AFRL-AFOSR-VA-TR-2019-0087

The UniMath library for type theory and programming languages

Robert MacPherson
Institute For Advanced Study

01/04/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
14-04-2019	Final Performance	30 Sep 2017 to 29 Sep 2018

4. TITLE AND SUBTITLE
The UniMath library for type theory and programming languages

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-17-1-0363

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Robert MacPherson, Benedikt Ahrens, Lennart Beringer

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute For Advanced Study
1 Einstein Dr
Princeton, NJ 08540-4952 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2019-0087

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Matthew Weaver, Ph.D. student at Princeton University, worked under the direction of Lennart Beringer and also (initially) under Voevodsky, and also in collaboration with Dan Licata (Wesleyan University) and Dimitris Tsementzis (Rutgers). Weaver began to develop bicubical directed type theory in collaboration with Daniel R. Licata. Bicubical directed type theory is a constructive model of type theory in where types are augmented with higher-dimensional structure in two ways: beyond being just a set of terms, each type also has terms corresponding to paths between terms in the type, and terms representing directed morphisms from one term to another. This additional higher-dimensional data is organized 'cubically' to improve the computational properties of the theory. Given types contain two notions of pathsthe undirected paths and directed morphismsevery term is represented as two cubes: the first containing the path structure it represents and the other the morphism structure. The goal of bicubical directed type theory is for it to ultimately be a constructive model of a type theory with directed univalence.
Lennart Beringer, research scholar in the Computer Science Department at Princeton University, carried out research on dependently-typed verification of security protocols in Coq. This work used the Verified Software Toolchain (VST), an implementation of impredicative higher-order concurrent separation logic for the C language. VST's step-indexed semantic model Appel et al., 2014, foundationally justifies functional correctness assertions with respect to the operational semantics of Clight, the topmost intermediate language of the CompCert verified C compiler.
Benedikt Ahrens, Birmingham Fellow at University of Birmingham, UK, carried out research into the use of categorical structures for the semantics of homotopy type theory. Two papers were published, one was a

15. SUBJECT TERMS
Univalent Foundations, Dependent Type Theory, Programming Languages, Formalized Mathematics

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			NGUYEN, TRISTAN
Unclassified	Unclassified	Unclassified	UU		19b. TELEPHONE NUMBER *(Include area code)* 703-696-7796

University of Illinois
at Urbana-Champaign

Department of Mathematics

Daniel R. Grayson
Professor Emeritus
http://dangrayson.com/
December 10, 2018

Grant/Contract Title: The UniMath library for type theory and programming languages
Grant/Contract Number: FA9550-17-1-0363
Principal Investigator: Vladimir Voevodsky; replaced by Robert MacPherson
Date of report: November 15, 2018

Vladimir Voevodsky, the principal investigator for this grant, died shortly after the start of the grant period, on September 30, 2017. As a result, the Institute for Advanced Study School of Mathematics arranged for Robert MacPherson to take over as the principal investigator, and asked Daniel Grayson to help with the technical aspects of the grant administration.

The expenditures were distributed essentially in accord with Voevodsky's original plan, with the bulk of the funding going to Lennart Beringer, Princeton University, and his student Matthew Weaver, for whom Voevodsky had been serving as co-advisor.

The remaining portion of the funding went to support Benedikt Ahrens. The original plan was to support Ahrens partially in a post-doctoral position at The Ohio State University, but as it happened, Ahrens won an appointment at the University of Birmingham, so the revised plan was to support his research program there.

We include progress reports from both teams.

Sincerely,

Robert MacPherson

Daniel R. Grayson

Robert MacPherson
Daniel R. Grayson

AFOSR FA9550-17-1-0363: FINAL REPORT

MATTHEW WEAVER, LENNART BERINGER

1. Introduction

This report summarizes research and activities carried out under AFOSR grant FA9550-17-1-0363 *The UniMath Library for Type Theory and Programming Languages*. This grant was originally led by Vladimir Voevodsky (IAS Princeton), with Lennart Beringer (Princeton University) as subcontractor. Following Voevodsky's death during the first months of the grant duration, administration of the grant was transferred to Daniel R. Grayson.

