compilation of C++ template code PDF M. Cole and S. Parker Sci. Program. · Dec 1, 2003 · 10 Citations · Cite	Show Abstract
0%	0.8	2024	[235] Clog: A Declarative Language for C Static Code Checkers PDF Alexandru Dura and Christoph Reichenbach Proceedings of the 33rd ACM SIGPLAN International Conference on Compiler Construction · Feb 17, 2024 · 1 Citations · Cite	Show Abstract
0%	0.5	2014	[236] Foreign inline code HTML M. Chakravarty ACM SIGPLAN Notices · Sep 3, 2014 · 5 Citations · Cite	Show Abstract
0%	0.3	2010	[237] Static macro data flow: Compiling global control into local control PDF Pritish Jetley and L. Kalé 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW) · Apr 19, 2010 · 4 Citations · Cite	Show Abstract
0%	0.5	2017	[238] Generic Library Interception for Improved Performance Measurement and Insight PDF Ronny Brendel, ..., and Sebastian Oeste ArXiv · Nov 12, 2017 · 4 Citations · Cite	Show Abstract
0%	7.4	2004	[239] Semantics of types for mutable state HTML Amal J. Ahmed Date Not Available · 158 Citations · Cite	No abstract or full text available.
0%	0.2	2019	[240] Learning to Adapt: Analyses for Configurable Software HTML J. Toman Journal Not Provided · Date Not Available · 1 Citations · Cite	No abstract or full text available.
0%	3.7	2024	[241] Studying the Impact of TensorFlow and PyTorch Bindings on Machine Learning Software Quality PDF Hao Li, ..., and C. Bezemer ArXiv · Jul 7, 2024 · 3 Citations · Cite	Show Abstract
0%	0.3	1998	[242] Optimizing away C++ exception handling PDF Jonathan L. Schilling ACM SIGPLAN Notices · Aug 1, 1998 · 8 Citations · Cite	Show Abstract
0%	0.5	1998	[243] High-level information-an approach for integrating front-end and back-end compilers HTML Sangyeun Cho, ..., and P. Yew Proceedings. 1998 International Conference on Parallel Processing (Cat. No.98EX205) · Aug 10, 1998 · 13 Citations · Cite	Show Abstract
0%	1.2	2007	[244] Improving software quality with static analysis HTML J. Foster, ..., and W. Pugh Jun 13, 2007 · 21 Citations · Cite	Show Abstract
0%	0.9	2014	[245] High-performance language interoperability in multi-language runtimes HTML Matthias Grimmer Oct 20, 2014 · 9 Citations · Cite	Show Abstract
0%	0.1	2008	[246] Foreign-Function Interfaces for Garbage-Collected Programming Languages HTML Marcus Crestani Journal Not Provided · Date Not Available · 2 Citations · Cite	No abstract or full text available.
0%	0.8	2024	[247] A C Subset for Ergonomic Source-to-Source Analyses and Transformations PDF João N. Matos, ..., and L. Sousa Proceedings of the 16th Workshop on Rapid Simulation and Performance Evaluation for Design · Jan 18, 2024 · 1 Citations · Cite	Show Abstract
0%	1.4	2023	[248] The Awkward World of Python and C++ PDF Manasvi Goyal, ..., and J. Pivarski ArXiv · Mar 3, 2023 · 3 Citations · Cite	Show Abstract
0%	0.2	2012	[249] On AOP techniques for C++-based HW/SW component implementation HTML T. Muck and A. Frôhlich 2012 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012) · Dec 1, 2012 · 3 Citations · Cite	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	1.3	2020	[250] MetaCG: annotated call-graphs to facilitate whole-program analysis Jan-Patrick Lehr, ..., and C. Bischof Proceedings of the 11th ACM SIGPLAN International Workshop on Tools for Automatic Program Analysis · Nov 15, 2020 · 6 Citations · Cite	HTML	Show Abstract
0%	0.9	2007	[251] A type-preserving closure conversion in haskell Louis-Julien Guillemette and Stefan Monnier Sep 30, 2007 · 16 Citations · Cite	HTML	Show Abstract
0%	8.1	2024	[252] RetroLLM: Empowering Large Language Models to Retrieve Fine-grained Evidence within Generation Xiaoxi Li, ..., and Zhicheng Dou ArXiv · Dec 16, 2024 · 3 Citations · Cite	PDF	Show Abstract
0%	0.3	2016	[253] Understanding and Fixing Multiple Language Interoperability Issues: The C/Fortran Case Nawrin Sultana, ..., and M. Hafiz 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE) · May 14, 2016 · 3 Citations · Cite	HTML	Show Abstract
0%	1.2	2006	[254] Strongly typed memory areas programming systems-level data structures in a functional language Iavor S. Diatchki and Mark P. Jones Sep 17, 2006 · 22 Citations · Cite	HTML	Show Abstract
0%	0.2	2015	[255] Formalizing a Secure Foreign Function Interface – Extended Version Adriaan Larmuseau and D. Clarke Journal Not Provided · Date Not Available · 2 Citations · Cite	HTML	No abstract or full text available.
