References

[1] H. Hojjat and P. Rümmer, "The ELDARICA horn solver," in *2018 Formal Methods in Computer Aided Design, FMCAD 2018, Austin, TX, USA, October 30 - November 2, 2018*, N. Bjørner and A. Gurfinkel, Eds. IEEE, 2018. doi: 10.23919/FMCAD.2018.8603013 pp. 1–7. [Online]. Available: https://doi.org/10.23919/FMCAD.2018.8603013

[2] A. Alshnakat, D. Gurov, C. Lidström, and P. Rümmer, "Constraint-based contract inference for deductive verification," *Deductive Software Verification: Future Perspectives*, to appear in: Hähnle, R., Bubel R. (eds), KeY 20 Years State-of-Art Volume on Deductive Verification. Lecture Notes in Computer Science, vol. 12345, 2020.

[3] D. R. Cok, "Java automated deductive verification in practice: Lessons from industrial proof-based projects," in *Leveraging Applications of Formal Methods, Verification and Validation. Industrial Practice*, T. Margaria and B. Steffen, Eds. Cham: Springer International Publishing, 2018. ISBN 978-3-030-03427-6 pp. 176–193.

[4] G. Klein, "From a verified kernel towards verified systems," in *Programming Languages and Systems*, K. Ueda, Ed. Berlin,: Springer Berlin Heidelberg, 2010. ISBN 978-3-642-17164-2 pp. 21–33.

[5] G. T. Leavens and C. Clifton, *Lessons from the JML Project*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 134–143. ISBN 978-3-540-69149-5. [Online]. Available: https://doi.org/10.1007/978-3-540-69149-5_15

[6] M. Huth and M. Ryan, *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge university press, 2004. ISBN 13 978-0-511-26401-

[7] E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, *Handbook of model checking*. Springer, 2018, vol. 10. ISBN 978-3-319-10575-8

[8] L. Lamport, "The temporal logic of actions," *ACM Trans. Program. Lang. Syst.*, vol. 16, no. 3, p. 872–923, May 1994. [Online]. Available: https://doi.org/10.1145/177492.177726

[9] A. Pnueli, "The temporal logic of programs," in *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. IEEE, 1977. doi: 10.1109/SFCS.1977.32 pp. 46–57.

[10] E. M. Clarke and E. A. Emerson, "Design and synthesis of synchronization skeletons using branching time temporal logic," in *Logics of Programs*, D. Kozen, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982. ISBN 978-3-540-39047-3 pp. 52–71.

[11] E. A. EMERSON, "Chapter 16 – Temporal and modal logic," in *Formal Models and Semantics*, ser. Handbook of Theoretical Computer Science, J. VAN LEEUWEN, Ed. Amsterdam: Elsevier, 1990, pp. 995–1072. ISBN 978-0-444-88074-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780444880741500214

[12] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement for symbolic model checking," *J. ACM*, vol. 50, no. 5, p. 752–794, Sep. 2003. doi: 10.1145/876638.876643. [Online]. Available: https://doi.org/10.1145/876638.876643

[13] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu, "Bounded model checking," *Adv. Comput.*, vol. 58, pp. 117–148, 2003. doi: 10.1016/S0065-2458(03)58003-2. [Online]. Available: https://doi.org/10.1016/S0065-2458(03)58003-2

[14] D. Kroening and M. Tautschnig, "CBMC – C bounded model checker," in *Tools and Algorithms for the Construction and Analysis of Systems*, E. Ábrahám and K. Havelund, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN 978-3-642-54862-8 pp. 389–391.

[15] R. Floyd., "Assigning meaning to programs," in *Symposium on Applied Mathematics*, vol. 19. American Math. Soc., 1967, pp. 19–32.

[16] H. Riis Nielson, *Semantics with Applications: An Appetizer*, 1st ed., ser. Undergraduate Topics in Computer Science, 2007. ISBN 1-84628-692-1

[17] C. A. R. Hoare, "An axiomatic basis for computer programming," *Commun. ACM*, vol. 12, no. 10, pp. 576–580, 1969. doi:

10.1145/363235.363259. [Online]. Available: https://doi.org/10.1145/363235.363259

[18] K. R. Apt and E. Olderog, "Fifty years of Hoare's logic," *Formal Aspects Comput.*, vol. 31, no. 6, pp. 751–807, 2019. doi: 10.1007/s00165-019-00501-3. [Online]. Available: https://doi.org/10.1007/s00165-019-00501-3

[19] C. A. Furia, B. Meyer, and S. Velder, "Loop invariants: Analysis, classification, and examples," *ACM Comput. Surv.*, vol. 46, no. 3, Jan. 2014. doi: 10.1145/2506375. [Online]. Available: https://doi.org/10.1145/2506375

[20] C. A. R. Hoare, "Procedures and parameters: An axiomatic approach," in *Symposium on Semantics of Algorithmic Languages*, E. Engeler, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1971. ISBN 978-3-540-36499-3 pp. 102–116.