Research concerning the primary topic of the grant, applications of homotopy type theory in computer science, is described in Section 2. This part was composed by the graduate student funded by this grant, Matthew Weaver (Princeton University) and describes work that he carried out first under the guidance of Voevodsky and later in collaboration with Dan Licata (Wesleyan University) and Dimitris Tsementzis (Rutgers). Section 3 describes complementary work on formal verification of cryptographic code in Coq, carried out during the duration of this grant by Beringer. In coordination with Weaver's PhD advisor, Andrew W. Appel (Princeton), Beringer also advised Weaver's research, encouraged him to pursue the mathematical collaborations with Licata and Tsementzis, and gave feedback on technical results from a (functional) programming language and computer science perspective.

2. Research in Homotopy type theory (Matthew Weaver)

2.1. **Bicubical Directed Type Theory.** During the grant period, I began to develop bicubical directed type theory in collaboration with Daniel R. Licata. Bicubical directed type theory is a constructive model of type theory in where types are augmented with higher-dimensional structure in two ways: beyond being just a set of terms, each type also has terms corresponding to paths between terms in the type, and terms representing directed morphisms from one term to another. This additional higher-dimensional data is organized "cubically" to improve the computational properties of the theory. Given types contain two notions of paths—the undirected paths and directed morphisms—every term is represented as two cubes: the first containing the path structure it represents and the other the morphism structure. The goal of bicubical directed type theory is for it to ultimately be a constructive model of a type theory with directed univalence.

In the undirected setting, univalence is a property of a universe of types that states that the type of paths in the universe is equivalent to the type of equivalences of types: In particular, a path between A and B is the same thing has having a pair of invertible functions between A and B. This allows us to freely move back and forth along paths between types (by applying the equivalence), and furthermore transfer proofs and properties along an equivalence (by coercing along the corresponding path).

While undirected univalence still is meaningful and useful in the directed setting, directed type theory allows us to express a second notion of univalence with respect to the morphisms.

Directed univalence is the property that morphisms in the universe are equivalent to functions in the universe. Assuming the universe with directed univalence is a type in a universe with undirected univalence, this results in a path between the type of morphisms from A to B—$\mathsf{Hom}_{\mathsf{U}}\, A\, B$—and the function space $A \to B$. Again, this is quite useful as whenever one has something defined or proven over a type A and a function from A to B they can use directed univalence to get the same definition or proof over B.

2.2. **What I have proven.** We define the type theory as the internal language of a topos, as is done in Orton and Pitts [2016], and build our type theory as a direct extension of (undirected) cartesian cubical type theory [Angiuli et al., 2017]. We add the directed structure to cartesian cubical type theory in a way akin to what was first done in bisimplicial directed type theory [Riehl and Shulman, 2018].

Using this setup, we have so far been able to define a model of type theory in bicubical sets with the following:

- directed and undirected path types;
- a universe that is univalent with respect to the undirected paths, called the universe of Kan fibrations;
- a universe of covariant fibrations that is a subuniverse of the universe of Kan fibrations (i.e. it is univalent for undirected paths). For this universe, we have constructed:
  - a function that sends a directed paths to functions,
  - a function that sends functions to directed paths,
  - such that this pair shows the function space $A \to B$ is a retract of the directed path space $\mathsf{Hom}_{\mathsf{U}}\, A\, B$ for any two types A and B in the universe of covariant fibrations.

Therefore, the only thing remaining to complete directed univalence is to show that these two functions additionally show that $\mathsf{Hom}_{\mathsf{U}}\, A\, B$ is a retract of $A \to B$, and thus we have an equivalence between the directed path space $\mathsf{Hom}_{\mathsf{U}}\, A\, B$ and the function space $A \to B$.

2.3. **What needs to be finished.** In order to finish our proof/construction of directed univalence, we need to work out in full detail the semantics of our theory as a model category, and add new syntax/structure constructively as is justified by the semantics. While we have some ideas of how this should be completed, this direction of our work is still ongoing.

2.4. **Finite Inverse Categories.** In addition to working on directed type theory, I also developed an inductive syntax for expressing finite inverse categories – a class of categories that appear commonly in the semantics of type theory as they represent dependency/fibration structure of dependent/fibrant types. A preprint is available on the arXiv [Tsementzis and Weaver, 2017].

2.5. **Activities.** I attended the following events using funding from this grant:

- Big Proof Workshop on Computer-Aided Mathematical Proof. Isaac Newton Institute for Mathematics, Cambridge, U.K.. 10th July 2017 - 14th July 2017
- Workshop on Homotopy Type Theory and Univalent Foundations at the International Conference on Formal Structures for Computation and Deduction. Oxford, U.K.. 3rd September 2017 - 9th September 2017

3. Verification of cryptographic implementations (Lennart Beringer)

In addition to work envisioned in the grant proposal, we also carried out research on dependently-typed verification of security protocols in Coq. This work used the Verified Software Toolchain (VST), an implementation of impredicative higher-order concurrent separation logic for the C language. VST's step-indexed semantic model Appel et al. [2014] foundationally justifies functional correctness assertions with respect to the operational semantics of Clight, the topmost intermediate language of the CompCert verified C compiler.

