Concordia University
Hardware Verification Group
Faculty of Engineering and Computer Science

UNIVERSITÉ Concordia UNIVERSITY

Using Machine Learning to Minimize User Intervention in Theorem Proving based Dynamic Fault Tree Analysis

Yassmeen Elderhalli, Osman Hasan and **Sofiène Tahar**

Concordia University

Montreal, QC, Canada

AITP 2019

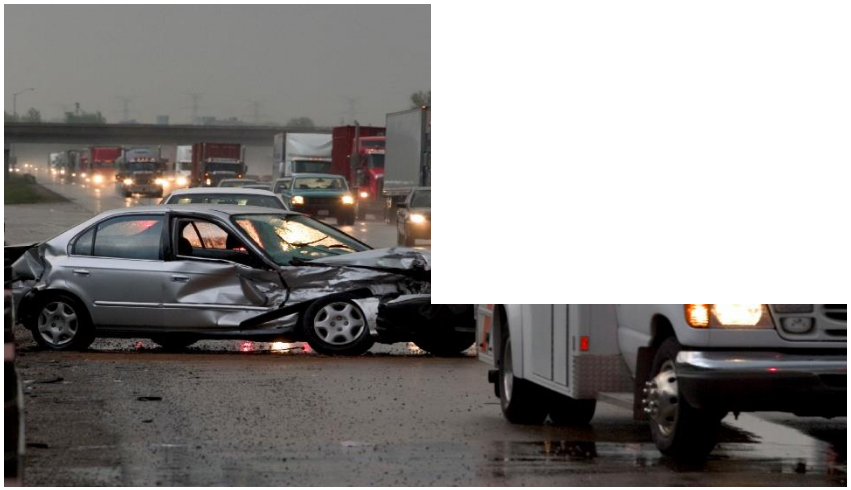
Obergurgl, Austria

April 9, 2019

Outline

- Introduction
- Dynamic Fault Trees
- Proposed Methodology
- Preliminary Results
- Conclusion and Future Work

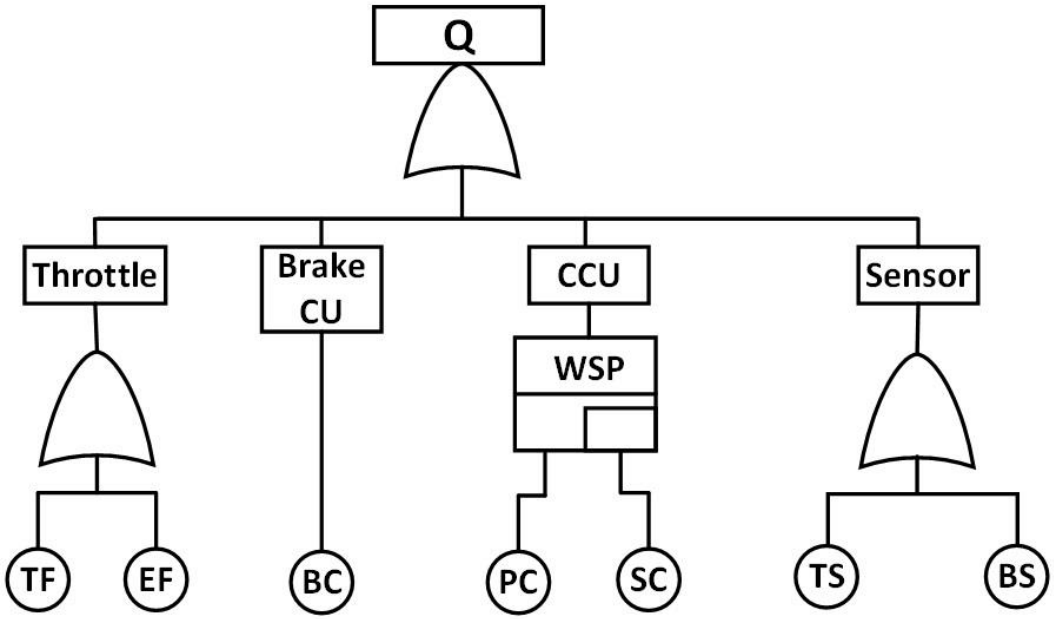
Failure Analysis



Analyze the effect of components faults on the system failure

Fault Trees

- Graphical representation of faults in the system
- Critical top event which will cause system failure
- The conditions are modeled using fault tree gates