0%	0.3	2015	[256] The Design of Terra: Harnessing the Best Features of High-Level and Low-Level Languages Zach DeVito and P. Hanrahan Date Not Available · 3 Citations · Cite	HTML	Show Abstract
0%	0.7	2022	[257] Cloudprofiler: TSC-based inter-node profiling and high-throughput data ingestion for cloud streaming workloads Shinhyung Yang, ..., and Bernd Burgstaller ArXiv · May 19, 2022 · 2 Citations · Cite	PDF	Show Abstract
0%	0.1	2008	[258] A Compile-Time Infrastructure for GCC Using Haskell Peter Collingbourne and P. Kelly Journal Not Provided · Date Not Available · 2 Citations · Cite	HTML	No abstract or full text available.
0%	1.2	2022	[259] Boxroot, fast movable GC roots for a better FFI Guillaume Munch-Maccagnoni INRIA Guillaume Munch-Maccagnoni Journal Not Provided · Date Not Available · 4 Citations · Cite	HTML	No abstract or full text available.
0%	0.4	2022	[260] On the Security of Python Virtual Machines: An Empirical Study Xinrong Lin, ..., and Qiliang Fan 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME) · Oct 1, 2022 · 1 Citations · Cite	HTML	Show Abstract
0%	0.6	2012	[261] Smalltalk in a C world D. Chisnall Aug 28, 2012 · 7 Citations · Cite	PDF	Show Abstract
0%	3.4	2015	[262] Dynamically composing languages in a modular way: supporting C extensions for dynamic languages Matthias Grimmer, ..., and H. Mössenböck Proceedings of the 14th International Conference on Modularity · Mar 16, 2015 · 34 Citations · Cite	PDF	Show Abstract
0%	0.7	2019	[263] Foreign language interfaces by code migration S. Chiba Proceedings of the 18th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences · Oct 21, 2019 · 4 Citations · Cite	HTML	Show Abstract
0%	3.2	2017	[264] Reliable and automatic composition of language extensions to C: the ableC extensible language framework Ted Kaminski, ..., and E. V. Wyk Proceedings of the ACM on Programming Languages · Oct 12, 2017 · 24 Citations · Cite	PDF	Show Abstract
0%	0.0	2023	[265] LLVM CFI Support for Rust Ramon Valle H2HC Magazine · Dec 9, 2023 · 0 Citations · Cite	HTML	Show Abstract
0%	16.1	2001	[266] An indexed model of recursive types for foundational proof-carrying code A. Appel and David A. McAllester ACM Trans. Program. Lang. Syst. · Sep 1, 2001 · 380 Citations · Cite	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	0.0	2022	[267] Building fast and reliable reverse engineering tools with Frida and Rust István-Attila Császár and R. R. Slavescu 2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP) · Sep 22, 2022 · 0 Citations · Cite	HTML	Show Abstract
0%	1.3	2021	[268] MPIs Language Bindings are Holding MPI Back M. Ruefenacht, ..., and P. Bangalore ArXiv · Jul 22, 2021 · 5 Citations · Cite	PDF	Show Abstract
0%	0.1	2012	[269] Composing typemaps in Twig G. Hulette, ..., and A. Malony Sep 26, 2012 · 1 Citations · Cite	HTML	Show Abstract
0%	1.9	2011	[270] JET: exception checking in the Java native interface Siliang Li and Gang Tan Oct 22, 2011 · 26 Citations · Cite	HTML	Show Abstract
0%	1.2	2010	[271] Domain-specific language integration with compile-time parser generator library Z. Porkoláb and Ábel Sinkovics Oct 10, 2010 · 18 Citations · Cite	HTML	Show Abstract
