COMBINATORIAL METHODS FOR TESTING AND ANALYSIS OF COMPLEX SYSTEMS

Dimitris Simos

MATRIS Research Group, SBA Research, Austria

Institute of Software Technology, Graz University of Technology, Austria

Information Technology Laboratory, National Institute of Standards and Technology, USA

Lecture Series of the Research Institute for Supply Chain Management

Winter Semester 2024/2025, October 18, 2024, WU Wien

Outline

1. Combinatorial Testing
2. Testing of **Complex Systems**
3. Future Outlook

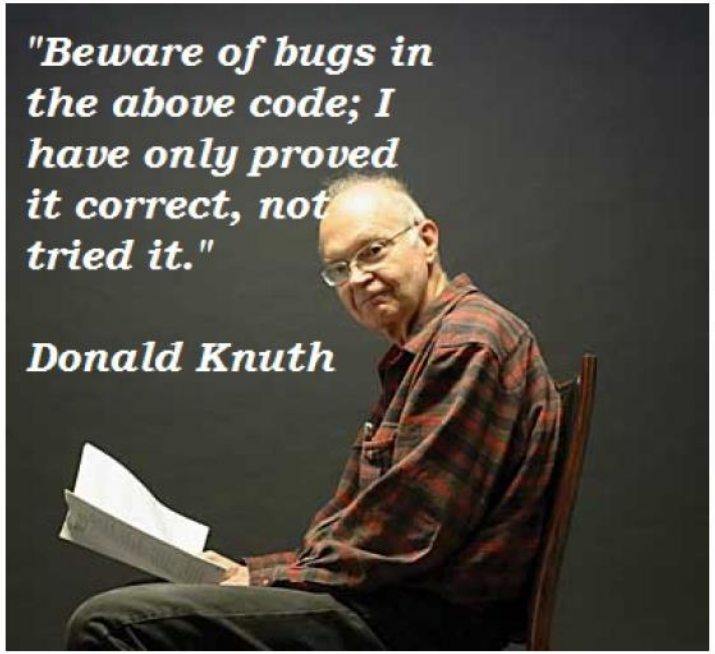
MATRIS

Combinatorial Testing

Introduction

MATRIS

Should we care about Software (Systems) Testing?



- Proving correctness seems to be not quite enough
- **Testing** is required: both on the sides of verification and validation!
  - "The process of analyzing a software system to detect the differences between existing and expected conditions (that is, bugs)" [IEEE]

Should we really care about Software (Systems) Testing?

Finding 90% of flaws is pretty good, right?

"Relax, our engineers found 90 percent of the flaws."

A Large Example for Testing

- Suppose we have a system with on-off switches
- 34 switches = 2^{34} = 1.7×10^{10} possible settings

- How do we **test** this system?

MATRIS

Example of a Mathematical Structure used in Testing

System Under Test (SUT) with 3 Boolean Input Parameters a, b, c

- Could be function, application, configuration file, etc.
- Exhaustive test set: $2^3 = 8$ tests
- 2-way covering array (test set): 4 tests

a	b	c	(a, b)	(b, c)	(a, c)
0	0	0	(0, 0)	(0, 0)	(0, 0)
0	1	1	(0, 1)	(1, 1)	(0, 1)
1	0	1	(1, 0)	(0, 1)	(1, 1)
1	1	0	(1, 1)	(1, 0)	(1, 0)

Table 1: 2-way test set (left) covering all pairs of parameters (right)

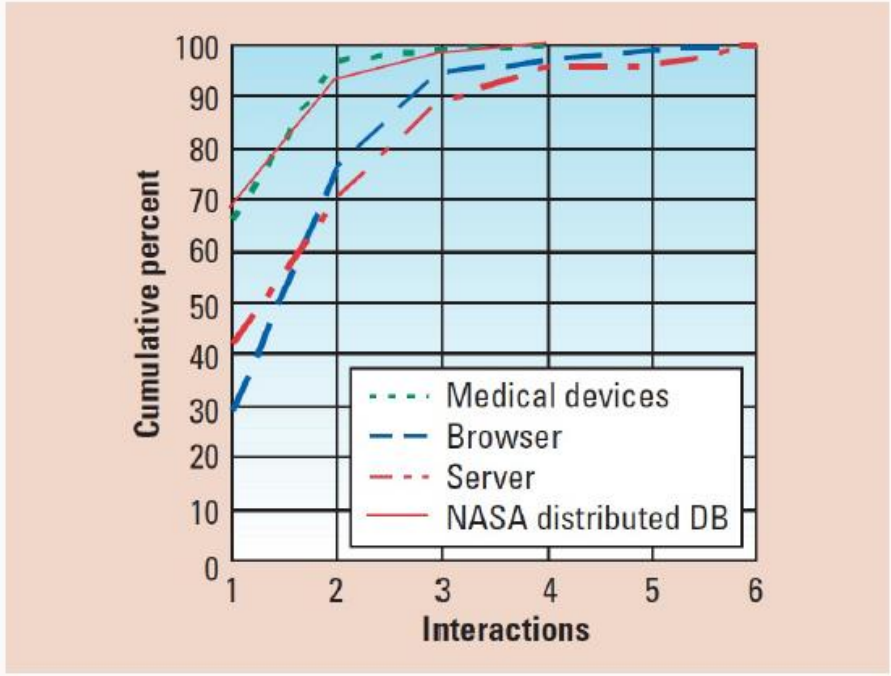
Covering Arrays $CA(N; t, k, v)$ **of Strength** t

- Cover all t-way combinations of k input parameters at least once
- Input parameters have v total values each
- Such a mathematical object has N total rows (tests)

How is this Knowledge Useful?

- Recall the system with on-off switches
- 34 switches = $2^{34} = 1.7 \times 10^{10}$ possible settings
- **Assumption: What if we knew no failure involves more than 3 switch settings interacting?**
  - If only 3-way combinations, need a CA with only 33 tests
  - If only 4-way combinations, need a CA with only 85 tests

Empirical Evidence: Fault Coverage vs. Interactions



- The maximum degree of interaction observed so far in actual real-world faults is **relatively** small (six)
  - 2-way interaction: **age** > 100 and **zip-code** = 5001, DB push fails
- Most failures are induced by single factor faults or by the **joint combinatorial effect** (interaction) of two factors, with progressively fewer failures induced by interactions between three or more factors

Combinatorial Testing (CT)

What is Combinatorial Testing?
Combinatorial Strategy for Higher Interaction Testing ($t \geq 2$)

Where it can be Applied?
To system configurations, input data or both

Key Facts:

- CT utilizes 100% coverage of t-way combinations of k input data or system configuration parameters
- Coverage is provided by **mathematical objects** (covering arrays), that are later transformed to software artifacts
- t-way tests that cover **all** such few parameter (factor) interactions can be very effective and provide strong assurance

Combinatorial Testing of a Server Configuration

Example

Application must run on any configuration of OS, browser, protocol, CPU and DBMS (very efficient for interoperability testing)

Test	OS	Browser	Protocol	CPU	DBMS
1	XP	IE	IPv4	Intel	MySQL
2	XP	Firefox	IPv6	AMD	Sybase
3	XP	IE	IPv6	Intel	Oracle
4	OS X	Firefox	IPv4	AMD	MySQL
5	OS X	IE	IPv4	Intel	Sybase
6	OS X	Firefox	IPv4	Intel	Oracle
7	RHEL	IE	IPv6	AMD	MySQL
8	RHEL	Firefox	IPv4	Intel	Sybase
9	RHEL	Firefox	IPv4	AMD	Oracle
10	OS X	Firefox	IPv6	AMD	Oracle