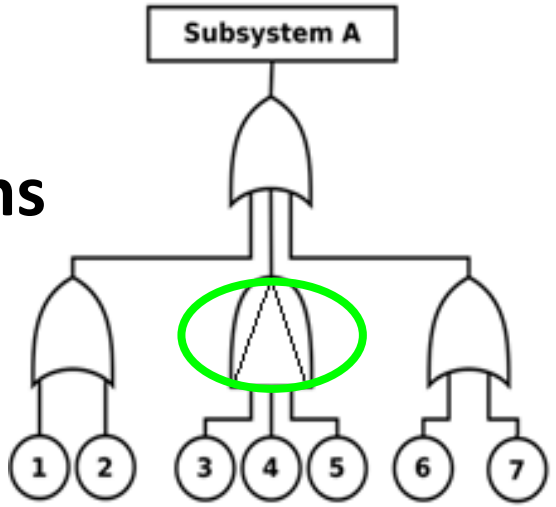
Fault Trees

Fault Trees

Static Fault Trees
SFTs

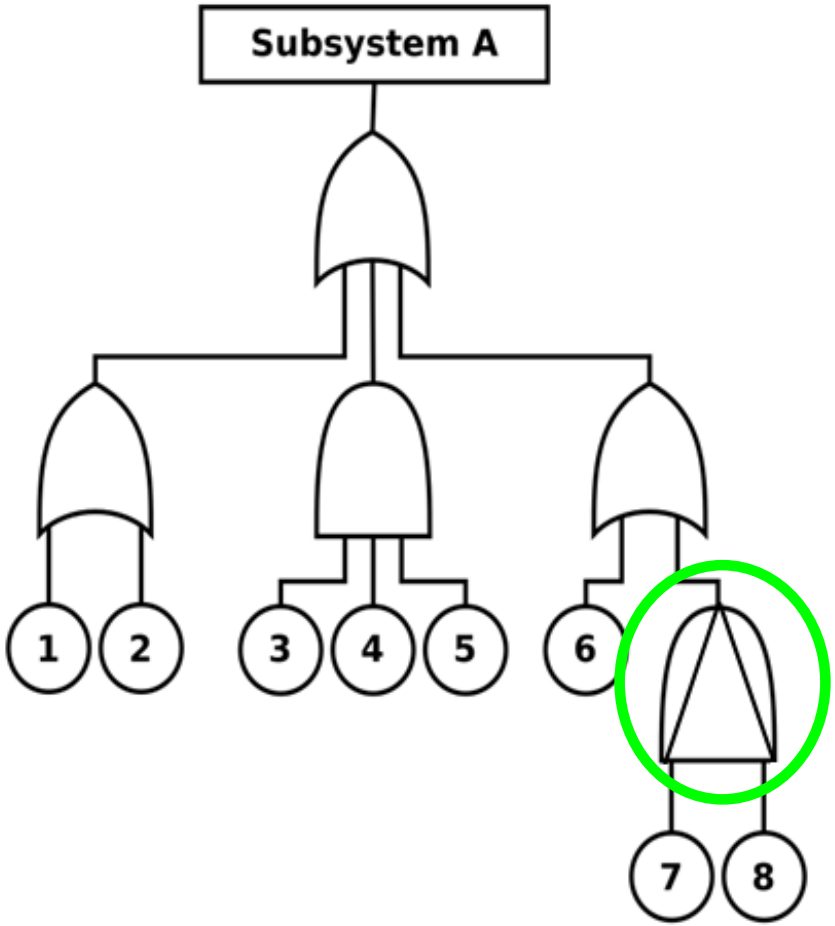
Dynamic Fault Trees
DFTs

Failure dependencies in real systems

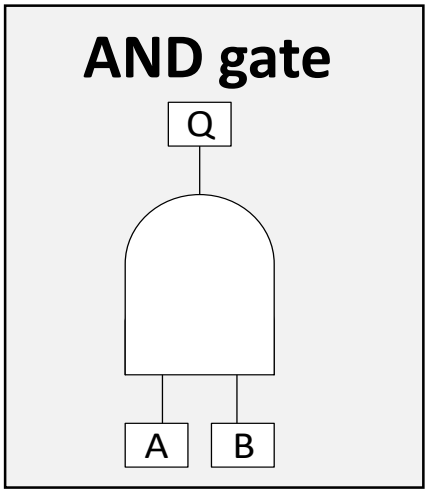


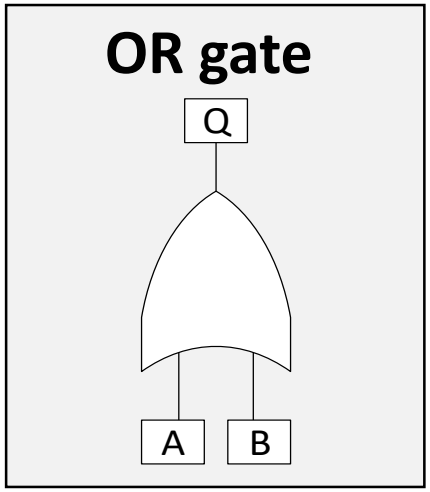
Dynamic Fault Trees

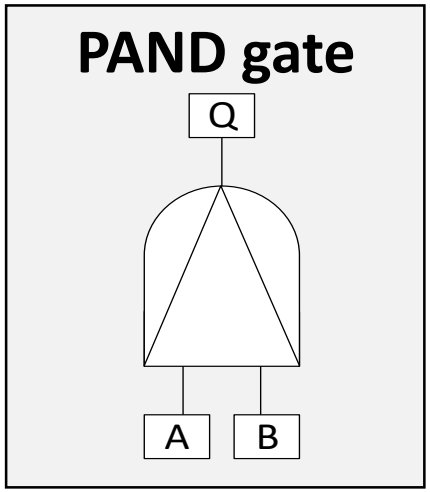
- Critical top event which will cause system failure
- The conditions are modeled using DFT and SFT gates
- DFTs capture the failure dependency using DFT gates (e.g. Priority-And gate)

