Making Fairness Actionable

Julian Alfredo Mendez

Umeå University

julian.mendez@cs.umu.se
julianmendez.github.io

2025-01-16

Motivation

Ethical Machine

- How to ensure that a machine follows ethical principles?
- What ethical decisions can be automated?
- How to formally represent fairness?

Motivation

Why ethics for computers

Current computers can:

- track, monitor, and profile human beings;
- recognize speech and face.

We rely on computers because:

- our **money** is mostly (and almost only) digital;
- we upload **personal information** to central servers we do not control;
- we make **decisions** based on information we get digitally;
- we depend on **mobile devices**, which are very difficult to opt out for, and we have no superuser control.

Motivation

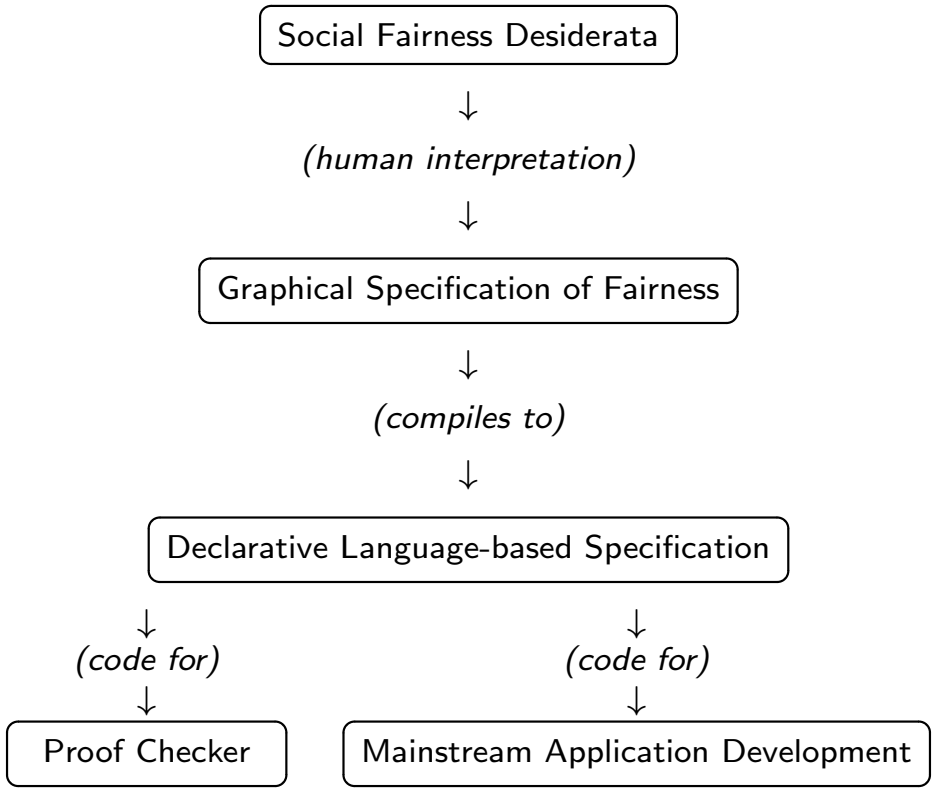
What society requires

Computer systems must

- be reliable so that society can **verify** and **control** their behavior;
- comply with **societal values** and **ethical principles** (e.g. fairness, non-discrimination, safety, and privacy).

Overview

Conceptual Overview



Tiles

We wanted to model fairness, and we started with this image:

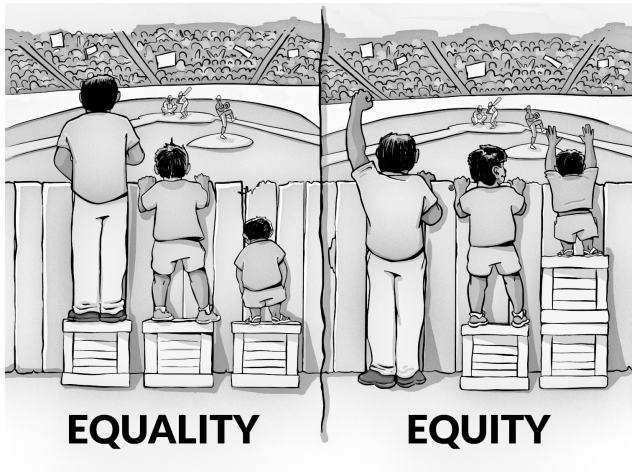


Image: Interaction Institute for Social Change

What are the essential components in a fairness scenario?
We propose:
Actor, Resource, Outcome, Measure, Aggregation, Attribute

Tiles

Tiles

We developed Tiles, a software framework to create formal configurations of constraints.