Figure: Pairwise test configurations

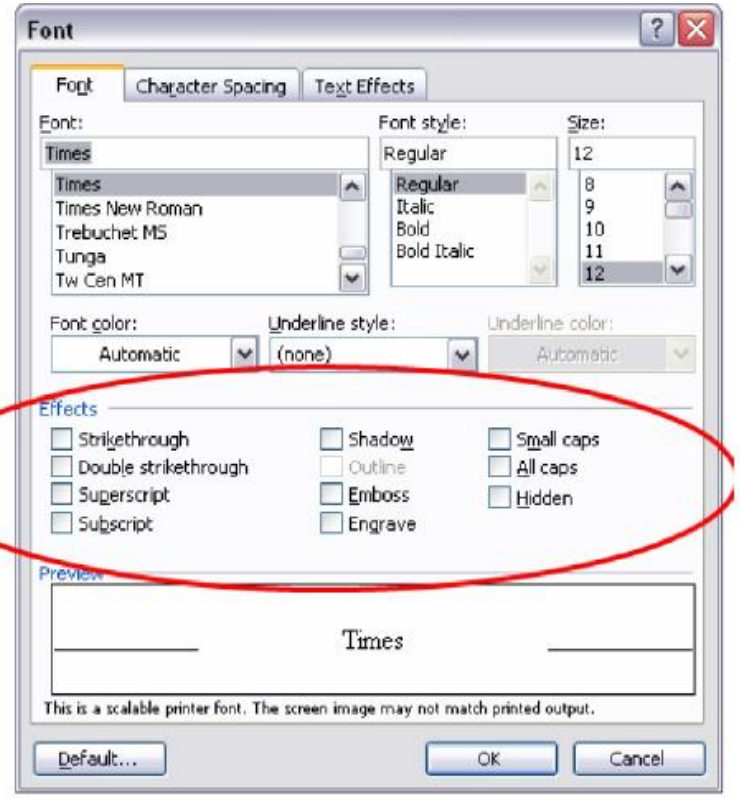
MATRIS

Combinatorial Testing of a Word-Processing App

Example

Testing of a word-processing application having 10 effects to highlight text (each can be on or off)

0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	0	0	1
1	0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	0	0
0	1	1	0	0	1	0	0	1	0
0	0	1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0	1	0
0	0	0	1	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1
0	1	0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	1	1	1
0	1	0	0	0	1	1	1	0	1



0 = effect off
1 = effect on

13 tests for **all 3-way combinations**

2^{10} = **1,024** tests for **all combinations**

MATRIS

Motivation for Combinatorial Testing of Complex Systems

- **Economic Impact:** Software testing may consume up to half of the overall software development cost ("system of systems view")
  - **Combinatorial explosion:** Exhaustive search of input space increases time needed exponentially
  - Added level of complexity for system testing (modelling real-world environments)
  - CT can provide **minimal tests** which provides for ~99% reduction of test set sizes => Reduced **testing budget** by several orders of magnitude => significantly less **costs**
- How can we estimate the residual risk that remains after testing? How can we guarantee aspects of test quality (e.g. test coverage, locating faults)?

- **In this Talk:** Formulate testing problems as combinatorial problems and then use efficient methods to tackle them

MATRIS

Testing of Complex Systems

R&D Examples in System Testing

Testing of an F-16 Lantirn Pod

- **Problem:** Unknown factors causing failures of F-16 ventral fin
- LANTIRN pod carriage on the F-16

MATRIS

F-16 Ventral Fin Damage on Flight with LANTIRN

It's not supposed to look like this:

MATRIS

Input Model for Testing of F-16 Ventral Fin

- **Original solution:** Lockheed Martin engineers spent many months with wind tunnel tests and expert analysis to consider interactions that could cause the problem
- **CT solution:** modelling and simulation using ACTS/CAgen

Parameter	Values
Aircraft	15, 40
Altitude	5k, 10k, 15k, 20k, 30k, 40k, 50k
Maneuver	hi-speed throttle, slow accel/dwell, L/R 5 deg side slip, L/R 360 roll, R/L 5 deg side slip, Med accel/dwell, R-L-R-L banking, Hi-speed to Low, 360 nose roll
Mach (100th)	40, 50, 60, 70, 80, 90, 100, 110, 120

MATRIS

Combinatorial Testing for Aerospace Industry

- **Input Model for CT**
  - 4 Parameters
  - Number of values: 2, 7, 9, 9
  - Total space (exhaustive testing): 1134 tests

- **Combinatorial Testing**
  - 2-way: 81 tests (=> ~93% reduction)
  - 3-way: 567 tests (=> 50% reduction)

- **How does this translate in real-life Budget Costs?**
  - Costs of executing one test:
    - A Lockheed engineer costs $ 75 / hour
    - 3 engineers / 1 day (10hours each) => $ 2,250
  - Costs of executing exhaustive test set: $ 2,551,500
  - Combinatorial-based testing (NIST study) costs:
    - 2-way: $ 182,250 => only ~7% of total costs
    - 3-way: $ 1,275,750 => only 50% of total costs

Aircraft	Altitude	Maneuver	Mach (100th)
15	5k	hi-speed throttle	40
40	10k	hi-speed throttle	50
15	15k	hi-speed throttle	60
40	20k	hi-speed throttle	70
15	30k	hi-speed throttle	80
40	40k	hi-speed throttle	90
15	50k	hi-speed throttle	100
40	5k	hi-speed throttle	110
15	10k	hi-speed throttle	120
40	15k	slow accel/dwell	40
15	20k	slow accel/dwell	50
40	30k	slow accel/dwell	60
15	40k	slow accel/dwell	70
40	50k	slow accel/dwell	80
15	5k	slow accel/dwell	90
40	10k	slow accel/dwell	100
15	15k	slow accel/dwell	110
40	20k	slow accel/dwell	120
15	30k	L/R 5 deg side slip	40
40	40k	L/R 5 deg side slip	50

Excerpt of 2-way test set
(demo, if time allows)

Virtual Driving Function Testing Problem

TECH

Tesla must provide NHTSA with Autopilot recall data by July or face up to $135 million in fines

PUBLISHED TUE, MAY 7 2024•5:08 PM EDT | UPDATED TUE, MAY 7 2024•7:04 PM EDT

Lora Kolodny
@IN/LORAKOLODNY/

CNBC

SHARE

WIKIPEDIA
The Free Encyclopedia

Search Wikipedia | Search

List of Tesla Autopilot crashes

Add languages

Contents hide

(Top)

Fatal crashes

Handan, Hebei, China (January 20, 2016)

Williston, Florida, USA (May 7, 2016)

Mountain View, California, USA (March 23, 2018)

Kanagawa, Japan (April 29, 2018)

Delray Beach, Florida, USA (March 1, 2019)

Key Largo, Florida, USA (April 25, 2019)

Fremont, California, USA (August 24, 2019)

Article Talk | Read Edit View history Tools

From Wikipedia, the free encyclopedia

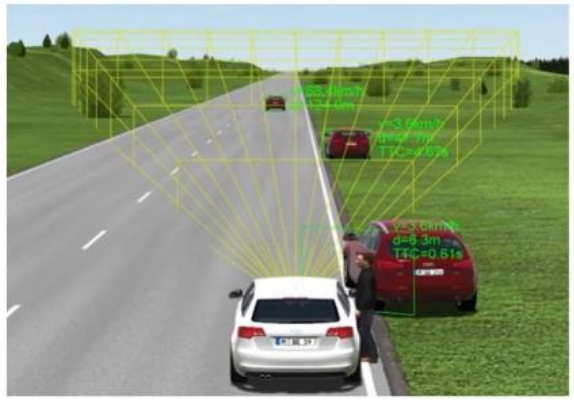
Tesla Autopilot was released in October 2015, and the first fatal crashes involving the advanced driver assistance system (ADAS) occurred less than one year later. The fatal crashes have attracted attention from news publications and United States government agencies, including the National Transportation Safety Board (NTSB) and National Highway Traffic Safety Administration (NHTSA), which has argued the Tesla Autopilot death rate is higher than the reported estimates.[1] In addition to the fatal crashes, there have been many nonfatal crashes, the causes have included the ADAS failing to recognize other vehicles, insufficient Autopilot driver engagement, and violating the operational design domain.

