Center for Embedded Systems

An NSF Industry/University Cooperative Research Center

On the Verification of Formal Methods for Digital Embedded Control Systems

Dr. Dimitri Kagaris

Dr. Spyros Tragoudas

Ehsan Ahmadi

SIUC

SIU Southern Illinois University CARBONDALE

NATIONAL SCIENCE FOUNDATION

ASU Ira A. Fulton Schools of Engineering ARIZONA STATE UNIVERSITY

Project Overview

Modelling and checking of specifications and requirements

Specifications and requirements are constantly being changed/refined as a project is developed.

Fundamental Problem: identify conflicts/incompatibilities

Approach:

- ❑ **Part 1: Translate from Semi-formal Description**
- ❑ **Part 2: Investigate the capabilities of the Z3 SMT solver**
- ❑ **Part 3: Develop an incremental approach and compare with static.**

Approach

Translate from Semi-Formal description

- Ideally, requirements should be

  Loosely described

  Strictly checked

To help bridge this gap, we propose to develop a tool that can translate from a "semi-formal" description of requirements to the extremely formal description that a theorem prover/solver needs.

Approach

Translate from Semi-Formal description

R1: [Average Main Frequency] shall be an average of the last 3 readings of [In XMS Frequency].

R2: [Main Power OverFrequency Condition] shall be set TRUE when [Average Main Frequency] is greater than 125Hz else it is set to FALSE.

R3: [Main Power OverFrequency Fault] shall be set TRUE when [Main Power OverFrequency Condition] is set to TRUE for 1 Second else it is set to FALSE.

R4: [Monitor Fault] shall be set TRUE when any of the following exist else it is set to FALSE:

[UnderFequency Fault] is set to TRUE.

[OverFequency Fault] is set to TRUE.

[OverCurrent Fault] is set to TRUE.

[UnderVoltage Fault] is set to TRUE.

[OverVoltage Fault] is set to TRUE.

R5: [Out XRM Close] shall be set TRUE when all of the following exist:

[Good Power] is set to TRUE.

[In XPOS] is set to FALSE.

[Monitor Fault] is set to FALSE.

Approach

Translate from Semi-Formal description

(1) Extract variables

(2) Check for variable naming inconsistencies

(3) Handle the type and values of variables (logical or arithmetic);

(4) Recognize "any" and "all" requirements;

(5) Handle the occurrence of composite operations like "average";

(6) Handle timing durations (such as "TRUE for 1 Sec")

(7) Check for misspellings/alternatives

Approach

Translate to Formal description

R1: = _AverageMainFrequency_ = (_InXMSFrequency_(t)_ + _InXMSFrequency_(t-1)_
+ _InXMSFrequency_(t-2)_) / 3

R2: = _MainPowerOverFrequencyCondition_ = (_AverageMainFrequency_ > 125 Hz)