[21] S. Owicki and D. Gries, "An axiomatic proof technique for parallel programs I," *Acta Inf.*, vol. 6, no. 4, p. 319–340, Dec. 1976. doi: 10.1007/BF00268134. [Online]. Available: https://doi.org/10.1007/BF00268134

[22] E. W. Dijkstra, "Guarded commands, nondeterminacy and formal derivation of programs," *Commun. ACM*, vol. 18, no. 8, p. 453–457, Aug. 1975. doi: 10.1145/360933.360975. [Online]. Available: https://doi.org/10.1145/360933.360975

[23] E. Dijkstra, *A Discipline of Programming*. Prentice-Hall, 1976. ISBN 0-13-215871

[24] P. D. Mosses, "Formal semantics of programming languages: An overview," *Electronic Notes in Theoretical Computer Science*, vol. 148, no. 1, pp. 41–73, 2006. doi: https://doi.org/10.1016/j.entcs.2005.12.012 Proceedings of the School of SegraVis Research Training Network on Foundations of Visual Modelling Techniques (FoVMT 2004). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1571066106000429

[25] E. Cohen, M. Dahlweid, M. Hillebrand, D. Leinenbach, M. Moskal, T. Santen, W. Schulte, and S. Tobies, "VCC: A practical system for verifying concurrent C," in *Theorem Proving in Higher Order Logics*,

S. Berghofer, T. Nipkow, C. Urban, and M. Wenzel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03359-9 pp. 23–42.

[26] P. Cuoq, F. Kirchner, N. Kosmatov, V. Prevosto, J. Signoles, and B. Yakobowski, "Frama-C," in *Software Engineering and Formal Methods*, G. Eleftherakis, M. Hinchey, and M. Holcombe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33826-7 pp. 233–247.

[27] G. T. Leavens, E. Poll, C. Clifton, Y. Cheon, C. Ruby, D. Cok, P. Müller, J. Kiniry, P. Chalin, D. M. Zimmerman *et al.*, "JML reference manual," 2008.

[28] B. Meyer, "Applying "design by contract"," *Computer*, vol. 25, no. 10, p. 40–51, Oct. 1992. doi: 10.1109/2.161279. [Online]. Available: https://doi.org/10.1109/2.161279

[29] ——, "Eiffel: A language and environment for software engineering," *Journal of Systems and Software*, vol. 8, no. 3, pp. 199–246, 1988. doi: https://doi.org/10.1016/0164-1212(88)90022-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0164121288900222

[30] R. B. Findler and M. Felleisen, "Contracts for higher-order functions," *SIGPLAN Not.*, vol. 37, no. 9, p. 48–59, Sep. 2002. doi: 10.1145/583852.581484. [Online]. Available: https://doi.org/10.1145/583852.581484

[31] M. Blume and D. McAllester, "Sound and complete models of contracts," *J. Funct. Program.*, vol. 16, no. 4–5, p. 375–414, Jul. 2006. doi: 10.1017/S0956796806005971. [Online]. Available: https://doi.org/10.1017/S0956796806005971

[32] D. Gurov and J. Westman, *A Hoare Logic Contract Theory: An Exercise in Denotational Semantics*. Cham: Springer International Publishing, 2018, pp. 119–127. ISBN 978-3-319-98047-8. [Online]. Available: https://doi.org/10.1007/978-3-319-98047-8_8

[33] C. Lidström and D. Gurov, "An abstract contract theory for programs with procedures," in *Fundamental Approaches to Software Engineering*, E. Guerra and M. Stoelinga, Eds. Cham: Springer International Publishing, 2021. ISBN 978-3-030-71500-7 pp. 152–171.