The model S after it was recovered from the crash scene in Williston, Florida

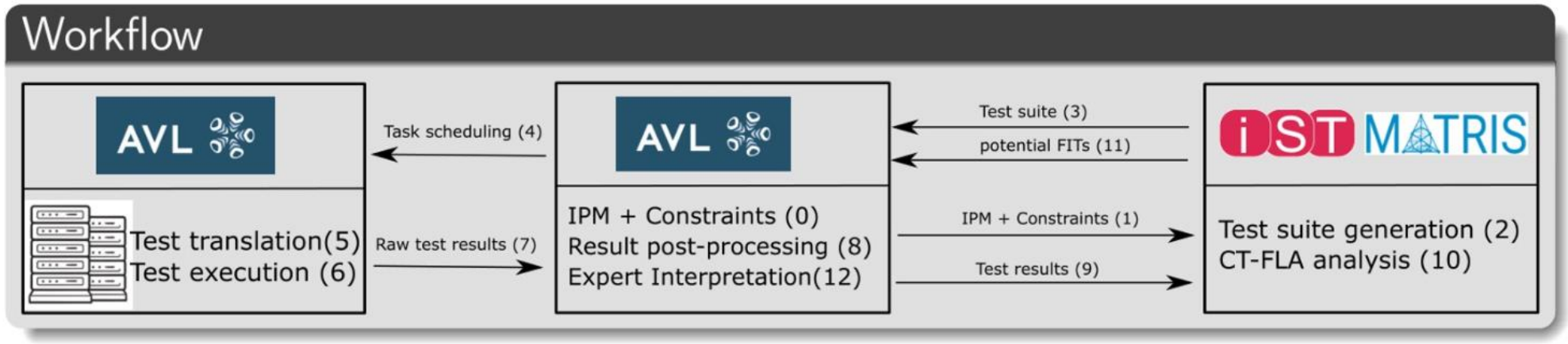
As of June 2024, there have been forty-four verified fatalities involving Autopilot[2] and hundreds of nonfatal incidents.[3] Collectively, these have led to a formal investigation by the NHTSA, culminating in a general recall in December 2023 of all vehicles equipped with Autopilot, which was resolved by an over-the-air software update. Immediately after closing its investigation in April 2024, NHTSA opened a recall query to determine the effectiveness of the recall.

AEB Function Testing

Mapping and testing workflow



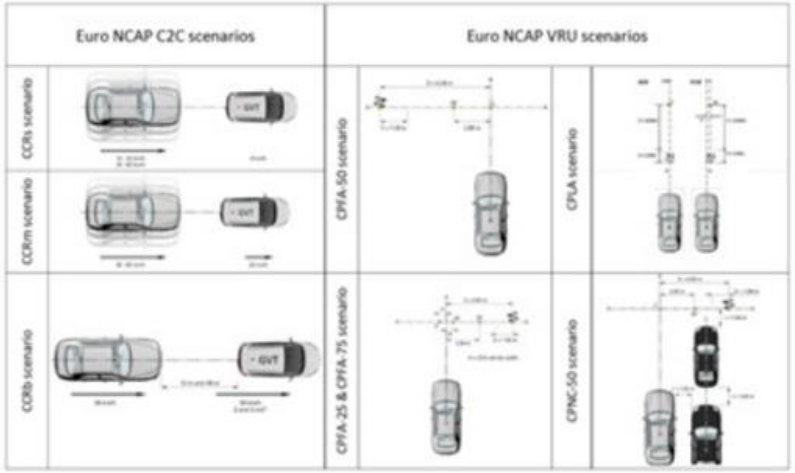
Combinatorial Testing	Automated Driving Function Testing
System under test (SUT)	AEB function
Input-parameter model (IPM)	Driving scenario specification
Test/row of CA	Driving scenario
Test execution	Simulation of driving scenario
Execution oracle (oracle)	Time-to-collision (TTC)
Passing test	Non-crash scenario ($TTC > 0$)
Failing test	Crash scenario ($TTC = 0$)
Failure inducing t-way interaction (FIT)	Crash inducing parameter setting



Virtual Driving Scenarios

IPM for Driving Scenarios

- Developed and used in previous works (Wotawa et al.[a])
- Description of traffic situation via parameters + values
  - Speed of the car: EgoVehicle1_Start_speed
  - Type of car: EgoVehicle1_VehicleType
  - Position of the car: EgoVehicle1_Offset_s
  - Number of pedestrians: NumberOfPede
- Resulting IPM consists of 39 parameters & 42 constraints:



(3, 3, 31, 3, 5, 1, 6, 4, 12, 12, 12, 10, 14, 12, 12, 12, 10, 14, 31, 4, 3, 20, 9, 3, 3, 31, 4, 3, 20, 9, 3, 3, 31, 4, 3, 20, 9, 3, 3)

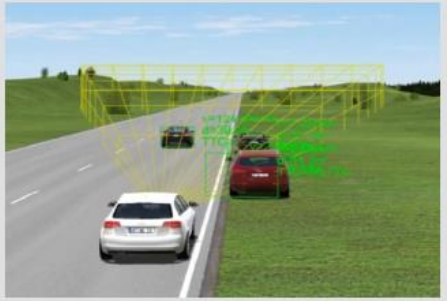
```
[Parameter]
NumberOfVehiclePlayer (enum) : 1, 2, 3
NumberOfPede (enum) : 1, 2, null
EgoVehicle1_Start_speed (enum) : 0, 1.388888889 ,2.777777778  ,4.166666667  ,5.555555556  ,6.944444444  ,8.333333333  ,9.722222222  , ...
EgoVehicle1_VehicleType (enum) : Audi_A3_2009_white, Audi_Q5_2008_red, Audi_S5_2009_black metallic
EgoVehicle1_Offset_s (enum) : -2,-1,0, 1, 2
EgoVehicle1_Offset_t (enum) : 0
EgoVehicle1_Rate (enum) : 5, 6, 7, 8, 9, 10
EgoVehicle1_Target_Speed (enum) : 14, 28, 42, 55
Pedestrians_Objects1_Start_speed (enum) : 0, 0.28, 0.56, 0.83, 1.11, 1.39, 1.6, 1.94, 2.22, 2.5, 2.8, null
Pedestrians_Objects1_Offset_s (enum) : 3, 4, 5,6,7,8,9,10,11,12,13, null
Pedestrians_Objects1_Offset_t (enum) : 10,11,12,13,14,15,16,17,18,19,20, null
Pedestrians_Objects1_Rate (enum) : 0, 1, 2, 3, 4, 5, 6, 7, 8, null
Pedestrians_Objects1_Type (enum) : Adult, Child, Wheelchair, Animal, Ballon, Paper, Dog, Stone, Adult_w_Bicycle, Adult_w_child, custom1, custom2, custom3, null
Pedestrians_Objects2_Start_speed (enum) : 0, 0.28, 0.56, 0.83, 1.11, 1.39, 1.6, 1.94, 2.22, 2.5, 2.8, null
Pedestrians_Objects2_Offset_s (enum) : -9,-10,-11,-12,-13,-14,-15,-16,-17,-18,-19,null
```