The object of study was a key programmatic interface in cryptographic libraries of the OpenSSL family, hash envelopes and contexts. This interface presents a client programmer with a uniform way to invoke and utilize different hash-functions (SHA256, MD5,...), and is implemented using a custom-designed object-style protocol that extensively relies on function pointers. Previous work had resulted in a VST-verification of OpenSSL's implementation of SHA256 Appel [2015], as well as verifications of additional cryptographic constructions that are built on top of SHA256: hash-based message authentication (HMAC-SHA256, Beringer et al. [2015]), HMAC-based deterministic random number generation (HMAC-DRBG Ye et al. [2017]), and HMAC-based key derivation (HKDF). A key advantage of using a Coq-embedded verification tool such as VST is the ability to connect the verification of *implementation correctness* – which in our case consists of applying VST's symbolic execution of Hoare triples, followed by semi-automatic discharging of side conditions Cao et al. [2018] – with the *model-level* verification of *cryptographic correctness*. The latter task is achieved by using the Foundational Cryptography Framework (FCF, Petcher and Morrisett [2015]), a Coq-embedded library of definitions, theorems, and proof automation tactics for reasoning about relational properties over the monad of probability spaces. This enables a smooth encoding of Bellare and Rogaway's approach Bellare and Rogaway [2006] of phrasing cryptographic proofs as games between a system and an attacker in which the latter's likelihood to break a cryptographic construction can be numerically bound.

The main challenge of the present work was to identify specification idioms that match the API abstractions inherent in OpenSSL's envelope API, equip the functions with sufficiently precise specifications, and then verify an implementation. Indeed, authors of OpenSSL-style libraries confirmed that the structuring principles, invariants, and even the order in which an client programmer of this code base would be expected to invoke the API functions are somewhat underspecified and leave room for misinterpretations in certain corner cases.

We identified a minor implementation glitch in Google's implementation of the interface, BoringSSL, that broke the abstraction boundary and permitted application programmers to invoke externally visible functions with unexpected call arguments; following our suggestion, Google engineers corrected in their codebase[1].

Our verification stress-tested the VST and its proof automation, highlighting shortcomings in the treatment of function pointers, object invariants, and of subsumption and context management rules for mutually recursive functions. Solutions to these challenges were jointly developed with other VST members, to ensure they are sufficiently general to be useful to other VST verification efforts beyond the envelope case study. At present I am exploring how Coq's strong encapsulation features can be further exploited to foundationally enforce established software-engineering principles: the current user-facing specifications constitute

[1]See https://boringssl.googlesource.com/boringssl/+/6b35262272e435aa5a6366b3d846fc9125984261.

whitebox abstractions, where symbolic object states in pre- and postconditions are abbreviations of low-level invariants that *need not* be unfolded during the verification of clients. I contend that Coq's module system, existential types, and (meta-logical) parametricity can be used to enforce the stronger encapsulation discipline of *blackbox abstraction*, where the object invariants are encoded as fully opaque entities that a client *cannot* unfold. If successful, this would lead to a specification discipline in which data abstraction is complemented by representation independence: substituting one implementation of the API with another one is not only opaque to a client programmer but remains so during a subsequent verification: code reuse is extended to proof reuse.

A conference publication describing the conceptual and practical results of this work is in preparation.

References

C. Angiuli, G. Brunerie, T. Coquand, K.-B. Hou (Favonia), R. Harper, and D. R. Licata. Cartesian cubical type theory. Available from `https://github.com/dlicata335/cart-cube/blob/master/cart-cube.pdf`, 2017.

A. W. Appel. Verification of a cryptographic primitive: SHA-256. *ACM Trans. Program. Lang. Syst.*, 37(2):7:1–7:31, 2015. doi: 10.1145/2701415. URL `http://doi.acm.org/10.1145/2701415`.

A. W. Appel, R. Dockins, A. Hobor, L. Beringer, J. Dodds, C. Stewart, S. Blazy, and X. Leroy. *Program Logics for Certified Compilers*. Cambridge University Press, 2014. ISBN 978-1-10-704801-0. URL `http://www.cambridge.org/de/academic/subjects/computer-science/programming-languages-and-applied-logic/program-logics-certified-compilers`.