[34] K. R. M. Leino, "This is boogie 2," June 2008. [Online]. Available: https://www.microsoft.com/en-us/research/publication/this-is-boogie-2-2/

[35] P. Baudin, P. Cuoq, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, and V. Prevosto, *ACSL: ANSI/ISO C Specification Language*.

[36] C. Ellison and G. Rosu, "An executable formal semantics of C with applications," in *Proceedings of the 39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '12. New York, NY, USA: Association for Computing Machinery, 2012. doi: 10.1145/2103656.2103719. ISBN 9781450310833 p. 533–544. [Online]. Available: https://doi.org/10.1145/2103656.2103719

[37] G. Roșu and T. F. Șerbănuță, "An overview of the K semantic framework," *The Journal of Logic and Algebraic Programming*, vol. 79, no. 6, pp. 397–434, 2010. doi: https://doi.org/10.1016/j.jlap.2010.03.012 Membrane computing and programming. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1567832610000160

[38] N. S. Papaspyrou, "Denotational semantics of ANSI C," *Computer Standards Interfaces*, vol. 23, no. 3, pp. 169–185, 2001. doi: https://doi.org/10.1016/S0920-5489(01)00059-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548901000599

[39] S. Blazy and X. Leroy, "Mechanized Semantics for the Clight Subset of the C Language," *Journal of Automated Reasoning*, vol. 43, no. 3, pp. 263–288, Oct. 2009. doi: 10.1007/s10817-009-9148-3. [Online]. Available: https://doi.org/10.1007/s10817-009-9148-3

[40] N. Bjørner, A. Gurfinkel, K. McMillan, and A. Rybalchenko, *Horn Clause Solvers for Program Verification*. Cham: Springer International Publishing, 2015, pp. 24–51. ISBN 978-3-319-23534-9. [Online]. Available: https://doi.org/10.1007/978-3-319-23534-9_2

[41] A. Horn, "On sentences which are true of direct unions of algebras," *The Journal of Symbolic Logic*, vol. 16, no. 1, pp. 14–21, 1951. [Online]. Available: http://www.jstor.org/stable/2268661

[42] A. Colmerauer and P. Roussel, *The Birth of Prolog*. New York, NY, USA: Association for Computing Machinery, 1996, p. 331–367. ISBN 0201895021. [Online]. Available: https://doi.org/10.1145/234286.1057820

[43] A. Gupta, C. Popeea, and A. Rybalchenko, "Predicate abstraction and refinement for verifying multi-threaded programs," in *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '11. New York, NY, USA: Association for Computing Machinery, 2011. doi: 10.1145/1926385.1926424. ISBN 9781450304900 p. 331–344. [Online]. Available: https://doi.org/10.1145/1926385.1926424

[44] S. Grebenshchikov, N. P. Lopes, C. Popeea, and A. Rybalchenko, "Synthesizing software verifiers from proof rules," in *Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI '12. New York, NY, USA: Association for Computing Machinery, 2012. doi: 10.1145/2254064.2254112. ISBN 9781450312059 p. 405–416. [Online]. Available: https://doi.org/10.1145/2254064.2254112

[45] P. Rümmer, H. Hojjat, and V. Kuncak, "Disjunctive interpolants for Horn-clause verification," in *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, ser. Lecture Notes in Computer Science, N. Sharygina and H. Veith, Eds., vol. 8044. Springer, 2013. doi: 10.1007/978-3-642-39799-8_24 pp. 347–363. [Online]. Available: https://doi.org/10.1007/978-3-642-39799-8_24

[46] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 032114306X

[47] A. Cimatti, E. M. Clarke, F. Giunchiglia, and M. Roveri, "NUSMV: A new symbolic model checker," *Int. J. Softw. Tools Technol. Transf.*, vol. 2, no. 4, pp. 410–425, 2000. doi: 10.1007/s100090050046. [Online]. Available: https://doi.org/10.1007/s100090050046

[48] Z. Esen, "Extension of the Eldarica C model checker with heap memory," 2019.