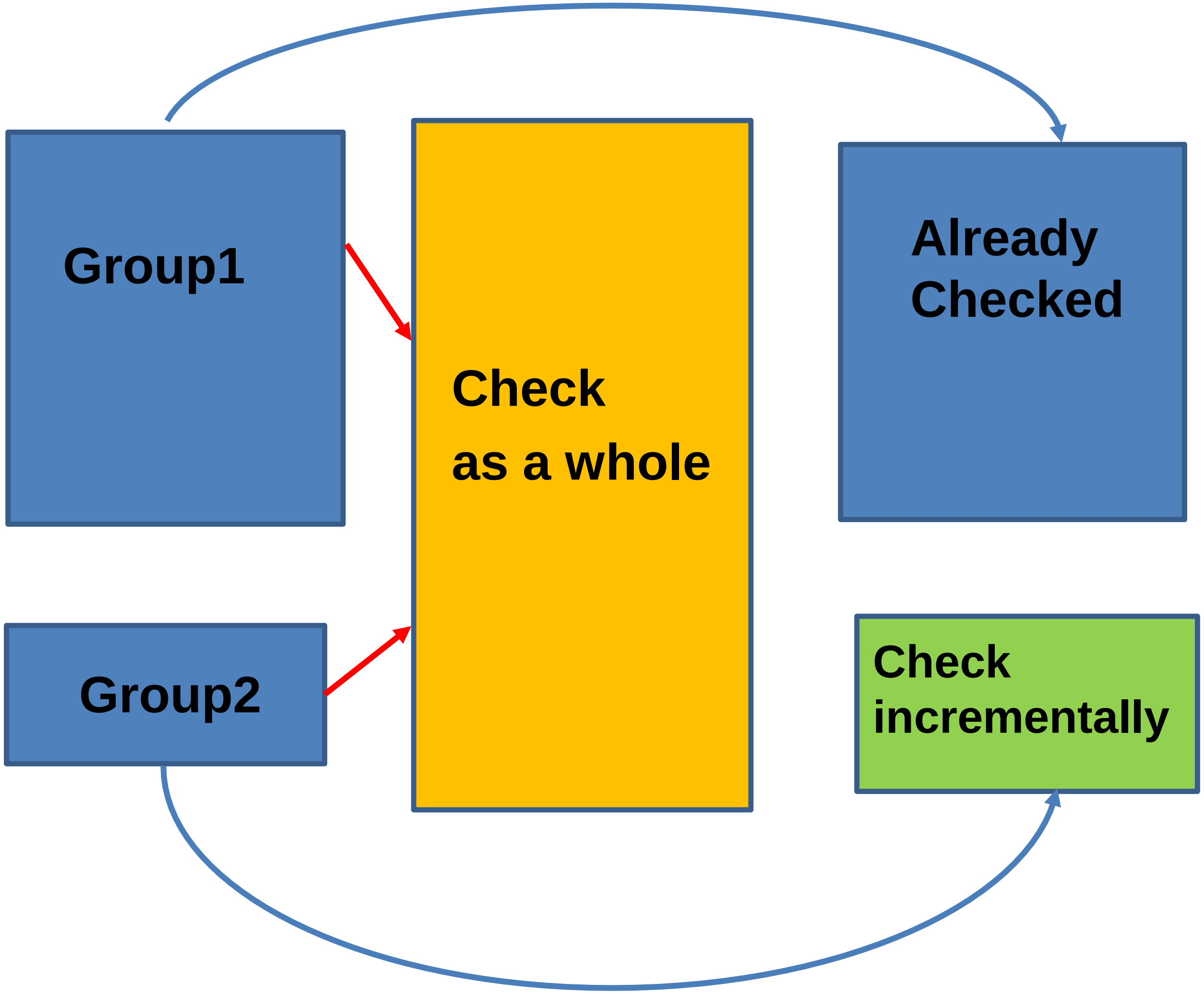
R3: = _MainPowerOverFrequencyFault_ = (_MainPowerOverFrequencyCondition_ AND
(_DURATION_MainPowerOverFrequencyCondition_ = 1 Sec))

R4: = _MonitorFault_ = _UnderFequencyFault_ OR _OverFequencyFault_ OR
OverCurrentFault OR _UnderVoltageFault_ OR _OverVoltageFault_