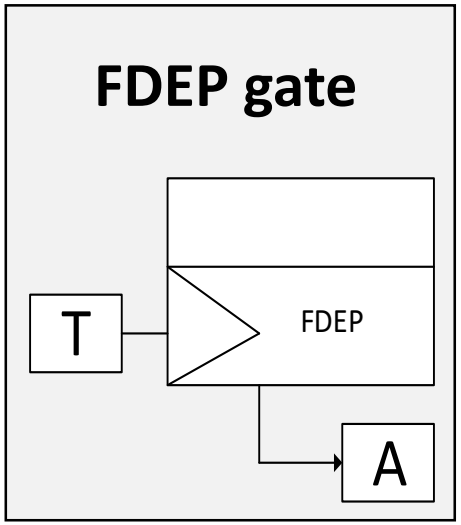


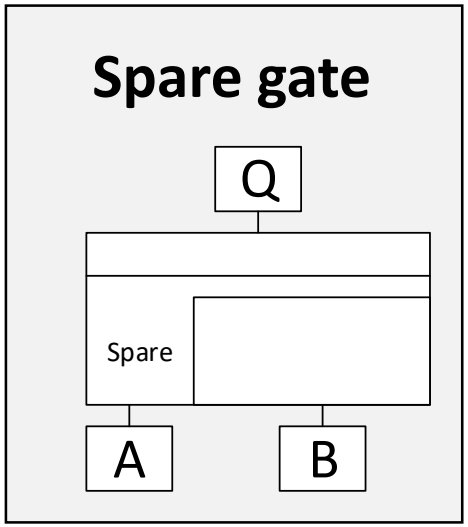
Dynamic Fault Trees Gates



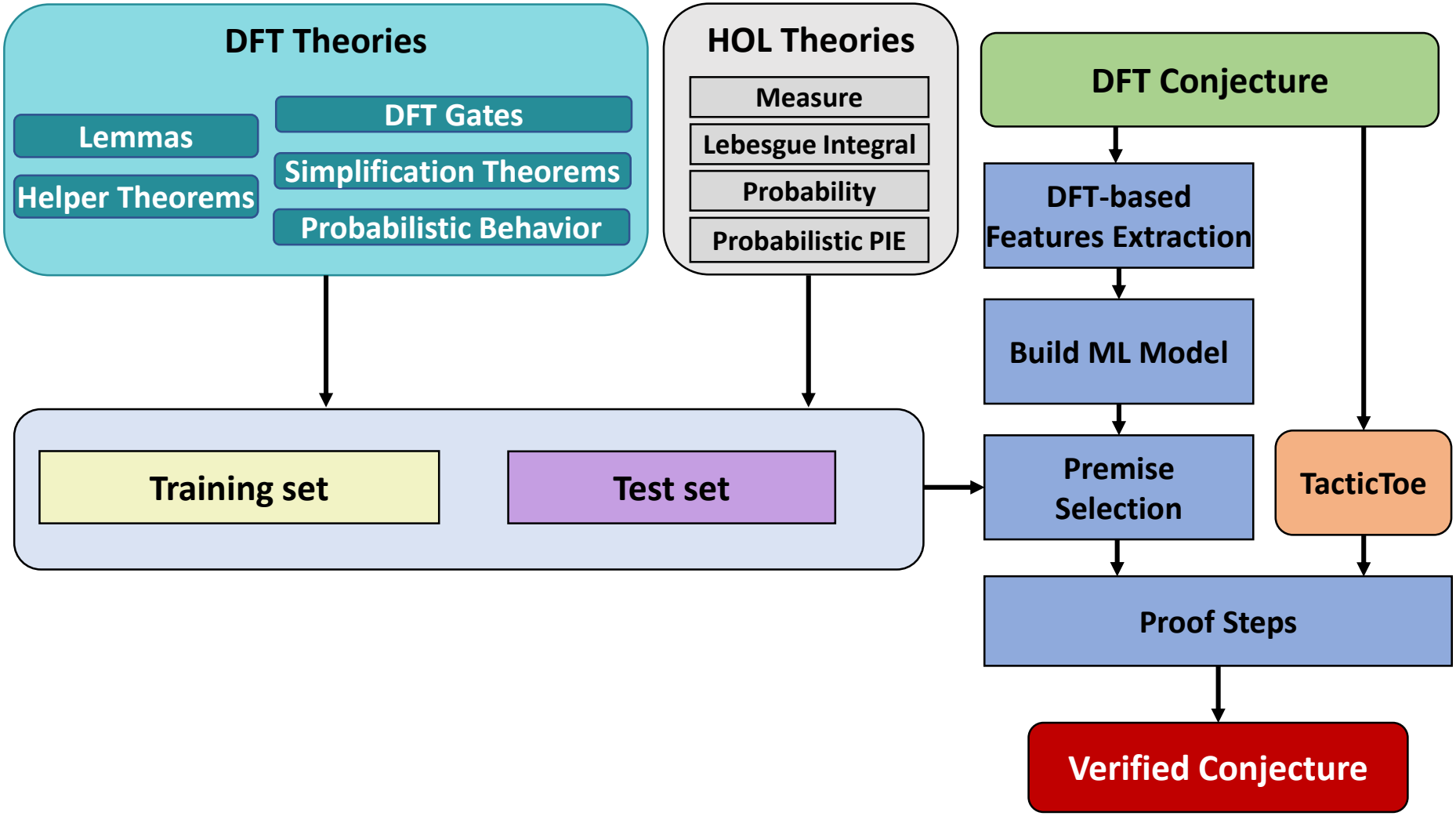








Ultimate Goal (HOL4)



Work done

- Formalization of DFT in HOL4 Theorem Prover

- Y. Elderhalli, O. Hasan, W. Ahmad and S. Tahar. "Formal Dynamic Fault Trees Analysis using an Integration of Theorem Proving and Model Checking". *In NASA Formal Methods (NFM-2018).*

- Y. Elderhalli, W. Ahmad, O. Hasan and S. Tahar "Probabilistic Analysis of Dynamic Fault Trees using HOL Theorem Proving", *In Journal of Applied Logic, 2019 [to appear]*

✓ Sound

✗ Interactive (not Automated)

The current libraries only support the analysis of Dynamic Fault Trees interactively

Dynamic Fault Trees

- Visualization of the cause of failure of the top event based on the basic events
- Dynamic gates in addition to the static gates
  - AND gate
  - OR gate
  - Priority AND gate
  - Functional Dependency gate
  - Spare gate
- Algebraic representation used in the DFT analysis

Dynamic Fault Trees Operators

- DFT temporal operators based on the time of failure:
  - **AND**

$$d(A\,.\,B) = \max(d(A), d(B))$$

  - **OR**

$$d(A + B) = \min(d(A), d(B))$$

  - **Simultaneous**

$$d(A\,\Delta\,B) = \begin{cases} d(A) & if\ d(A) = d(B) \\ +\infty & if\ d(A) \neq d(B) \end{cases}$$

Dynamic Fault Trees Operators

- **Before**

$$d(A \triangleleft B) = \begin{cases} d(A) & if\ d(A) < d(B) \\ +\infty & if\ d(A) \geq d(B) \end{cases}$$