Analysis of Potential Crash Scenarios

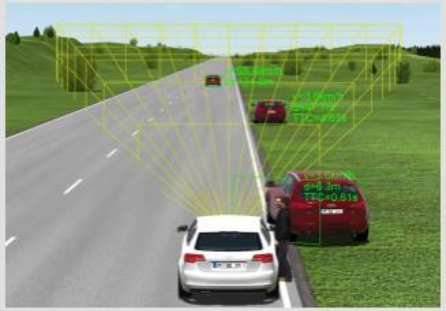
- We generated 39061 driving scenarios of which **7928 are failing (high percentage of failing test cases)**
- In **>** 46 tests specific **3-way failure patterns** appear (from a CT point of view)
- An AVL engineer reviewed these 46 CT generated scenarios and evaluated them visually (e.g. ones where EgoVehicleOffset=2, Pedetstrian1StartSpeed={0.56, 0.28}, Pedestrian1OffSet={3,4} appear)

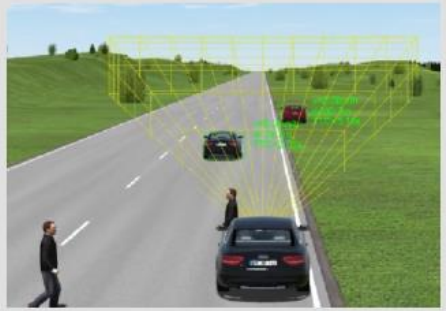
Relative Frequency Analysis

EgoVehicleOffset=2
Pedestrain1StartSpeed=0.56
Pedestrain1Offset=4



EgoVehicleOffset=2
Pedestrain1StartSpeed=0.28
Pedestrain1Offset=4

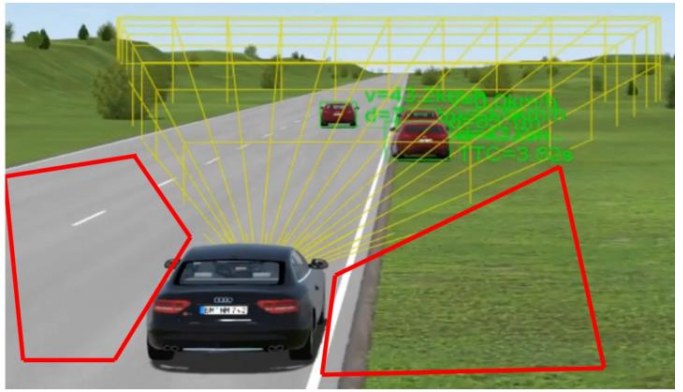




EgoVehicleOffset=2
Pedestrain1StartSpeed=0.56
Pedestrain1Offset=3

Comments from the domain expert

- Large majority of crashes are side-wards
- Simulation uses ideal object sensor
  - No detection issues if object field of view (yellow)
- Objects outside this area (red area) are not detected.
- Scenario specifications, i.e. parameter values that impact the position (offset) of the ego vehicle and relevant objects to the side of the ego vehicle contribute more towards outcomes with a collision.

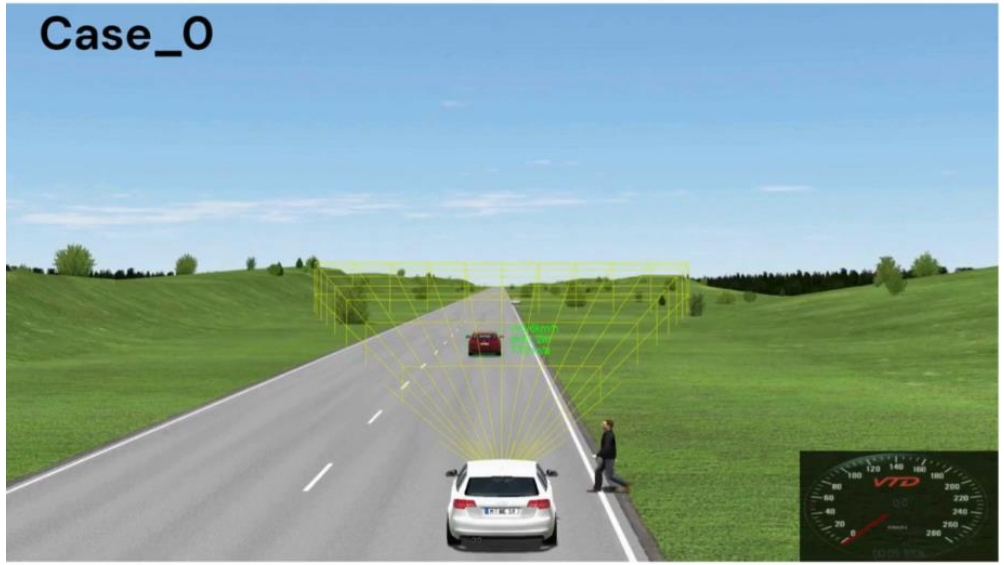


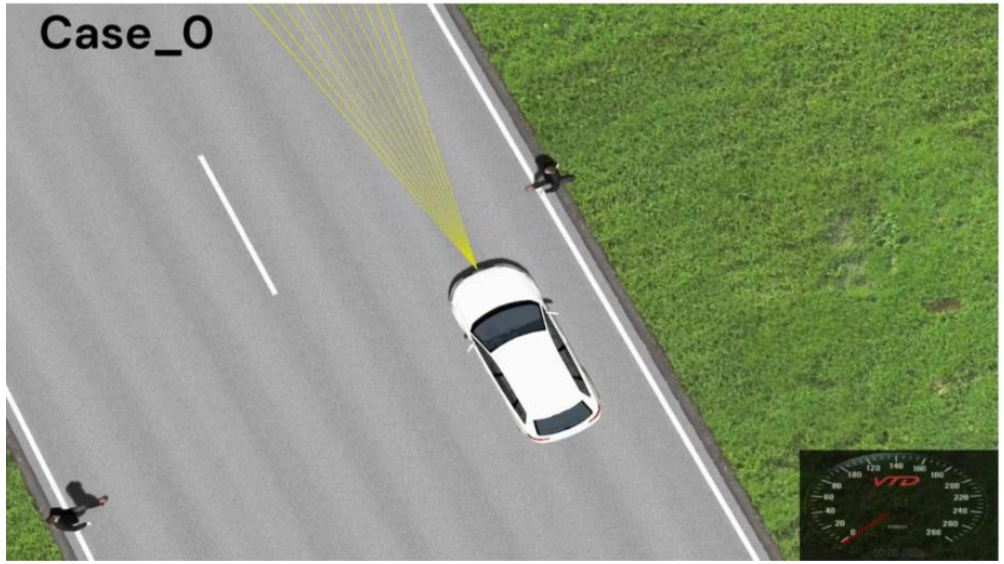
Inspection of Individual Crash Scenario

One Factor at a Time (OFAT) Strategy

- Consider single crash scenario
- Change every parameter to all other values one by one
- $\Rightarrow \geq$6-way interaction identified
  - $(-,-,-,-,1,-,-,-,-,-,1.39,-,-,-,-,10,20,-,-,\text{-}18,-,-,-,-,-,-,-,-,-,-,65,-,-,-,-,-,-,-,-,-,-)$
  - This 6-way interaction appears in millions of tests (roughly 10^{26})