R5: = _OutXRMClose_ = _GoodPower_ AND ~_InXPOS_ AND ~_MonitorFault_

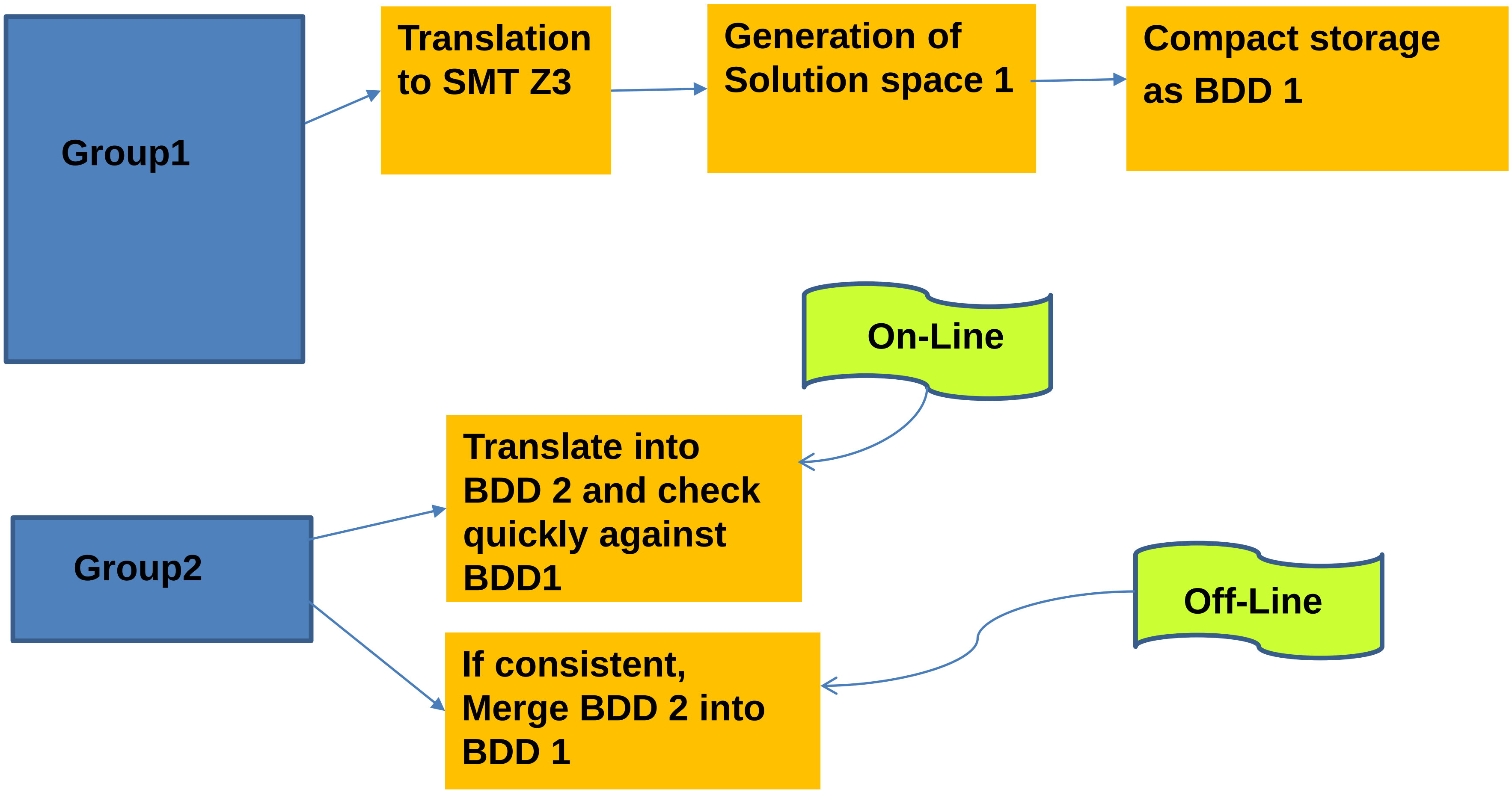
Approach

Incremental Verification/Consistency Check



Approach

Incremental Verification/Consistency Check



Group1

Translation to SMT Z3

Generation of Solution space 1

Compact storage as BDD 1

On-Line

Translate into BDD 2 and check quickly against BDD1

Group2

Off-Line

If consistent, Merge BDD 2 into BDD 1

Technical Detail

Example:

OLD REQUIREMENTS:

$$(0 < A < 8), (0 < B < 8),$$
$$(A + B)/2 < 3$$

NEW REQUIREMENT:

$$(2 < A < 8) \ \ \& \ \ (0 < B < 4)$$

Technical Detail

REQUIREMENT 2: set of satisfiable answers:

- (A=3, B=1) (A=3, B=2) (A=3, B=3)
- (A=4, B=1) (A=4, B=2) (A=4, B=3)
- (A=5, B=1) (A=5, B=2) (A=5, B=3)
- (A=6, B=1) (A=6, B=2) (A=6, B=3)
- (A=7, B=1) (A=7, B=2) (A=7, B=3)

Technical Detail

For first answer: (A = 3, B = 1)

A = 0011, B = 01 → related function = $\overline{a_3}\,\overline{a_2}\,a_1 a_0 \overline{b_1} b_0$

For another answer: (A = 6, B = 2)

A = 0110, B = 10 → related function = $\overline{a_3}\,a_2 a_1 \overline{a_0} b_1 \overline{b_0}$

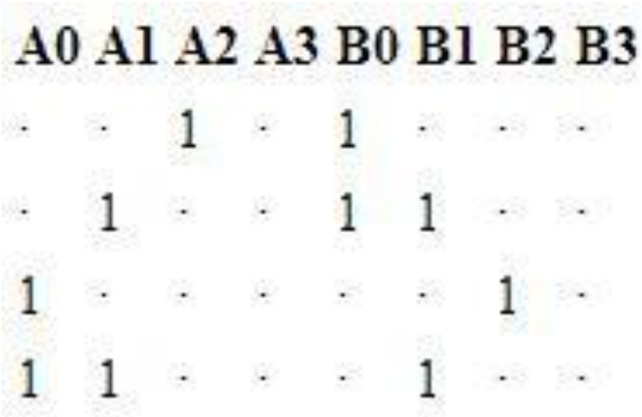
- We make binary decision diagram of the functions related to all the satisfiable for first answer.