0%	0.0	2022	[272] GNOLL: Efficient Software for Real-World Dice Notation and Extensions Ian Frederick Vigogne Goodbody Hunter ArXiv · May 26, 2022 · 0 Citations · Cite	PDF	Show Abstract
0%	2.9	2007	[273] Ilea: inter-language analysis across java and c Gang Tan and Greg Morrisett Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications · Oct 21, 2007 · 51 Citations · Cite	HTML	Show Abstract
0%	2.8	2000	[274] Next-generation generic programming and its application to sparse matrix computations N. Mateev, ..., and V. Kotlyar May 8, 2000 · 70 Citations · Cite	PDF	Show Abstract
0%	10.6	2015	[275] Secure Compilation to Protected Module Architectures Marco Patrignani, ..., and Frank Piessens ACM Transactions on Programming Languages and Systems (TOPLAS) · Apr 16, 2015 · 106 Citations · Cite	PDF	Show Abstract
0%	0.2	2015	[276] Modelling an Assembly Attacker by Reflection Adriaan Larmuseau and D. Clarke Journal Not Provided · Date Not Available · 2 Citations · Cite	HTML	No abstract or full text available.
0%	0.0	2002	[277] Low-Level Run-Time Systems for Generic Code Generators F. Reig Journal Not Provided · Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
0%	5.3	2018	[278] Cross-Language Interoperability in a Multi-Language Runtime Matthias Grimmer, ..., and M. Luján ACM Transactions on Programming Languages and Systems (TOPLAS) · May 28, 2018 · 37 Citations · Cite	PDF	Show Abstract
0%	4.3	1998	[279] From ML to Ada: Strongly-typed language interoperability via source translation A. Tolmach and D. Oliva Journal of Functional Programming · Jul 1, 1998 · 116 Citations · Cite	PDF	Show Abstract
0%	0.3	2015	[280] C++ EDSL for parallel code generation D. Berényi 2015 Conference Grid, Cloud & High Performance Computing in Science (ROLCG) · Dec 28, 2015 · 3 Citations · Cite	HTML	Show Abstract
0%	2.1	2012	[281] Rejuvenating C++ programs through demacrofication Aditya Kumar, ..., and B. Stroustrup 2012 28th IEEE International Conference on Software Maintenance (ICSM) · Sep 23, 2012 · 27 Citations · Cite	HTML	Show Abstract
0%	0.2	2007	[282] Hybridthreads Compiler: Generation of Application Specific Hardware Thread Cores from C Jim Stevens 2007 International Conference on Field Programmable Logic and Applications · Nov 12, 2007 · 3 Citations · Cite	HTML	Show Abstract
0%	0.1	2014	[283] MetaJC++: A flexible and automatic program transformation technique using meta framework Nadera S. Beevi, ..., and S. S. Vinodchandra Central European Journal of Engineering · Apr 3, 2014 · 1 Citations · Cite	PDF	Show Abstract
0%	0.1	2014	[284] Improving Quality of Soft ware with Foreign Function Interfaces using Static Analysis Lehigh Preserve and Siliang Li Date Not Available · 1 Citations · Cite	HTML	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
0%	0.2	2004	**[285] mGTK: An SML Binding of Gtk+** K. Larsen and H. Niss Jun 27, 2004 · 4 Citations · Cite	HTML	No abstract or full text available.
0%	3.7	2006	**[286] Safe Java Native Interface** Gang Tan, ..., and Daniel C. Wang Date Not Available · 71 Citations · Cite	HTML	No abstract or full text available.