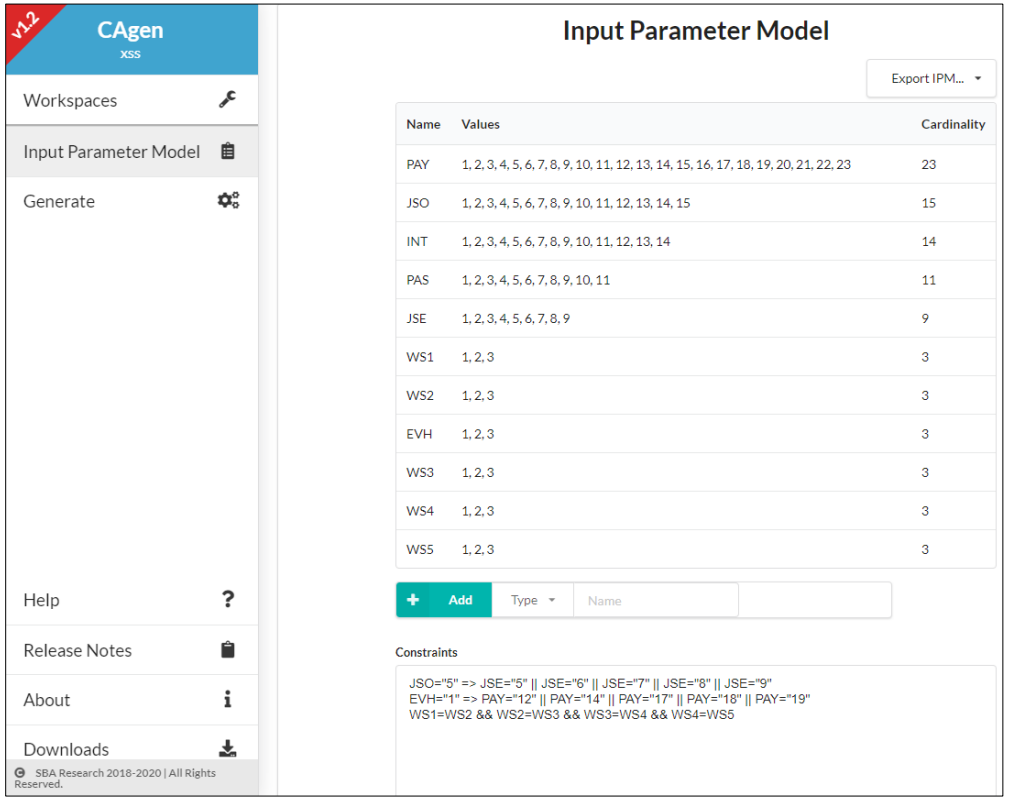
EgoVehicle1_Offset_s	Pedestrains_Objects1_Start_speed	Pedestrains_Objects1_Offset_s	Pedestrains_Objects1_Offset_t	Pedestrains_Objects2_Offset_s	Vehicles_Players2_Offset_t	Oracle
1	1.39	10	20	-18	65	crash
0	1.39	10	20	-18	65	non-crash
1	0	10	20	-18	65	non-crash
1	1.39	3	20	-18	65	non-crash
1	1.39	10	10	-18	65	non-crash
1	1.39	10	20	-9	65	non-crash
1	1.39	10	20	-18	15	non-crash





From Technology to Practice: CAGen Test Generation Tool (Greedy/Parallel/Quantum Computing Algorithms)

MATRIS



FFG Promoting Innovation.

SUCCESS STORY

SBA-K1
SBA Research GmbH

Programme: COMET – Competence Centers for Excellent Technologies

Programme line: COMET-Centre K1

Type of project: strategic

SUPERIOR SOFTWARE TESTING EFFICIENCY WITH QUALITY GUARANTEES

TOOL COMPETITION AT LEADING COMBINATORIAL TESTING CONFERENCE CROWNS CAGEN AS BEST GENERAL-PURPOSE COVERING ARRAY GENERATOR

IBM Z DELL SIEMENS LOCKHEED MARTIN Deutsche Bank DB NASA JPL

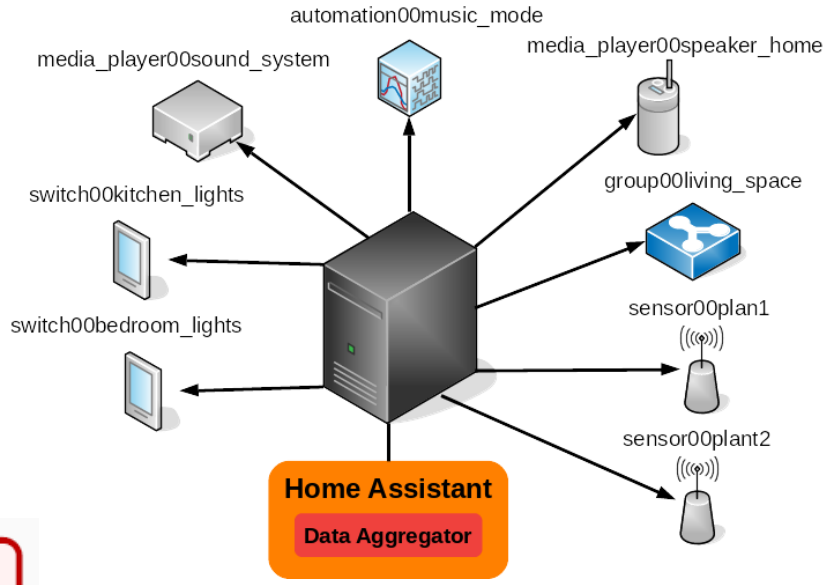
Combinatorial Security Testing (CST)

Joint Research Programme MATRIS/US NIST

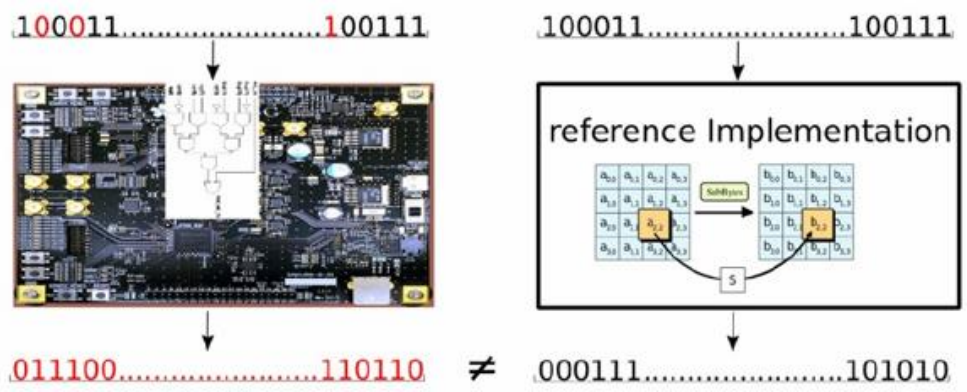
MATRIS

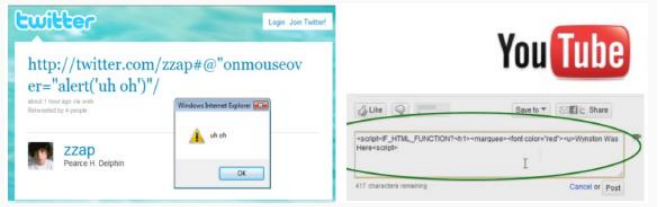
Proven method: automated testing for security

- Complex web applications
- Next generation protocol testing (IoT: z-wave, zigbee)
- Intelligent and autonomous systems
- Hardware Trojan Horse detection



Combinatorial methods can make **software security testing** much more **efficient** and effective than conventional approaches





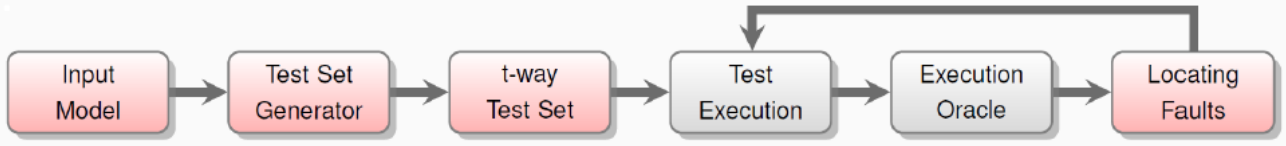
JSO	WS1	INT	WS2	EVH	WS3	PAY	WS4	PAS	WS5	JSE
"><script>	␣	';	␣	onError=	␣	alert(1)	␣	'>	␣	\>
"><script>	␣	'>	␣	onError=	␣	alert(1)	␣	'>	␣	\>
"><script>	␣	';	␣	onError=	␣	src="invalid"	␣	'>	␣	\>
"><script>	␣	'>	␣	onError=	␣	src="invalid"	␣	'>	␣	\>

