

Métodos Formais de Programação

2024/25

LMF / HASLab

MFP 24/25



MFP 24/25



MFP 24/25



MFP 24/25



Próximos 40 anos?

Incertezas?

Desafios?

O que vai mudar?

Próximos 40 anos?

Obsolescência do Programador

Próximos 40 anos?

Obsolescência do Programador

Obsolescência é a condição que ocorre a um **produto** ou **serviço** que deixa de ser útil, mesmo estando em perfeito estado de funcionamento, devido ao surgimento de um produto tecnologicamente mais avançado^[1]





ChatGPT

The official app by OpenAI



ChatGPT 4o mini

Log in

Will **formal methods** become obsolete with LLMs?



No, **formal methods** are unlikely to become obsolete with the advent of large language models (LLMs) like ChatGPT. While LLMs can assist with certain aspects of software development and analysis, **formal methods serve a distinct and critical role in ensuring software correctness, reliability, and safety, particularly in safety-critical systems.**

Why Formal Methods Are Still Essential:

- Mathematical Rigor and Verification:

 - Formal methods provide a **mathematical framework** for specifying and verifying the behavior of systems. This is especially important in fields like aerospace, automotive

Próximos 40 anos!



Desafios

Insegurança

- *Safety*



Desafios

Insegurança

- *Safety*
- *Security*



Insegurança

- *Testing?*

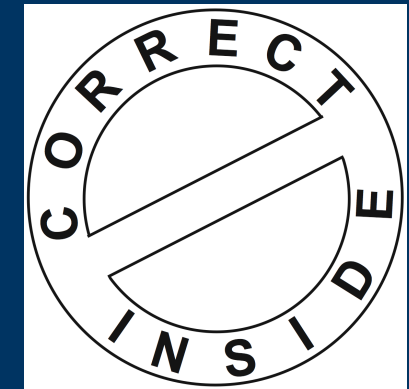
NOTES ON STRUCTURED PROGRAMMING
by
prof. dr. Edsger W. Dijkstra



Corollary of the first part of this section:
Program testing can be used to show the presence of bugs, but never to show their absence!

Segurança!

- *Safety*
- *Security*



BACK TO THE BUILDING BLOCKS:

A PATH TOWARD SECURE AND MEASURABLE SOFTWARE

FEBRUARY 2024



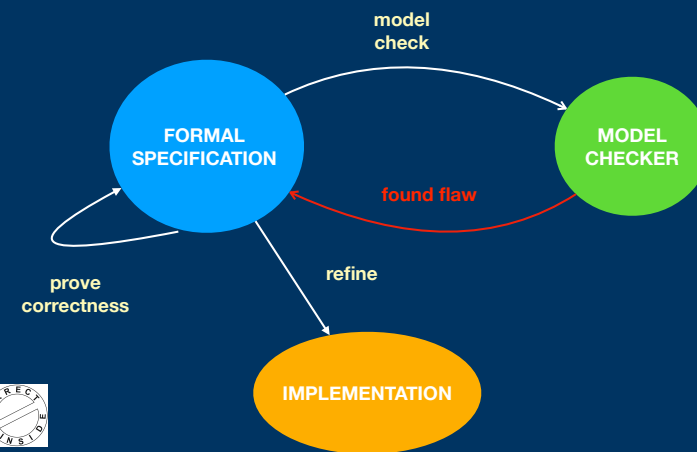
BACK TO THE BUILDING

First, in order to reduce memory safety vulnerabilities in software and hardware can better secure the building blocks. This report focuses on the **programming language** as a primary focus, **hardware architecture** and **formal methods** as complementary approaches to achieve similar outcomes. Second, in order to establish quality metrics, advances can be made to address the long-standing problem of software measurability. This report explores



BACK TO THE BUILDING

While there are several types of **formal methods** that span a range of techniques and stages in the software development process, this report highlights a few specific examples. **Sound static analysis** examines the software for specific properties without executing the code.^{xxi} This method is effective because it can be used across many representations of software, including the source code, architecture, requirements, and executables. **Model checkers** can answer questions about a number of higher-level properties.^{xxii} These algorithms can be used during production; however, they are limited in their scaled use due to their computational complexity. **Assertion-based testing** is a formal statement of properties carried in the code that may be used to cross-check the code during testing or production.^{xxiii} These generated proofs allow for faults to be detected much earlier and closer to the erroneous code, rather than tracing back from externally visible systems failures.



MFP 24/25

David Basin, ETH Zurich



“(...) We have **formalized** various models of EMV in Tamarin, a **symbolic model checker** (...) Tamarin was extremely effective in finding critical flaws, both known and new.”



MFP 24/25

David Basin, ETH Zurich



“(...) For example, we discovered multiple ways that an attacker can use a victim's EMV card (e.g., **Mastercard** or **Visa Card**) for high-valued purchases without the victim's supposedly required PIN.”