M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006. ISBN 3-540-34546-9. doi: 10.1007/11761679_25. URL `https://doi.org/10.1007/11761679_25`.

L. Beringer, A. Petcher, K. Q. Ye, and A. W. Appel. Verified correctness and security of openssl HMAC. In J. Jung and T. Holz, editors, *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015.*, pages 207–221. USENIX Association, 2015. URL `https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/beringer`.

Q. Cao, L. Beringer, S. Gruetter, J. Dodds, and A. W. Appel. Vst-floyd: A separation logic tool to verify correctness of C programs. *J. Autom. Reasoning*, 61(1-4):367–422, 2018. doi: 10.1007/s10817-018-9457-5. URL `https://doi.org/10.1007/s10817-018-9457-5`.

I. Orton and A. M. Pitts. Axioms for modelling cubical type theory in a topos. In *Computer Science Logic*, 2016.

A. Petcher and G. Morrisett. The foundational cryptography framework. In R. Focardi and A. C. Myers, editors, *Principles of Security and Trust - 4th International Conference, POST 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings*, volume

9036 of *Lecture Notes in Computer Science*, pages 53–72. Springer, 2015. ISBN 978-3-662-46665-0. doi: 10.1007/978-3-662-46666-7_4. URL https://doi.org/10.1007/978-3-662-46666-7_4.

E. Riehl and M. Shulman. A type theory for syntheticâĹđ-categories. *Higher Structures*, 1 (1), 2018.

D. Tsementzis and M. Weaver. Finite inverse categories as signatures. arXiv:1707.07339, 2017.

K. Q. Ye, M. Green, N. Sanguansin, L. Beringer, A. Petcher, and A. W. Appel. Verified correctness and security of mbedtls HMAC-DRBG. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 2007–2020. ACM, 2017. ISBN 978-1-4503-4946-8. doi: 10.1145/3133956.3133974. URL http://doi.acm.org/10.1145/3133956.3133974.

Report by CO-PI Ahrens on AFOSR grant FA9550-17-1-0363

Publications and preprints resulting from work funded by the grant:

Categorical structures for type theory in univalent foundations
B. Ahrens, P. Lumsdaine, V. Voevodsky
In: Logical Methods in Computer Science 14:3, 2018
https://lmcs.episciences.org/4814

Displayed categories
B. Ahrens, P. Lumsdaine
submitted to LMCS

High-level signatures and initial semantics
B. Ahrens, A. Hirschowitz, A. Lafont, and M. Maggesi
Computer Science Logic (CSL) 2018, LIPIcs Vol. 119, pp. 4:1-4:22
http://dx.doi.org/10.4230/LIPIcs.CSL.2018.4

Univalent foundations and the equivalence principle
B. Ahrens, P. R. North
To be published in Synthese, Springer, final copy-editing in progress

A modular formalization of bicategories in type theory
B. Ahrens and M. Maggesi
Talk at TYPES 2018
Work on an article in progress

How the money was spent:

In January - February 2018, I travelled to IAS and OSU (host: Sanjeevi Krishnan). At IAS, I worked with Peter Lumsdaine on the journal versions of the two papers referenced above. At OSU, I worked with North on the joint paper referenced above, and on a joint project on "A Higher Structure Identity Principle (HSIP)" with M. Shulman and D. Tsementzis. I also gave 2 talks on Univalent Foundations at OSU. Cost ca. USD 6200 for myself, and ca USD 1500 for Lumsdaine.

An Apple iPad Pro was purchased, together with some accessories, to enable me to work on refereeing papers etc while travelling. Cost ca USD 1700.

In March 2018, Anders Moertberg's visit to IAS was funded by my grant. He conferred with Grayson, Favonia, and Brunerie, and he gave a talk on higher inductive types in the proof assistant "cubicaltt". Cost ca USD 850.

In April, I taught a course at the Midlands Graduate School about Voevodsky's Univalent Foundations. I was mostly funded from other sources. Cost for AFOSR ca USD 320.

Still in April, I visited M. Maggesi in Florence, for work on formalization of bicategories in UniMath. Mostly funded by EUTypes, cost for AFOSR ca USD 400.

In May - June 2018, North visited Birmingham for work on the aforementioned article and HSIP. Cost ca USD 2750.

In June 2018, I attended the TYPES workshop where I presented the aforementioned work with Maggesi. I then attended the TOPOS school in Italy, learning how to transfer mathematical results across topos "bridges". Cost ca USD 3350.