MATRIS

Large-scale Combinatorial Testing at Adobe

a	b	c	(a, b)	(b, c)	(a, c)
0	0	0	(0, 0)	(0, 0)	(0, 0)
0	1	1	(0, 1)	(1, 1)	(0, 1)
1	0	1	(1, 0)	(0, 1)	(1, 1)
1	1	0	(1, 1)	(1, 0)	(1, 0)

Table: 2-way test set (left) covering all pairs of parameters (right)



Simplified testing process (CT-dependent parts in red) for given **SUT**

Combinatorial Testing applied to Large-scale Data Processing at Adobe

- Application of largest combinatorial test sets documented in research
- Test sets combined from theoretical and algorithmic constructions
- New faults found in each subject systems; small number of tests

Adobe Analytics

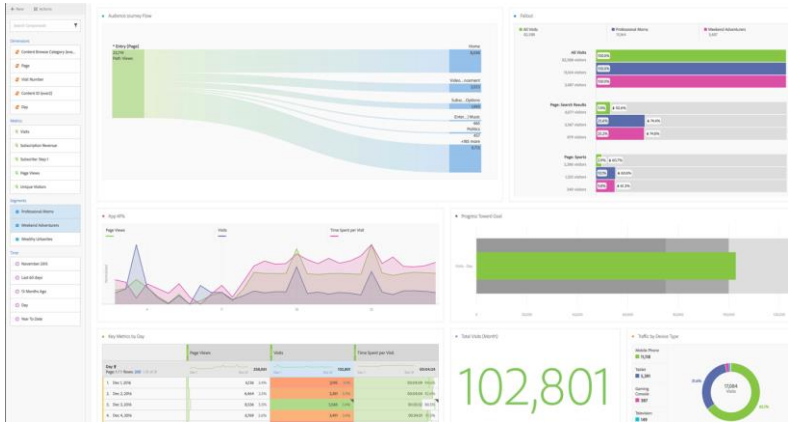
N	t	k	v
6337	4	2127	3
107514	5	2127	3
87669	5	2127	3
322	2	2127	7
7439	3	2127	7
688	2	2127	10
23422	3	2127	10

MATRIS

SBA Research

NIST National Institute of Standards and Technology

	Fault Descriptions, Causes, and Resolutions		
Description	*t-way*	*Cause*	*Resolution*
Flag-type fields throw error	2	Undocumented value constraint	Update input space model
Event-type fields throw error	2	Undocumented format constraint	Update input space model
Parser throws error (CDS)	2	Undocumented value constraint	Update input space model
Parser throws error (JSON)	3	Undocumented format constraint	Add input validation
Invalid date fields interaction	2	Undocumented value constraint	Update input space model

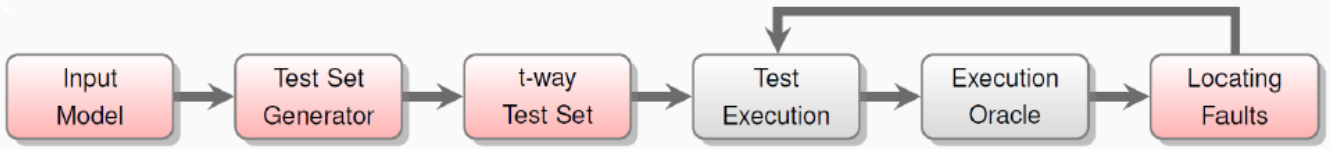


Adobe

Large-scale Combinatorial Testing at NASA

a	b	c	(a, b)	(b, c)	(a, c)
0	0	0	(0, 0)	(0, 0)	(0, 0)
0	1	1	(0, 1)	(1, 1)	(0, 1)
1	0	1	(1, 0)	(0, 1)	(1, 1)
1	1	0	(1, 1)	(1, 0)	(1, 0)

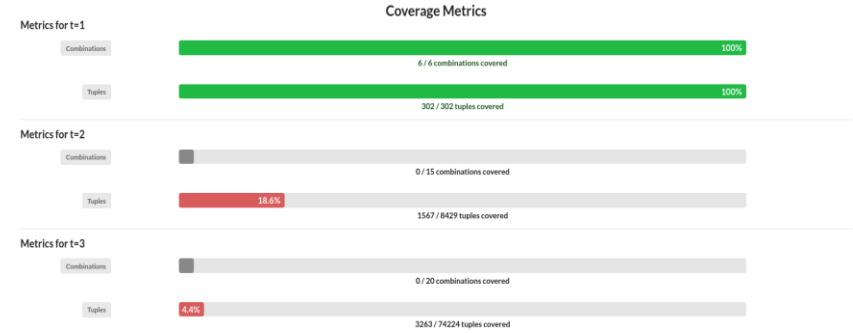
Table: 2-way test set (left) covering all pairs of parameters (right)

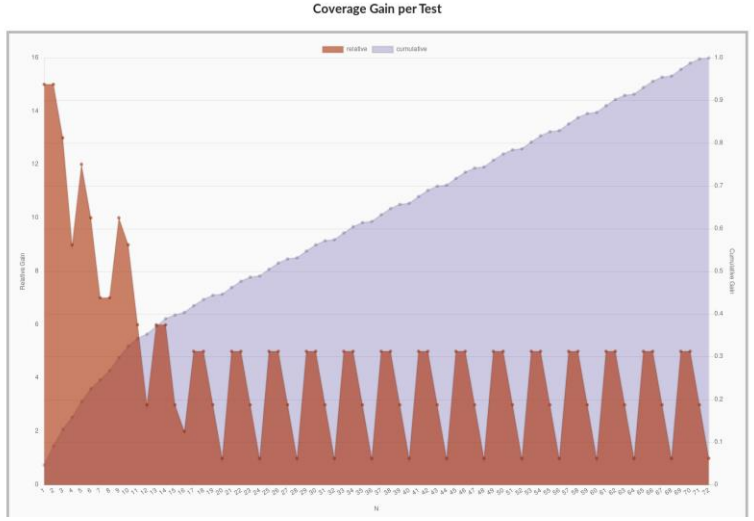


Simplified testing process (CT-dependent parts in red) for given **SUT**

Extraction of Input Parameter Models from existing unit tests in NASA's Core Flight System (cFS) software

- Generation of test cases using combinatorial means (i.e. *t-way testing*) and their execution
- Combinatorial coverage of existing tests





MATRIS

Future Outlook

Combinatorial Methods beyond System Testing

Combinatorial Methods **beyond** System Testing
Pattern identification in other research domains

MATRIS

- **Patterns appear in Every Instance of a Complex System**
- [recap] Software (Systems) Engineering (Software Systems)
  - Software/System Failures => t-way Faults (CT)

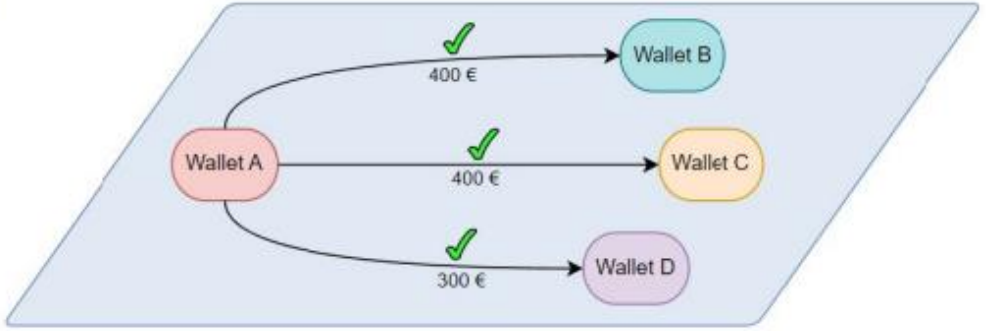
- FinTech (Financial Transactions)
  - Money Laundering Transactions => Integer Partitions
- Operations Research (Supply Chains)
  - Crisis Scenarios in Production Facilities => Seq. Covering Arrays
- Disaster Management (Natural/Technological Hazards)
  - Compound and Cascading Effects in Disasters => Permutation Sequences

MATRIS

Combinatorial Methods for (Anti-) Money Laundering

Problem: Given an amount of money in some currency and some regulations, find all IPs of the corresponding integer with properties such that the resulting transactions are not affected by applicable regulations.