[49] P. Baudin, F. Bobot, L. Correnson, Z. Dargaye, and A. Blanchard, "WP plug-in manual – frama-C 22.0 (Titanium)." [Online]. Available: https://frama-c.com/download/frama-c-wp-manual.pdf

[50] M. P. Robillard, E. Bodden, D. Kawrykow, M. Mezini, and T. Ratchford, "Automated API property inference techniques," *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 613–637, 2013. doi: 10.1109/TSE.2012.63

[51] M. D. Ernst, J. H. Perkins, P. J. Guo, S. McCamant, C. Pacheco, M. S. Tschantz, and C. Xiao, "The Daikon system for dynamic detection of likely invariants," *Sci. Comput. Program.*, vol. 69, no. 1–3, p. 35–45, Dec. 2007. doi: 10.1016/j.scico.2007.01.015. [Online]. Available: https://doi.org/10.1016/j.scico.2007.01.015

[52] N. Polikarpova, I. Ciupa, and B. Meyer, "A comparative study of programmer-written and automatically inferred contracts," in *Proceedings of the Eighteenth International Symposium on Software Testing and Analysis*, ser. ISSTA '09. New York, NY, USA: Association for Computing Machinery, 2009. doi: 10.1145/1572272.1572284. ISBN 9781605583389 p. 93–104. [Online]. Available: https://doi.org/10.1145/1572272.1572284

[53] Y. Wei, C. A. Furia, N. Kazmin, and B. Meyer, "Inferring better contracts," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11. New York, NY, USA: Association for Computing Machinery, 2011. doi: 10.1145/1985793.1985820. ISBN 9781450304450 p. 191–200. [Online]. Available: https://doi.org/10.1145/1985793.1985820

[54] J. Singleton, G. Leavens, H. Rajan, and D. Cok, "Inferring concise specifications of APIs," 05 2019.

[55] M. Alpuente, D. Pardo, and A. Villanueva, "Abstract contract synthesis and verification in the symbolic K framework," *Fundamenta Informaticae*, vol. 177, pp. 235–273, 12 2020. doi: 10.3233/FI-2020-1989

[56] M. K. Ramanathan, A. Grama, and S. Jagannathan, "Static specification inference using predicate mining," in *Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI '07. New York, NY, USA: Association

for Computing Machinery, 2007. doi: 10.1145/1250734.1250749. ISBN 9781595936332 p. 123–134. [Online]. Available: https://doi.org/10.1145/1250734.1250749

[57] P. Cousot, R. Cousot, and F. Logozzo, "Precondition inference from intermittent assertions and application to contracts on collections," in *Verification, Model Checking, and Abstract Interpretation*, R. Jhala and D. Schmidt, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18275-4 pp. 150–168.

[58] "International standard ISO/IEC 9899:2018 – programming languages – C." [Online]. Available: https://webstore.iec.ch/publication/63478

[59] G. Winskel, *The formal semantics of programming languages – an introduction*, ser. Foundation of computing series. MIT Press, 1993. ISBN 978-0-262-23169-5

[60] C. Flanagan and J. B. Saxe, "Avoiding exponential explosion: generating compact verification conditions," in *Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, London, UK, January 17-19, 2001*, C. Hankin and D. Schmidt, Eds. ACM, 2001. doi: 10.1145/360204.360220 pp. 193–205. [Online]. Available: https://doi.org/10.1145/360204.360220

[61] M. Ben-Ari, *Mathematical logic for computer science*, 3rd ed., 2012. ISBN 9781447141297

[62] J. Signoles, A. Thibaud, L. Correnson, M. Lemerre, and V. Prevosto, "Plug-in development guide – release 22.0 (Titanium)." [Online]. Available: https://frama-c.com/download/frama-c-plugin-development-guide.pdf

[63] D. Skantz, "Synthesis of annotations for partially automated deductive verification," 2021.

[64] T. A. Beyene, C. Popeea, and A. Rybalchenko, "Solving existentially quantified horn clauses," in *Computer Aided Verification*, N. Sharygina and H. Veith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN 978-3-642-39799-8 pp. 869–882.

Appendix A

Code and contracts for Scania battery module

Below follow the program code and the inferred contracts from the mock-up example of a Scania battery diagnostics module considered in Section 6.3. Please note that this example is heavily simplified to serve as an illustrating example in this project.

A.1 The program code

Below follows the program code together with the original ACSL function contract for the *batteryDiag* function.

```
1  //Global variables 'acting' as return variables
2  int return_modMin;
3  int return_modMax;
4  int return_modStatus;
5
6  int mod0_status;
7  int mod1_status;
8
9  int mod0_min;
10 int mod1_min;
11
12 int mod0_max;
13 int mod1_max;
14
15 int batt_min_output;
16 int batt_max_output;
17 int batt_status_output;
18
```