0%	8.7	2008	**[287] Design and implementation of transactional constructs for C/C++** Yang Ni, ..., and Xinmin Tian Proceedings of the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications · Oct 19, 2008 · 143 Citations · Cite	PDF	Show Abstract
0%	0.4	2020	**[288] Painting Flowers: Reasons for Using Single-State State Machines in Model-Driven Engineering** Nan Yang, ..., and Alexander Serebrenik 2020 IEEE/ACM 17th International Conference on Mining Software Repositories (MSR) · Mar 2, 2020 · 2 Citations · Cite	HTML	Show Abstract
0%	0.3	2016	**[289] Lowcode: Extending Pharo with C Types to Improve Performance** R. Salgado and Stéphane Ducasse Proceedings of the 11th edition of the International Workshop on Smalltalk Technologies · Aug 23, 2016 · 3 Citations · Cite	PDF	Show Abstract
0%	0.7	2007	**[290] Position : Lightweight static resources ? Sexy types for embedded and systems programming** O. Kiselyov and Chung-chieh Shan Journal Not Provided · Date Not Available · 13 Citations · Cite	HTML	No abstract or full text available.
0%	0.1	2016	**[291] Short Paper: Superhacks: Exploring and Preventing Vulnerabilities in Browser Binding Code** Fraser Brown Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security · Oct 24, 2016 · 1 Citations · Cite	PDF	Show Abstract
0%	0.0	2023	**[292] Cross-Language Call Graph Construction Supporting Different Host Languages** Mingzhe Hu, ..., and Yan Xiong 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER) · Mar 1, 2023 · 0 Citations · Cite	HTML	Show Abstract
0%	15.3	1999	**[293] Java Native Interface: Programmer's Guide and Specification** Sheng Liang Jun 20, 1999 · 396 Citations · Cite	HTML	No abstract or full text available.
0%	1.3	2020	**[294] Improving mobile app development using transpilers with maintainable outputs** Vinícius Jorge Vendramini, ..., and G. Mounié Proceedings of the XXXIV Brazilian Symposium on Software Engineering · Oct 21, 2020 · 6 Citations · Cite	PDF	Show Abstract
0%	0.5	2010	**[295] JNI light: an operational model for the core JNI** Gang Tan Mathematical Structures in Computer Science · Nov 28, 2010 · 7 Citations · Cite	HTML	Show Abstract
0%	0.0	2021	**[296] Svar: A Tiny C++ Header Brings Unified Interface for Multiple programming Languages** Yong Zhao, ..., and Hongkai Jiang ArXiv · Aug 19, 2021 · 0 Citations · Cite	PDF	Show Abstract
0%	0.0	1993	**[297] Language Interoperability Issues in the Integration of Heterogeneous Systems ; CU-CS-675-93** S. Sutton and P. Tarr Sep 1, 1993 · 1 Citations · Cite	HTML	Show Abstract
0%	0.1	2010	**[298] Semantic APIs for programming languages** Eugene Zouev 2010 6th Central and Eastern European Software Engineering Conference (CEE-SECR) · Oct 1, 2010 · 1 Citations · Cite	HTML	Show Abstract
0%	0.0	2024	**[299] Cross-Language Taint Analysis: Generating Caller-Sensitive Native Code Specification for Java** Shuangxiang Kan, ..., and Yulei Sui IEEE Transactions on Software Engineering · Jun 1, 2024 · 0 Citations · Cite	HTML	Show Abstract
0%	2.4	2023	**[300] MacoCaml: Staging Composable and Compilable Macros** Ningning Xie, ..., and J. Yallop Proceedings of the ACM on Programming Languages · Aug 30, 2023 · 4 Citations · Cite	PDF	Show Abstract

MATCH	CIT./YEAR	DATE	REFERENCE	RELEVANCE
0%	0.9	2020	[301] Really Embedding Domain-Specific Languages into C++ PDF H. Finkel, ..., and Johannes Doerfert 2020 IEEE/ACM 6th Workshop on the LLVM Compiler Infrastructure in HPC (LLVM-HPC) and Workshop on Hierarchical Parallelism for Exascale Computing (HiPar) · Oct 16, 2020 · 4 Citations · Cite	Show Abstract
0%	0.6	2022	[302] Bridging the language gap: an empirical study of bindings for open source machine learning libraries across software package ecosystems PDF Hao Li and C. Bezemer Empir. Softw. Eng. · Jan 18, 2022 · 2 Citations · Cite	Show Abstract
0%	20.4	1974	[303] Towards a theory of type structure HTML J. C. Reynolds Apr 9, 1974 · 1042 Citations · Cite	No abstract or full text available.