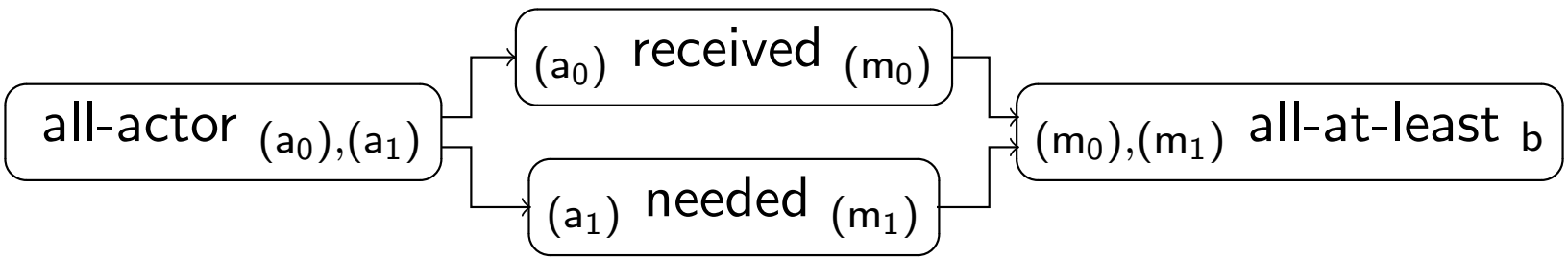
Implementation

`https://github.com/julianmendez/tiles`
(language: Soda)

Tiles

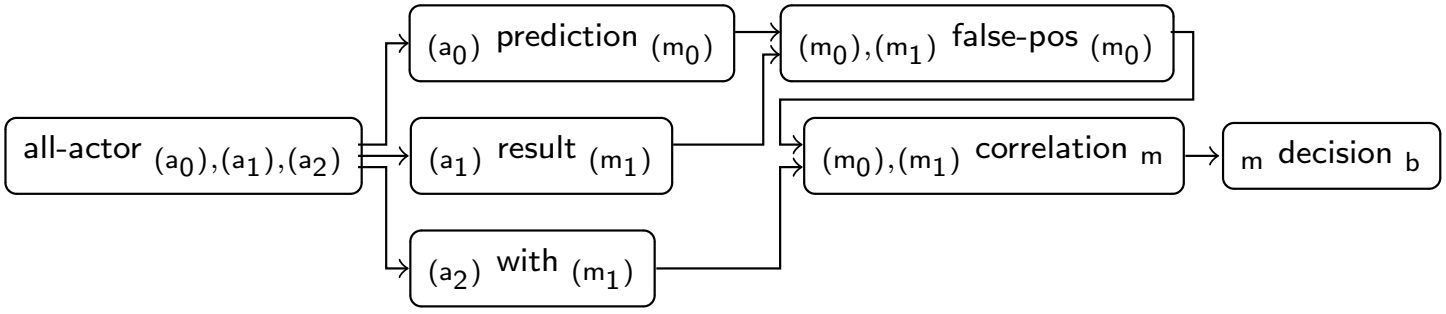
Configuration for equality

Configuration for equity



Tiles

Configuration for COMPAS (false positives)



- all-actor: creates triples of actors;
- prediction: the original prediction on an actor;
- result: the actual result of an actor;
- with: if the actor has a given (protected) attribute;
- false-pos: the false positives;
- correlation: the correlation between the sources;
- decision: whether there is a significant bias.

Conclusion and Future Work

We work on how we built a bridge to connect human values with computers. We presented a software framework for verification of constraints (Tiles)

Future Work

- Investigate extending programming languages for formal verification with proof assistants.
- Expand proof-of-concept implementations to complex scenarios.
- Analyze pros and cons of proof assistants versus model checkers.
- Apply formal verification to real-world problems (e.g., fairness in socio-technical systems).

References

Poster

Publications

- Licentiate Thesis: UMINF 24.12
- Paper I: DOI:10.1007/s10676-022-09636-z
- Paper II: DOI:10.1145/3593013.3594059
- Paper III: DOI:10.3384/ecp208013
- Paper IV: DOI:10.48550/arXiv.2310.01961
- Paper V: DiVA urn:nbn:se:umu:diva-232383

julian.mendez@cs.umu.se
julianmendez.github.io