Technical Detail

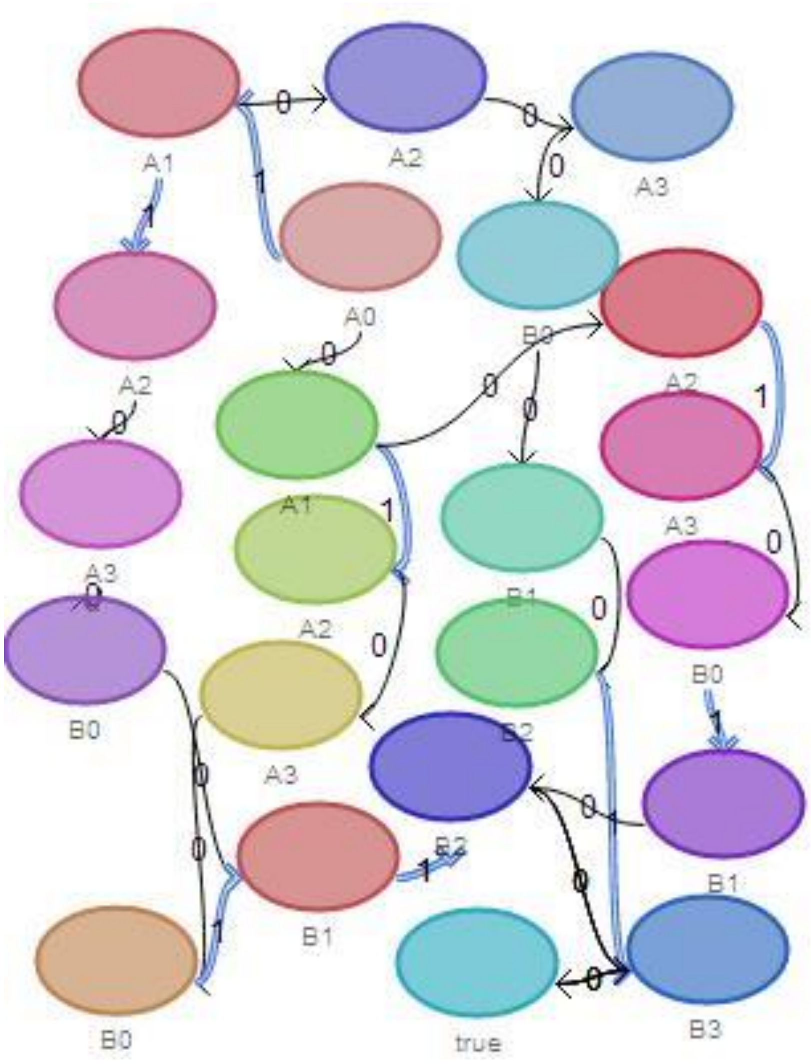
- **BDD for OLD Requirements**

(0 < A < 8), (0 < B < 8),

(A + B)/2 < 3



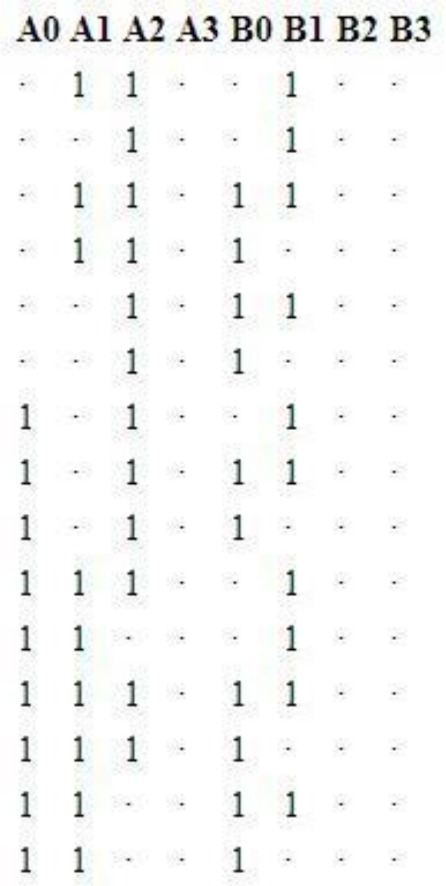
A0	A1	A2	A3	B0	B1	B2	B3
.	.	1	.	1	.	.	.
.	1	.	.	1	1	.	.
1	.	.	.	.	.	1	.
1	1	.	.	.	1	.	.