MFP 24/25

👋 Hey! My name is Wojciech Gawroński, but others call me *AWS Maniac*.



MFP 24/25

👋 Hey! My name is Wojciech Gawroński, but others call me *AWS Maniac*.



How formal methods helped AWS to design amazing services

Formal Methods of Distributed Systems Verification



Formal method B

- Paris Metro Line 14 contract won by Matra Transport (now Siemens Transportation Systems)
- 1995: B tools industrialised by Digilog (then Steria, now CLEARSY)
- ready by end 1998: 110 kLOC B model, 83% automatic proof, 86 kLOC Ada

Fully autonomous >1998; still in v 1.0; no software issue



MFP @ DIUM

MFP



Índice

Métodos Formais em Engenharia
de Software

Métodos Formais de Programação

(This page in English)

NEW Slides de apresentação do perfil.

Mestrado em Engenharia Informática

Bem vindo à página da edição de 2024/25 do perfil de *Métodos Formais de Programação* (MFP), que continua a unidade curricular obrigatória de *Métodos Formais em Engenharia de Software* (MFES).

Este perfil de especialização do MEI é lecionado por docentes do Grupo de



Índice

Métodos Formais em Engenharia
de Software

Cálculo de Sistemas de Informação

Programação Ciber-física

Verificação Formal

CSI

MFES

VF

PCF



Abordam-se técnicas clássicas de verificação de software apoiadas em três das ferramentas mais utilizadas: Coq, muito poderoso e com inúmeras aplicações; Why3, com grande variedade de ferramentas ao serviço da prova lógica; a suite TLA+, que integra um model-checker para verificação automática de propriedades de sistemas.

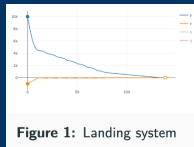


VF



Modelling and analysis
of computational
devices.

Focus on those that
interact with physical
processes.



PCF



Na vida lá fora, tudo
parece ser uma **relação**
- relação profissional,
numérica, internacional,
familiar,
Como será possível
ensinar tudo isso a uma
máquina sem se saber
o que uma **relação** é no
sentido **formal** do
termo?

É esse o principal
objectivo desta UC.

CSI



MFP @ DIUM



DigiLightRail (EFACEC)



MFP @ DIUM

- **WWW:** <https://haslab.github.io/MFP>
- Visibilidade internacional
(<https://fme-teaching.github.io/courses>)



MFP @ DIUM

Dúvidas? Falem connosco:

[http://lmf.di.uminho.pt/
#members](http://lmf.di.uminho.pt/#members)



Testemunhos

João Pereira



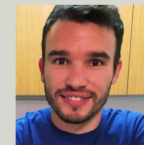
ETH-Zurich

Ter passado pelo perfil de métodos formais foi sem dúvida alguma a melhor decisão que tomei enquanto aluno de MIEI. Não só me equipou com as ferramentas de raciocínio e de programação que ainda hoje uso no dia-a-dia, como me expôs a uma área científica rica e estimulante e a uma indústria numa fase de crescimento explosivo, com oportunidades abundantes e que recompensa o empreendedorismo. Recomendo este perfil a todos os alunos!



Testemunhos

Tiago Jorge



GMV

Como engenheiro de software a trabalhar na GMV, no domínio crítico do "Espaço", tendo como principal cliente a ESA, e estando a trabalhar sobretudo em projetos de R&D, posso afirmar por experiência que desenvolvimento de sistemas e software baseado em modelos (MBSE), em que os MF se inserem como elemento chave, está entre as principais visões estratégicas da agenda europeia, e por isso praticamente todas as grandes empresas na área se estão a desenvolver nesta área.



Testemunhos



Armando Santos



WellTyped

O perfil de Métodos Formais permitiu-me adquirir bases e metodologias científicas e matemáticas importantes para o desenho, raciocínio e construção de sistemas de software precisos. Estes conhecimentos demonstraram ser uma mais valia no mercado de trabalho, distinguindo-me de outros candidatos, pela capacidade de abstração e interesse pelo rigor.



Testemunhos



Pedro Araújo



ISSU

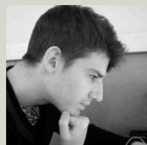
MFP (MFES) foi essencial para a minha carreira. Embora o seu uso no dia a dia não seja explícito eu sei que em todos os sistemas e código que crio o pensamento formal está lá. Compreender que programar pode ser uma ciência exata ajuda a dar a perspectiva necessária para criar sistemas robustos, bem especificados e formais.



Testemunhos



Daniel Murta



Farfetch

Engenharia de Software sob uma perspetiva cientificamente segura. Tal como uma ponte não se ergue, sem antes ter prova de que a mesma não irá ceder, também o software pode ser desenvolvido com garantias de que não falhará. O perfil MFP aborda a Engenharia de Software seguindo esta abordagem, dando ferramentas sólidas para provar e desenvolver software para além do modus operandi da indústria que se resume a cargas de testes