0%	1.3	2014	[304] LambdaJIT: a dynamic compiler for heterogeneous optimizations of STL algorithms HTML Thibaut Lutz and Vinod Grover Sep 3, 2014 · 14 Citations · Cite	Show Abstract
0%	0.0	2003	[305] Typing XHTML Web Applications in SMLserver HTML M. Elsman and K. Larsen Date Not Available · 1 Citations · Cite	No abstract or full text available.
0%	2.7	2020	[306] Copy-and-patch compilation: a fast compilation algorithm for high-level languages and bytecode PDF Haoran Xu and Fredrik Kjolstad Proceedings of the ACM on Programming Languages · Nov 26, 2020 · 12 Citations · Cite	Show Abstract
0%	7.2	2017	[307] Finding and Preventing Bugs in JavaScript Bindings HTML Fraser Brown, ..., and D. Stefan 2017 IEEE Symposium on Security and Privacy (SP) · May 22, 2017 · 57 Citations · Cite	Show Abstract
0%	3.9	1992	[308] No assembly required: compiling standard ML to C PDF D. Tarditi, ..., and A. Acharya LOPLAS · Jun 1, 1992 · 127 Citations · Cite	Show Abstract
0%	0.6	2016	[309] Transforming C++11 Code to C++03 to Support Legacy Compilation Environments PDF Gábor Antal, ..., and József Mihalicza 2016 IEEE 16th International Working Conference on Source Code Analysis and Manipulation (SCAM) · Oct 1, 2016 · 5 Citations · Cite	Show Abstract
0%	0.4	2008	[310] Serialization for Ubiquitous Systems: An Evaluation of High Performance Techniques on Java Micro Edition HTML N. Palmer, ..., and Henri E. Bal 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies · Sep 29, 2008 · 6 Citations · Cite	Show Abstract
0%	0.0	2002	[311] Retargetable cross compilation techniques: comparison and analysis of GCC and Zephyr HTML Guilan Dai, ..., and Jun Dai ACM SIGPLAN Notices · Jun 2, 2002 · 1 Citations · Cite	Show Abstract
0%	7	2014	[312] Verifying an Open Compiler Using Multi-language Semantics PDF James T. Perconti and Amal J. Ahmed Apr 5, 2014 · 77 Citations · Cite	No abstract or full text available.
0%	2.6	2006	[313] Polymorphic Type Inference for the JNI PDF Michael Furr and J. Foster Mar 27, 2006 · 50 Citations · Cite	No abstract or full text available.
0%	18.5	2007	[314] A history of Haskell: being lazy with class HTML P. Hudak, ..., and P. Wadler Proceedings of the third ACM SIGPLAN conference on History of programming languages · Jun 9, 2007 · 331 Citations · Cite	Show Abstract
0%	8.9	2013	[315] Fully abstract compilation to JavaScript PDF C. Fournet, ..., and B. Livshits Jan 23, 2013 · 109 Citations · Cite	Show Abstract
0%	1.4	2018	[316] Pallene: a statically typed companion language for lua HTML Hugo Musso Gualandi and R. Ierusalimschy Sep 20, 2018 · 9 Citations · Cite	Show Abstract
0%	2.9	2010	[317] Semantic foundations for typed assembly languages PDF Amal J. Ahmed, ..., and Daniel C. Wang ACM Trans. Program. Lang. Syst. · Mar 1, 2010 · 44 Citations · Cite	Show Abstract
0%	0.0	2019	[318] THE VERIFIERS ? “ Can you trust your compiler ? ” So began HTML	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
			Daniel Patterson Journal Not Provided · Date Not Available · 0 Citations · Cite		
0%	0.3	1997	[319] Making a dataparallel language portable for massively parallel array computers M. Herbordt, ..., and C. Weems Proceedings Fourth IEEE International Workshop on Computer Architecture for Machine Perception. CAMP'97 · Oct 20, 1997 · 8 Citations · Cite	HTML	Show Abstract
0%	0.3	2005	[320] Towards a High-Productivity and High-Performance Marshaling Library for Compound Data Albert Cohen and C. Herrmann Date Not Available · 7 Citations · Cite	HTML	No abstract or full text available.