- **Inclusive Before**

$$d(A \trianglelefteq B) = \begin{cases} d(A) & if\ d(A) \leq d(B) \\ +\infty & if\ d(A) > d(B) \end{cases}$$

Probabilistic Behavior of Gates

$$\mathcal{P}r\ \{A \cdot B\}\ (t) = F_A(t) \times F_B(t)$$

$$\mathcal{P}r\ \{A + B\}\ (t) = F_A(t)\ +\ F_B(t)\ -\ F_A(t) \times F_B(t)$$

$$\mathcal{P}r\ \{B.(A \lhd B)\}\ (t) = \int_0^t f_B(u) F_A(u)\ du$$

$$\mathcal{P}r\ \{Q_{spare}\}\ (t) = \int_0^t \left(\int_v^t f_{B_a}(u,v) du \right) f_A(v) dv\ +\ \int_0^t f_A(u) F_{B_d}(u) du$$

Quantitative Analysis

- The probability of the top event can be expressed using the probabilistic Principle of Inclusion Exclusion (PIE)

$$\begin{aligned}\mathcal{P}r\{TE\} &= \mathcal{P}r\{CSS_1 + CSS_2 + \ldots + CSS_m\} \\ &= \sum_{1\le i\le m} \mathcal{P}r\{CSS_i\} - \sum_{1\le i<j\le m} \mathcal{P}r\{CSS_i \cdot CSS_j\} \\ &\quad + \sum_{1\le i<j<k\le m} \mathcal{P}r\{CSS_i \cdot CSS_j \cdot CSS_k\} \\ &\quad + \ldots + (-1)^{m-1}\mathcal{P}r\{CSS_1 \cdot CSS_2 \cdot \ \ldots \ \cdot CSS_m\}\end{aligned}$$

Verification of Probabilistic Behavior of PAND

- The probabilistic failure behavior of the PAND

Theorem. Prob PAND

$$\vdash \forall\, X\ Y\ p\ f_Y\ t.\ \text{rv\_gt0\_ninfty}\ [X;Y] \wedge\ 0 \leq t\ \wedge \text{prob\_space}\ p \wedge$$

$$\text{indep\_var}\ p\ \text{lborel}\ X\ \text{lborel}\ Y \wedge \text{distributed}\ p\ \text{lborel}\ Y\ f_Y \wedge 0 \leq f_Y \wedge$$

$$\text{measurable\_CDF}\ \ (\text{real} \circ (\text{CDF}\ p\ \ (\text{real} \circ X)\ t)) \wedge$$

$$\text{cont\_CDF}\ (\text{real} \circ (\text{CDF}\ p\ (\text{real} \circ X)\ t)) \Longrightarrow$$

$$(\text{prob}\ p\ (\text{DFT\_event}\ p\ (Y\ .\ (X \lhd Y)\ t) = \int_0^t f_Y(y) \times F_X(y)\ dy)$$

