# On the Three-way Interaction Optimizing Feature Based Systems Verification

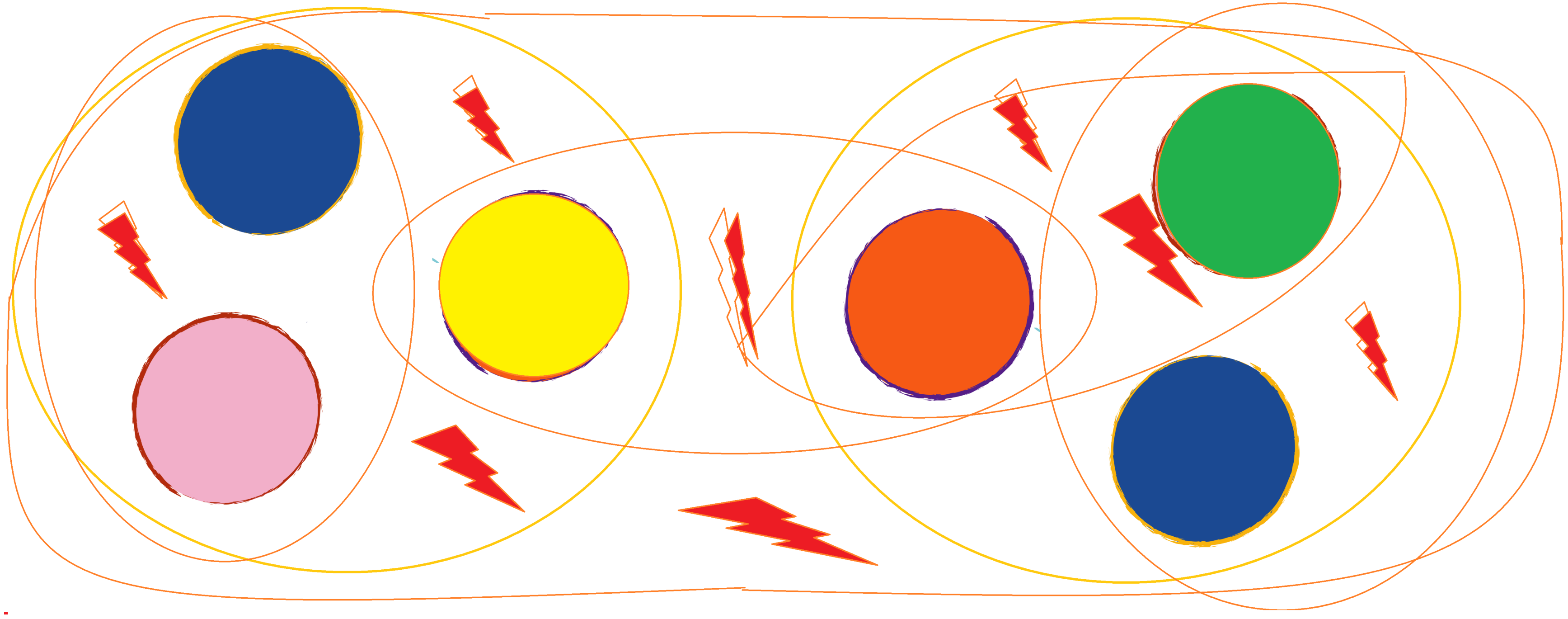
Laura Semini

joint work with Alessandro Fantechi

ISTI–CNR, Pisa, Italy
Universita' di Pisa,Italy
Universita' di Firenze,Italy

Formalise 2017, Buenos Aires

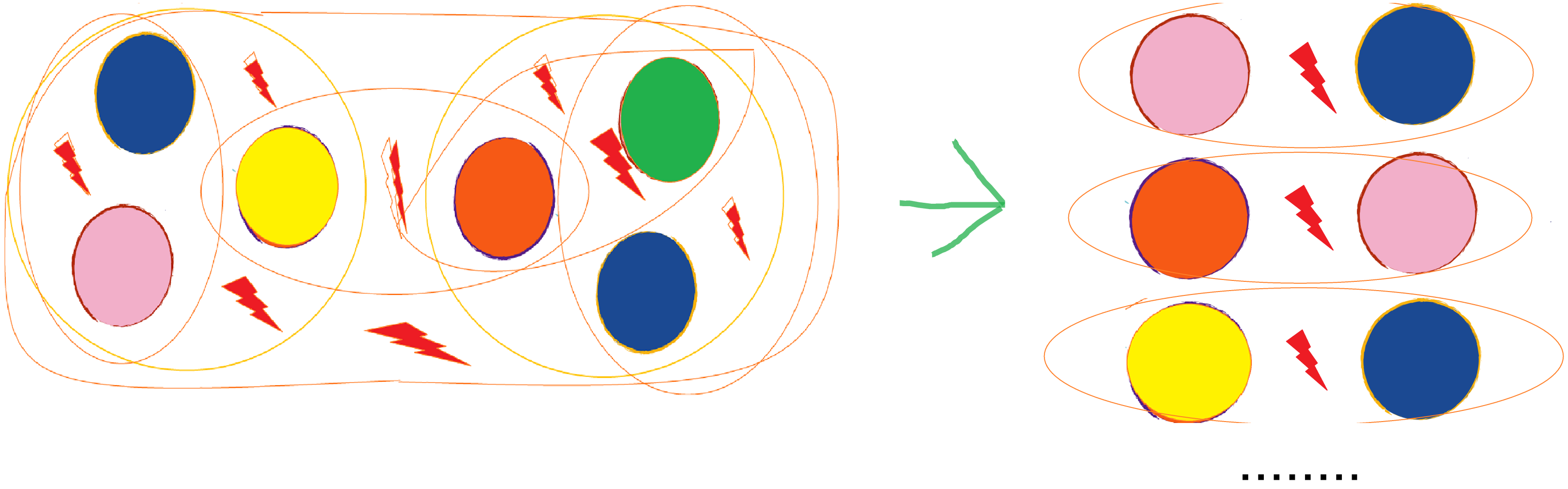
# The feature interaction problem



The feature interaction problem has been recognized as a general problem of software engineering.

> *"A feature interaction occurs when the behavior of one feature is affected by the presence of another".*

# Optimising feature interaction detection



One of the sensitive issues is the capability to make pairwise verification with respect to verify all possible combination, so to achieve a quadratic, in the number of features, cost instead of an exponential cost of a complete verification.

# The Example

**AC** **Air change** **3**
If it is not freezing, at 10:00 a.m. open the windows.

**EAC** **End of air change**
At 10:30 a.m. close the windows.

**CW** **Close window with rain**
Close the windows when the rain sensor is triggered.

**DP** **Danger prevention**
Open the main door and open the windows when smoke is sensed.

**IA** **Intruder alarm** **1**
Raise an intruder alarm when the main door is open and the alarm is ON.

**RIA** **Reation to intruder alarm** **2**
If an intruder alarm has been raises, then close all doors and windows.

# The Model for the Features

Syntax: $F = \langle C,\ [A] \rangle$

Semantics: $s \models C \Rightarrow s \xrightarrow{A} s'$

Parallel composition: interleaving $F_1 || \ldots || F_n$

$$AC = \langle \sim \textit{freezing} \wedge 10\!:\!00\ \textit{a.m.},\ [\textit{open the windows}] \rangle$$
$$IA = \langle \textit{main door open} \wedge \textit{alarm on}, [\textit{raise an intruder alarm}] \rangle$$
$$RIA = \langle \textit{intruder alarm}, [\textit{close the main door},\ \textit{close windows}] \rangle$$

# The Result

In our setting,

any 3-way interaction

is due

to the interaction between 2 of the considered features.

# Limitations

True 3-way interactions can exist when:

- Features are tags that drive conditional compilation in a java-like program.
- Features have to compete for the usage of shared (physical) resources: non-functional interactions.