In July 2018, I co-organized the HoTT/UF workshop affiliated to the FSCD conference, part of the Federated Logic Conference in Oxford, UK. I used grant money to fund three invited speakers, total cost USD 2200.

My own participation in the HoTT/UF workshop, and in the CategoryTheory conference right afterwards, cost ca USD 3000.

The paper "High-level signatures and initial semantics" was presented at CSL in Birmingham. Registration fees about USD 500.

Martin Escardo participated in the Logic Colloquium, where he gave a talk on Univalent Foundations. His travel was funded by the AFOSR grant. Escardo also received funding for a workshop at Dagstuhl on "Formalization of Mathematics in Type Theory" that he organized, and where I participated. Total cost ca USD 2400.

In September 2018, I participated in the Vladimir Voevodsky Memorial Conference, IAS, where I gave an invited talk, and visited OSU for two weeks, working on HSIP. Total cost USD 3600.

Also in September 2018, we had a workshop in Birmingham, "First Symposium on Compositional Structures", on, in particular, type theory and category theory. We funded some participants from the grant, total cost ca USD 3000.

AFOSR Deliverables Submission Survey

Response ID:10807 Data

1.

Report Type

Final Report

Primary Contact Email

Contact email if there is a problem with the report.

danielrichardgrayson@gmail.com

Primary Contact Phone Number

Contact phone number if there is a problem with the report

2173770458

Organization / Institution name

Institute for Advanced Study

Grant/Contract Title

The full title of the funded effort.

The UniMath library for type theory and programming languages

Grant/Contract Number

AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-17-1-0363

Principal Investigator Name

The full name of the principal investigator on the grant or contract.

Robert MacPherson

Program Officer

The AFOSR Program Officer currently assigned to the award

Tristan Nguyen

Reporting Period Start Date

12/30/2017

Reporting Period End Date

12/29/2018

Abstract

Matthew Weaver, Ph.D. student at Princeton University, worked under the direction of Lennart Beringer and also (initially) under Voevodsky, and also in collaboration with Dan Licata (Wesleyan University) and Dimitris Tsementzis (Rutgers). Weaver began to develop bicubical directed type theory in collaboration with Daniel R. Licata. Bicubical directed type theory is a constructive model of type theory in where types are augmented with higher-dimensional structure in two ways: beyond being just a set of terms, each type also has terms corresponding to paths between terms in the type, and terms representing directed morphisms from one term to another. This additional higher-dimensional data is organized "cubically" to improve the computational properties of the theory. Given types contain two notions of paths—the undirected paths and directed morphisms—every term is represented as two cubes: the first containing the path structure it represents and the other the

morphism structure. The goal of bicubical directed type theory is for it to ultimately be a constructive model of a type theory with directed univalence.

Lennart Beringer, research scholar in the Computer Science Department at Princeton University, carried out research on dependently-typed verification of security protocols in Coq. This work used the Verified Software Toolchain (VST), an implementation of impredicative higher-order concurrent separation logic for the C language. VST's step-indexed semantic model Appel et al., 2014, foundationally justifies functional correctness assertions with respect to the operational semantics of Clight, the topmost intermediate language of the CompCert verified C compiler.

Benedikt Ahrens, Birmingham Fellow at University of Birmingham, UK, carried out research into the use of categorical structures for the semantics of homotopy type theory. Two papers were published, one was accepted, one was submitted, and another is in progress.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

SF_298_-_AFOSR_IAS_progress_report.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

combined-progress-report-IAS-Voevodsky-2018.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

Categorical structures for type theory in univalent foundations
B. Ahrens, P. Lumsdaine, V. Voevodsky
In: Logical Methods in Computer Science 14:3, 2018
https://lmcs.episciences.org/4814

High-level signatures and initial semantics
B. Ahrens, A. Hirschowitz, A. Lafont, and M. Maggesi
Computer Science Logic (CSL) 2018, LIPIcs Vol. 119, pp. 4:1-4:22
http://dx.doi.org/10.4230/LIPIcs.CSL.2018.4

New discoveries, inventions, or patent disclosures:

Do you have any discoveries, inventions, or patent disclosures to report for this period?

No

Please describe and include any notable dates

Do you plan to pursue a claim for personal or organizational intellectual property?

Changes in research objectives (if any):

none

Change in AFOSR Program Officer, if any:

none

Extensions granted or milestones slipped, if any:

none

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Dec 30, 2018 16:18:09 Success: Email Sent to: danielrichardgrayson@gmail.com