(A) Transaction amount above a threshold of 1,000 Euro triggering an alert.



(B) Transaction amounts below a threshold of 1,000 Euro avoid detection.

```
x=iter(Partitions(1000, min_part=10, max_part=200, min_length=5, max_length=20))
```

- Integer partitions (IPs) visualizing splitting of amounts (Cryptocurrencies, FIAT) that are below of threshold of 1000 (Satoshi or EUR)
- Such patterns comprised of integer partitions model money laundering transactions

$$[200, 200, 200, 200, 200], [200, 200, 200, 200, 190, 10],$$
$$[200, 200, 200, 200, 189, 11], [200, 200, 200, 200, 188, 12],$$
$$[200, 200, 200, 200, 187, 13], [200, 200, 200, 200, 186, 14],$$
$$[200, 200, 200, 200, 185, 15], [200, 200, 200, 200, 184, 16],$$
$$[200, 200, 200, 200, 183, 17], [200, 200, 200, 200, 182, 18].$$

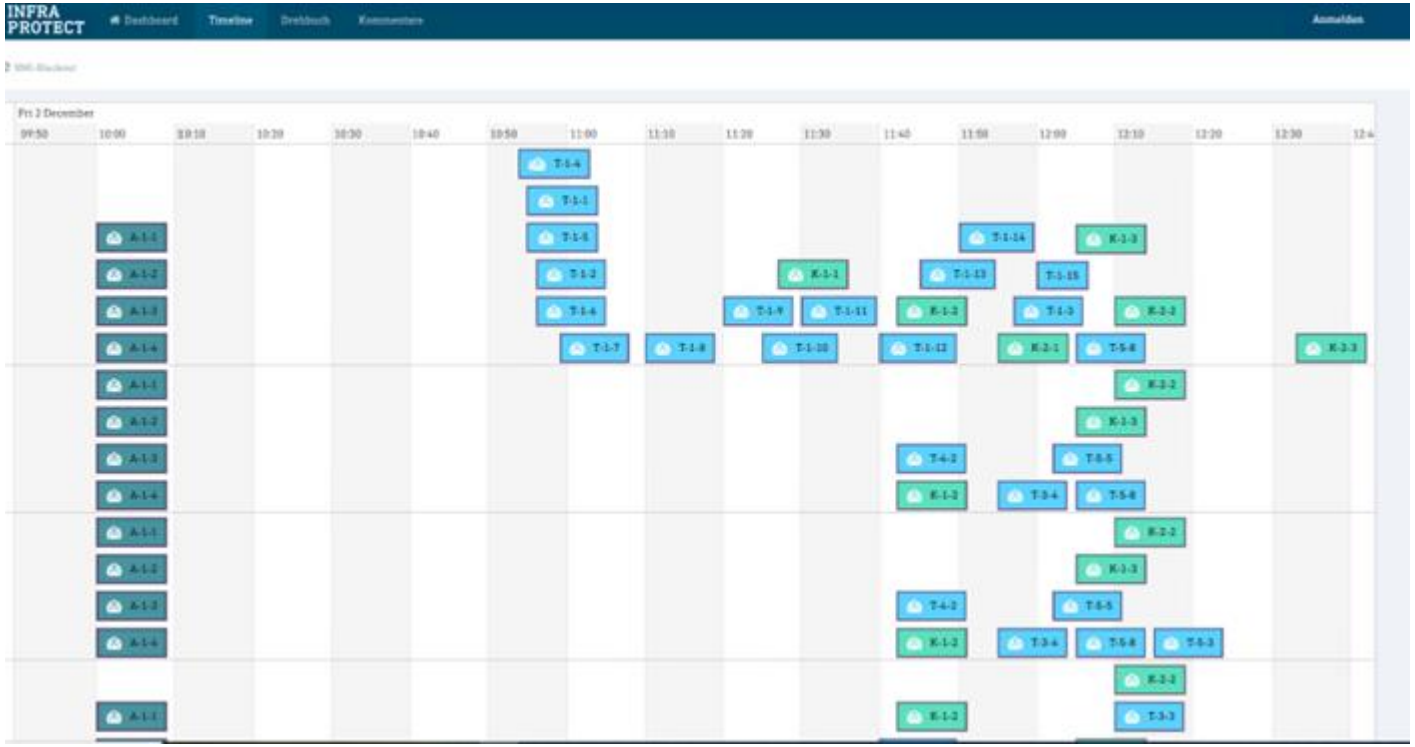
Combinatorial Generation of (Cyber-) Threat Scenarios for the Steel Industry

MATRIS

```
[Ausfall_MAs, Proteste, Sperre_Graz_Ost, Ausfall_MAs, Rad_KFV, Proteste, OEBB]
[Ueberfuellung_Lager, Sperre_Graz_Ost, Ausfall_MAs, WrLinien_LinzAG, Ueberfuellung_Lager, Rad_KFV, Ausfall_MAs]
[Ueberfuellung_Lager, Datenportal_Autobauer, Proteste, Ueberfuellung_Lager, Cyberattacke_CS, WrLinien_LinzAG, LKW_Traktor]
[Ueberfuellung_Lager, PKW_Glockner, Sperre_Graz_Ost, Ausfall_MAs, Datenportal_Autobauer, Rad_KFV, Ueberfuellung_Lager]
[Ueberfuellung_Lager, Cyberattacke_DDOS, Sperre_Graz_Ost, Ausfall_MAs, LKW_Traktor, Ueberfuellung_Lager, OEBB]
[Cyberattacke_CS, Sperre_Graz_Ost, Ausfall_MAs, PKW_Glockner, Cyberattacke_CS, Cyberattacke_DDOS, WrLinien_LinzAG]
```

Scenario 1	T-1-1	T-1-2	T-1-3	T-1-4	T-1-5	T-1-6	K-1-1
Scenario 2	T-1-2	T-1-1	K-1-6	K-1-5	K-1-4	K-1-3	K-1-2
Scenario 3	T-1-3	K-1-6	T-1-1	T-1-2	T-1-4	K-1-5	T-1-5
Scenario 4	T-1-4	K-1-6	T-1-3	K-1-2	T-1-2	K-1-1	T-1-1
Scenario 5	T-1-5	K-1-5	T-1-1	T-1-3	T-1-4	T-1-2	K-1-6
Scenario 6	K-1-1	T-1-6	T-1-2	K-1-6	T-1-3	K-1-5	T-1-5

INFRA PROTECT®



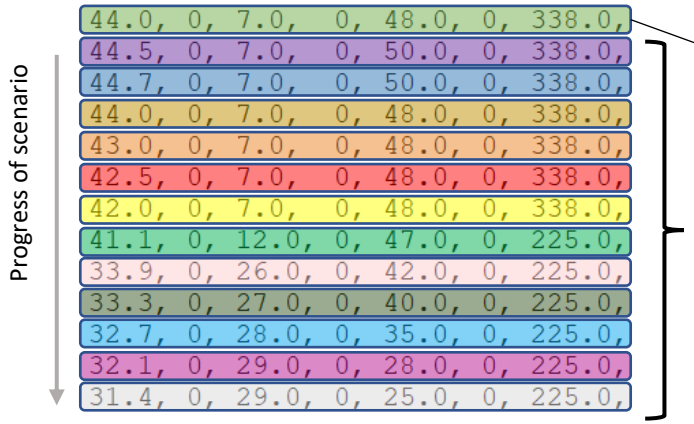
Real-world Crisis Scenario in the Steel Industry:

- Production facility faces both physical and cyber-threats
- Crisis exercise tests resilience of response plans coordinated by Q&A-department
- Combinatorial generation methods can model **cascading effects** in such threat-scenarios that can cause response plan to fail ➡ **production stop** causes massive financial loss

Scenario Generation by Permutation of Weather States

MATRIS

Original sequence of weather states

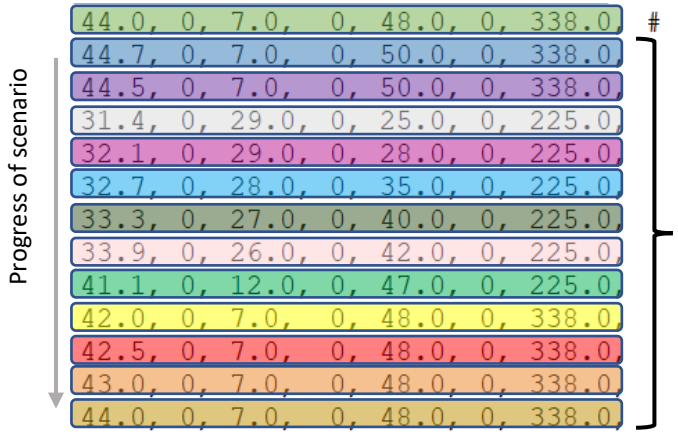


Starting state (fixed)

Sequence Covering Array (strength 3, cardinality 12)

A	B	C	D	E	F	G	H	I	J	K	L
B	A	L	K	J	I	H	G	F	E	D	C
C	L	A	B	D	K	E	J	F	I	G	H
D	L	C	H	B	G	A	I	J	K	E	F
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…

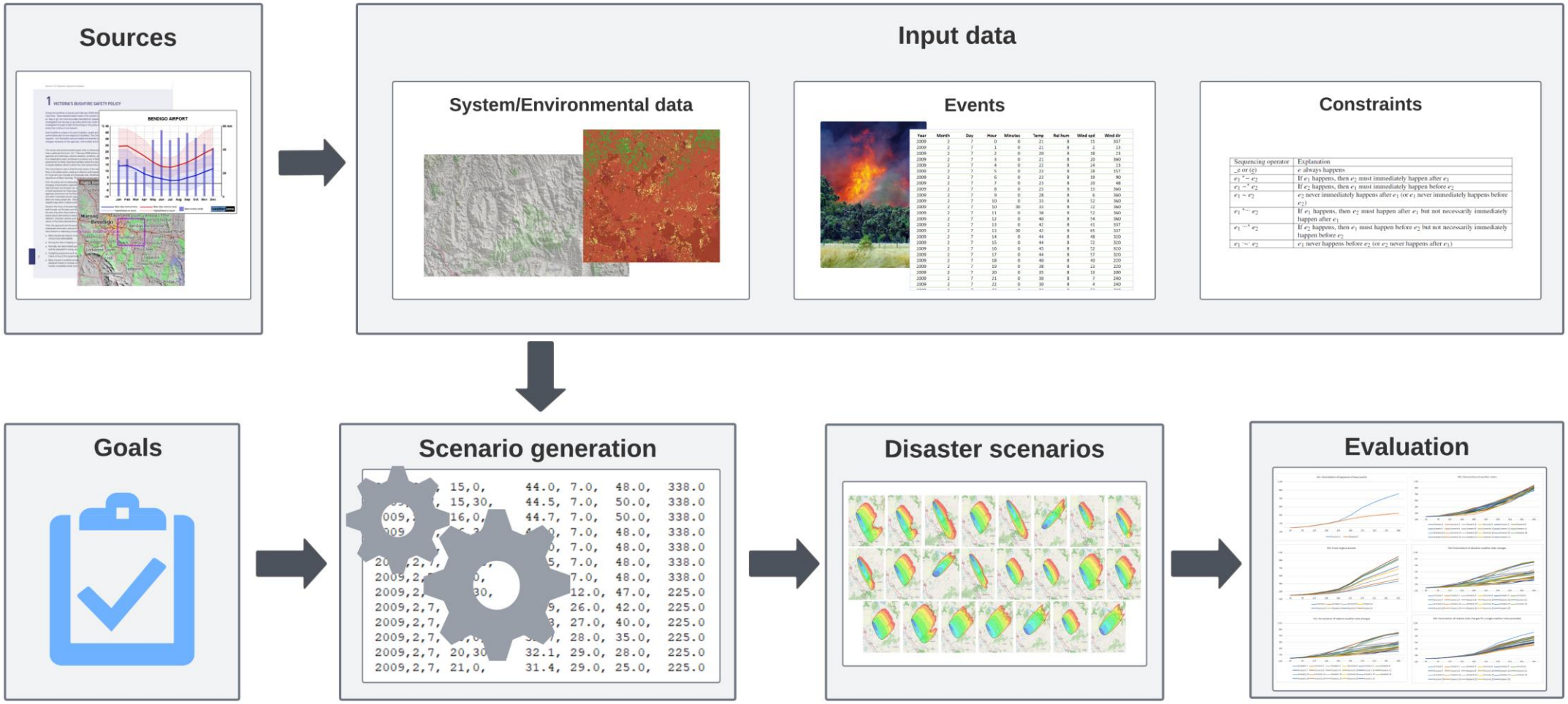
New sequence of weather states



A	B	C	D	E	F	G	H	I	J	K	L
B	A	L	K	J	I	H	G	F	E	D	C
C	L	A	B	D	K	E	J	F	I	G	H
D	L	C	H	B	G	A	I	J	K	E	F
…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…

MATRIS

Instantiation of Combinatorial Scenarios for Fire Simulation



MATRIS

Simulation: Comparison of two bushfire scenarios

Joint work with CSIRO, Australia

Integration of CT research into Teaching, Outreach and Policy Making

Course	Title	University
VU 716.204	Selected Topics of Software Technology: Quantum Computing (Lectures and Exercises)	TU Graz
VO 716.201	Selected Topics in Computer Science: Combinatorial Testing (Lectures)	TU Graz
UE 716.202	Selected Topics in Computer Science: Combinatorial Testing (Exercises)	TU Graz
VU 188.916	Introduction to Security (Lectures and Exercises)	TU Wien
VU 188.959	Software Security (Lectures and Exercises)	TU Wien

Master Theses Topics available (TUG/TUW/WU)

Activity Report

HARNESSING STI FOR DISASTER RISK REDUCTION WORKSHOP

29 February-01 March 2024

Crimson Hotel, Alabang, Muntinlupa City, Philippines

A Joint initiative of the

Department of Science and Technology (DOST) of the Republic of the Philippines,
Department of State of the United States of America, and
the United Nations Conference on Trade and Development (UNCTAD)

Under the Philippines and United States of America membership in the
Commission on Science and Technology for Development

Decision-making tool based on combinatorial methodology for best-use of resources or option for action intended for government actors and disaster risk managers. The model offers a decision tree following a combination and sequence of events that would trigger a response action or not.

Policy Recommendation Excerpt

The **MATRIS Disaster Game** is a board game designed for raising awareness and understanding about different *fundamental parameters of disasters*

SBA Research

NIST National Institute of Standards and Technology

TU Graz Graz University of Technology

MATRIS

Thank you very much for your attention!

Questions / Comments ?

dsimos@sba-research.org

https://matris.sba-research.org/publications/

(All mentioned works can be found above)