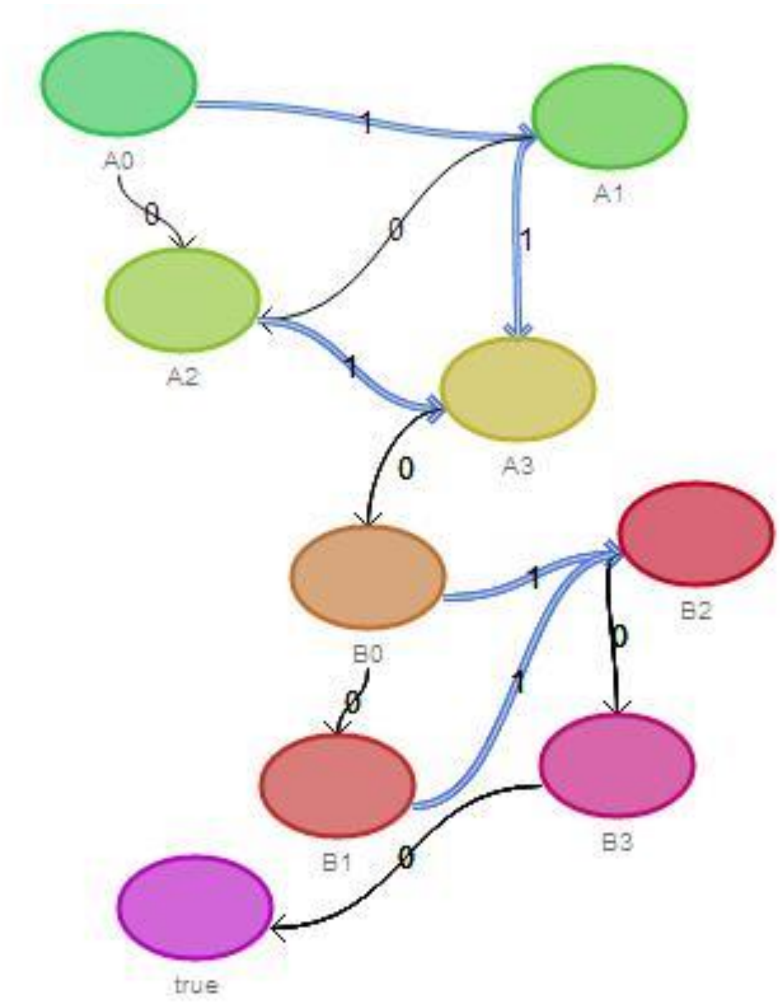
Technical Detail

- **BDD for NEW Requirement**

(2 < A < 8) & (0 < B < 4)



A0	A1	A2	A3	B0	B1	B2	B3
·	1	1	·	·	1	·	·
·	·	1	·	·	1	·	·
·	1	1	·	1	1	·	·
·	1	1	·	1	·	·	·
·	·	1	·	1	1	·	·
·	·	1	·	1	·	·	·
1	·	1	·	·	1	·	·
1	·	1	·	1	1	·	·
1	·	1	·	1	·	·	·
1	1	1	·	·	1	·	·
1	1	·	·	·	1	·	·
1	1	1	·	1	1	·	·
1	1	1	·	1	·	·	·
1	1	·	·	1	1	·	·
1	1	·	·	1	·	·	·