Not measured	9.2	2007	[321] A very modal model of a modern, major, general type system A. Appel, ..., and Jérôme Vouillon Jan 17, 2007 · 169 Citations · Cite	HTML	Show Abstract
Not measured	1.1	2018	[322] Reasoning About Foreign Function Interfaces Without Modelling the Foreign Language Alexi Turcotte, ..., and G. Richards Oct 28, 2018 · 7 Citations · Cite	PDF	Show Abstract
Not measured	0.7	2015	[323] Fine-grained Language Composition Edd Barrett, ..., and L. Tratt ArXiv · Mar 30, 2015 · 7 Citations · Cite	HTML	No abstract or full text available.
Not measured	1.6	2008	[324] Safe Cross-Language Inheritance Kathryn E. Gray Jul 7, 2008 · 27 Citations · Cite	HTML	No abstract or full text available.
Not measured	7.5	2012	[325] Chaperones and impersonators: run-time support for reasonable interposition T. Strickland, ..., and M. Flatt Oct 19, 2012 · 94 Citations · Cite	HTML	Show Abstract
Not measured	4.1	2006	[326] Leveraging .NET meta-programming components from F#: integrated queries and interoperable heterogeneous execution Don Syme Sep 16, 2006 · 76 Citations · Cite	HTML	Show Abstract
Not measured	1.3	1997	[327] Green card: a foreign-language interface for Haskell S. Jones, ..., and A. Reid Feb 14, 1997 · 38 Citations · Cite	HTML	No abstract or full text available.
Not measured	4	1999	[328] The design of a class mechanism for Moby Kathleen Fisher and John H. Reppy May 1, 1999 · 104 Citations · Cite	PDF	Show Abstract
Not measured	0.0	2015	[329] Generating safe boundary APIs between typed EDSLs and their environments Bob Reynders, ..., and Frank Piessens Proceedings of the 2015 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences · Oct 26, 2015 · 0 Citations · Cite	PDF	Show Abstract
Not measured	0.0	2014	[330] 1-1-2012 Dependent Interoperability Peter-Michael Osera Journal Not Provided · Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
Not measured	0.7	2005	[331] Recuperación de empresas por sus trabajadores y autogestión obrera: un estudio de caso de una empresa en Argentina L. M. Deledicque, ..., and J. Moser Date Not Available · 15 Citations · Cite	HTML	No abstract or full text available.
Not measured	1	2014	[332] Exception analysis in the Java Native Interface Siliang Li and Gang Tan Sci. Comput. Program. · Sep 1, 2014 · 11 Citations · Cite	PDF	No abstract or full text available.
Not measured	0.0	2011	[333] Composing heterogeneous software with style Stephen Kell Jul 25, 2011 · 0 Citations · Cite	HTML	Show Abstract
Not measured	0.0	2008	[334] Pidgin : Types for safe and natural multi-language interoperation Qualifying Exam Jonathan M. Hsieh Journal Not Provided · Date Not Available · 0 Citations · Cite	HTML	No abstract or full text available.
Not measured	10.6	2007	[335] Types, bytes, and separation logic Harvey Tuch, ..., and Michael Norrish Jan 17, 2007 · 194 Citations · Cite	HTML	Show Abstract
Not measured	0.7	2004	[336] Foreign Interface for PLT Scheme Eli Barzilay and Dmitry S. Orlovsky Date Not Available · 14 Citations · Cite	HTML	No abstract or full text available.

MATCH	CIT./YEAR	DATE	REFERENCE		RELEVANCE
Not measured	0.0	2020	**[337] A New Backend for Standard ML of New Jersey** Kavon Farvardin and John H. Reppy Proceedings of the 32nd Symposium on Implementation and Application of Functional Languages · Sep 2, 2020 · 0 Citations · Cite	PDF	Show Abstract
Not measured	2	2020	**[338] The history of Standard ML** David B. MacQueen, ..., and John H. Reppy Proceedings of the ACM on Programming Languages · Jun 12, 2020 · 10 Citations · Cite	PDF	Show Abstract