Defines a density function for Y

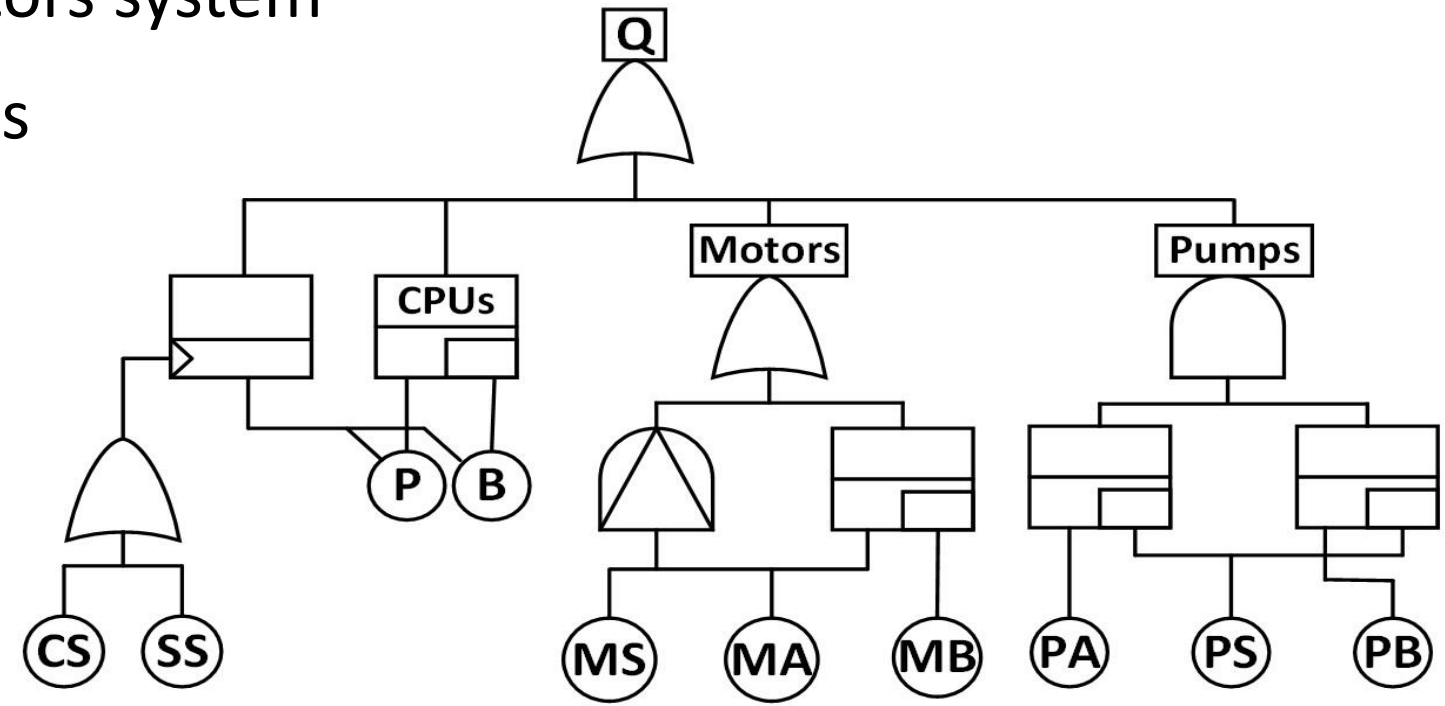
Algebraic Simplification Theorems

- Theorems needed to reduce the expression of the top event of the DFT (structure function)
- Many simplification theorems exist[1]:
  - Commutativity $$A \,\Delta\, B = B \,\Delta\, A$$
  - Associativity $$A + (B + C) = (A + B) + C$$
  - Distributivity $$A.(B + C) = A.B + A.C$$

1- [G. Merle , "Algebraic modelling of Dynamic Fault Trees, Contribution to Qualitative and Quantitative Analysis", PhD thesis, ENS, France, 2010].

The Cardiac Assist System

- It consists of:
  - Pumps system
  - Motors system
  - CPUs



The Cardiac Assist System

- A reduced structure function is obtained to conduct both qualitative and quantitative analyses.

Theorem. *Reduced cardiac assist system*

$$\vdash \forall CS\ SS\ MA\ MS\ MB\ P\ B\ PA\ PB\ PS.$$

$$(\forall\ s.\ \text{ALL\_DISTINCT}\ [MA\ s;\ MS\ s;\ PA\ s;\ PB\ s;\ PS\ s]) \Longrightarrow$$

$$((\text{shared\_spare}\ PA\ PB\ PS\ PS)\ .\ (\text{shared\_spare}\ PB\ PA\ PS\ PS) +$$