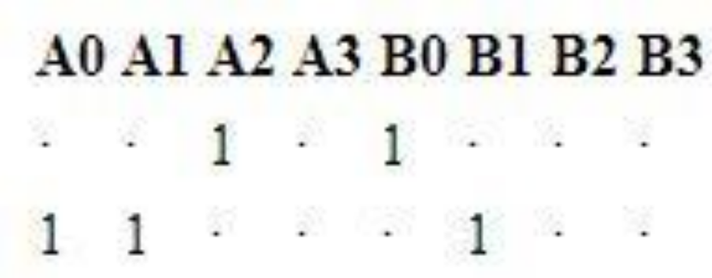
Technical Detail

- **UPDATED BDD**

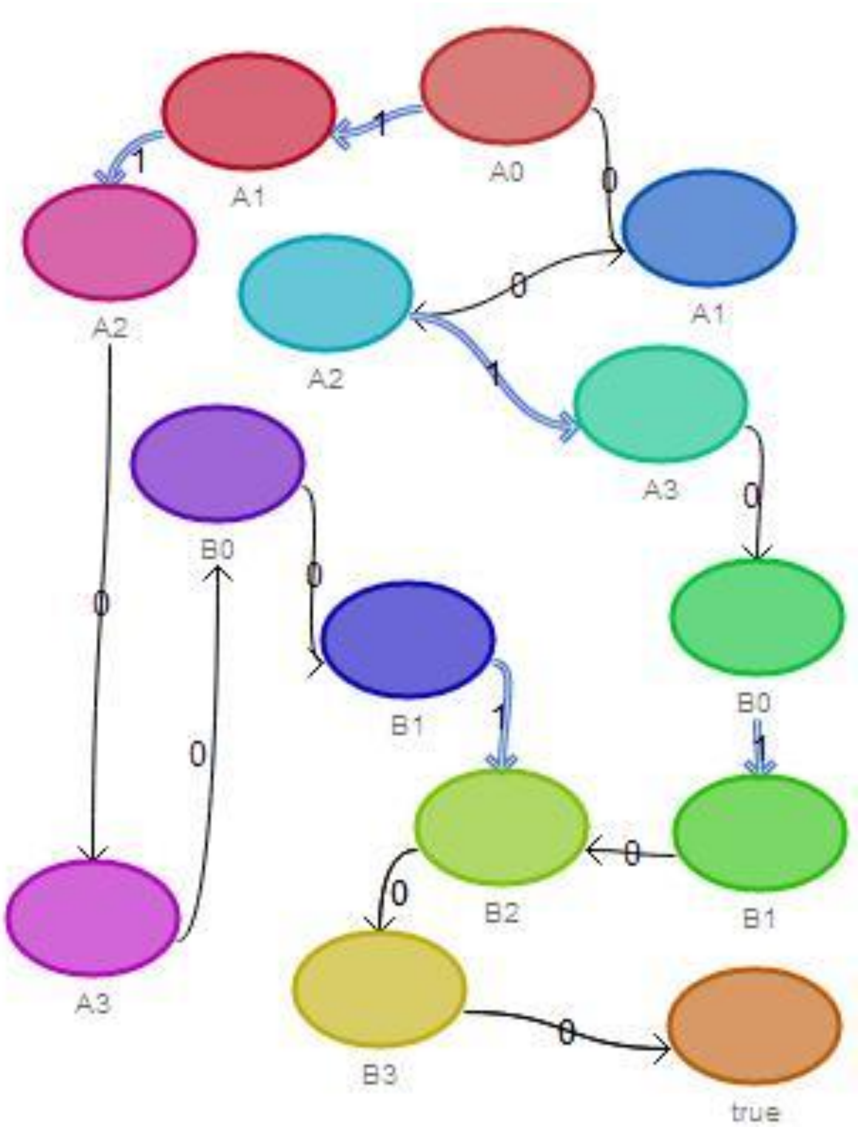
((A +B)/2 < 3)

&

(2 < A < 8) & (0 < B < 4)



A0	A1	A2	A3	B0	B1	B2	B3
.	.	1	.	1	.	.	.
1	1	.	.	.	1	.	.



Project Status

- **Satisfiability Modulo Theories (SMT) solver Z3**

Set of input requirements: Logic and Arithmetic

eg. R1: (A > 100) & (A < 10) & (B ≤ 10))

R2: (A+B)/2 < 80)

- **Script to translate in Z3**
- **Script to generate solution space**
- **Build BDD**
- **BDD intersection and merging**

Project Tasks/ Deliverables

	Description	Date	Status
1	Exploration of the capabilities of Z3 SMT solver. Script for translation into Z3 format	Q1	DONE
2	Script for generation of solution space BDD derivation and merging operations	Q2	In process
3	Extensive experimentation for scalability analysis	Q3, Q4	Not yet started

Deliverables:

- Methodology for automatic verification/validation/consistency check of a large amount of formal requirements.
- Application of the approach on industrial case studies.