$$(\text{PAND}\ MS\ MA) + (\text{HSP}\ MA\ MB\ ) +$$

$$(\text{HSP}\ (\text{FDEP}\ ((CS + SS)\ P)\ (\text{FDEP}\ ((CS + SS)\ B)) =$$

$$CS + SS + (MA\ .\ (MS \lhd MA)) + MA\ .\ MB + P.B + PA\ .\ PB.\ PS)$$

The Cardiac Assist System

Theorem. *Reduced cardiac assist system*

$$\vdash \forall CS\ SS\ MA\ MS\ MB\ P\ B\ PA\ PB\ PS.$$

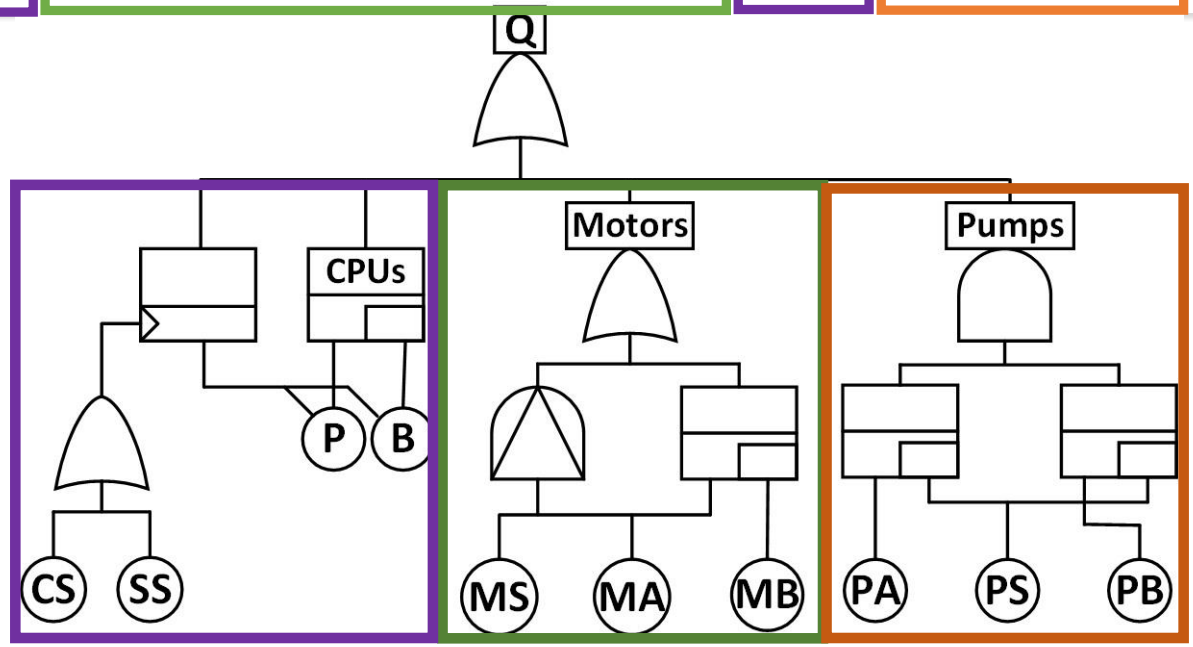
$$(\forall\, s.\ \mathrm{ALL\_DISTINCT}\ [MA\ s;\ MS\ s;\ PA\ s;\ PB\ s;\ PS\ s]) \Longrightarrow$$

$$((\mathrm{shared\_spare}\ PA\ PB\ PS\ PS)\ .\ (\mathrm{shared\_spare}\ PB\ PA\ PS\ PS)\ \text{--}$$

$$(\mathrm{PAND}\ MS\ MA) + (\mathrm{HSP}\ MA\ MB\ ) +$$

$$(\mathrm{HSP}\ (\mathrm{FDEP}\ ((CS + SS)\ P)\ (\mathrm{FDEP}\ ((CS + SS)\ B)) =$$

$$CS + SS + (MA\ .\ (MS \lhd MA)) + MA\ .\ MB\ \text{--}\ P.B + PA\ .\ PB.\ PS)$$



The Cardiac Assist System

Lemma. *Cardiac assist system union_list*

⊢ ∀PA PB PS MS MA MB CS SS P B p t.

DFT_event p

(CS + SS + (MA . (MS ◁ MA)) + MA . MB + P.B + PA . PB. PS) t =

union_list

[DFT_event p CS t; DFT_event p SS t;

DFT_event p (MA . (MS ◁ MA)) t;

DFT_event p (MA. MB) t;

DFT_event p (P. B) t; DFT_event p (PA. PB. PS) t]

The Cardiac Assist System

Theorem. Prob Cardiac assist system

$\vdash \forall CS\ SS\ MA\ MS\ MB\ P\ B\ PA\ PB\ PS\ \ p\ t\ f_{MA}.$

$0 \leq t \wedge\ prob\_space\ p \wedge$

$ALL\_DISTINCT\_RV\ [CS; SS; MA; MS; MB; P; B; PA; PB; PS]\ p\ t \wedge$

$indep\_vars\_sets\ [CS; SS; MA; MS; MB; P; B; PA; PB; PS]\ p\ t \wedge$

$distributed\ p\ lborel\ MA\ f_{MA} \wedge\ 0 \leq f_{MA} \wedge$

$cont\_CDF\ F_{MS} \wedge measurable\_CDF\ F_{MS} \Longrightarrow$

F_{MS} is continuous and measurable

The Cardiac Assist System

(prob p

(DFT_event p

(CS + SS + (MA . (MS ◅ MA)) + MA . MB + P.B + PA . PB. PS) t) =

$F_{CS}(t) + F_{SS}(t) + \int_0^t f_{MA}(y) \times F_{MS}(y)dy + F_{MA}(t) \times F_{MB}(t) + F_P(t) \times F_B(t) +$

$F_{PA}(t) \times F_{PB}(t) \times F_{PS}(t) - \cdots + \cdots -$

$F_{CS}(t) \times F_{SS}(t) \times (\int_0^t f_{MA}(y) \times F_{MS}(y)dy) \times F_{MA}(t) \times F_{MB}(t) \times F_P(t) \times F_B(t) \times$

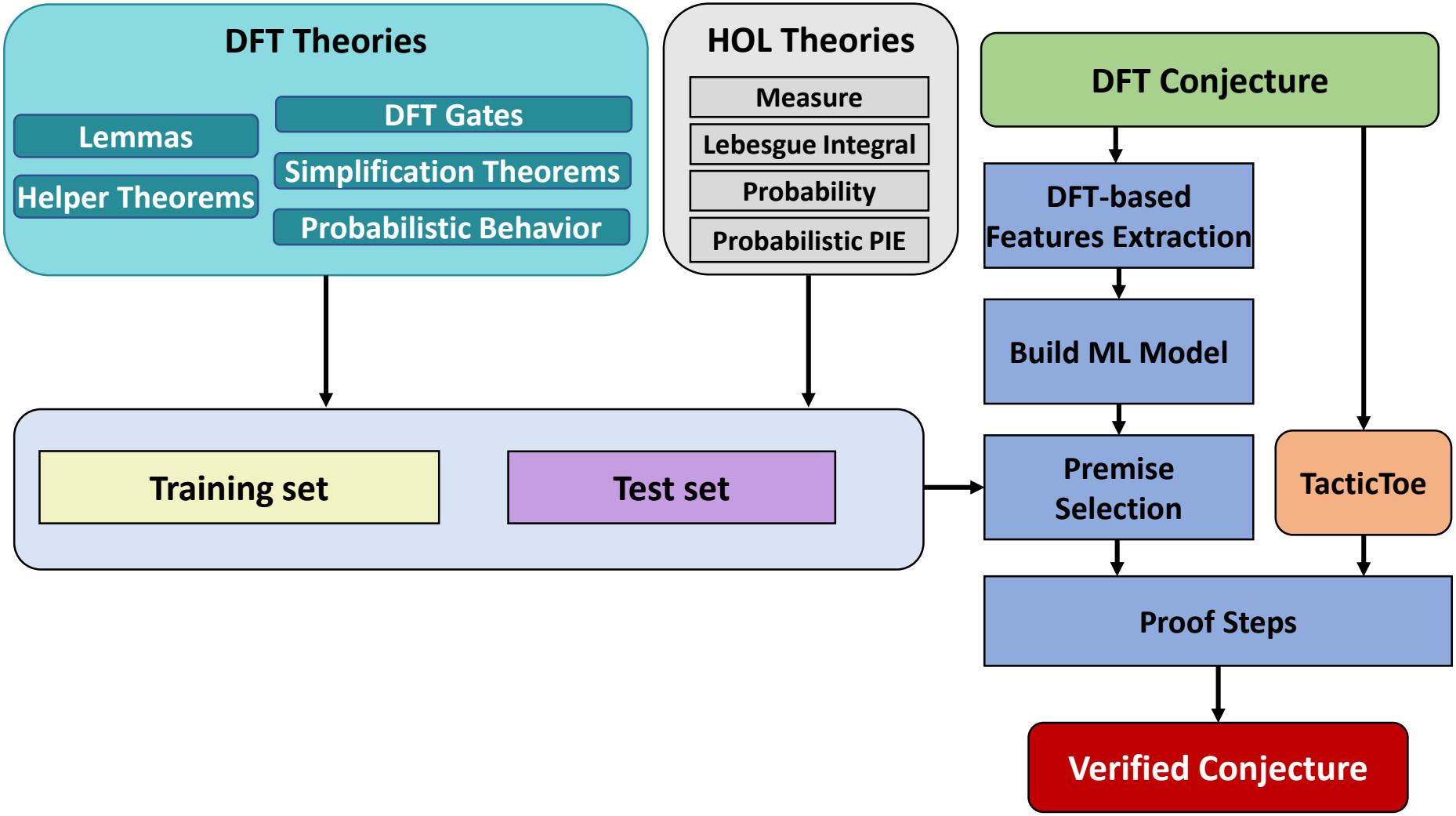
$F_{PA}(t) \times F_{PB}(t) \times F_{PS}(t)$

The result of applying PIE is 63 (2^6-1) elements

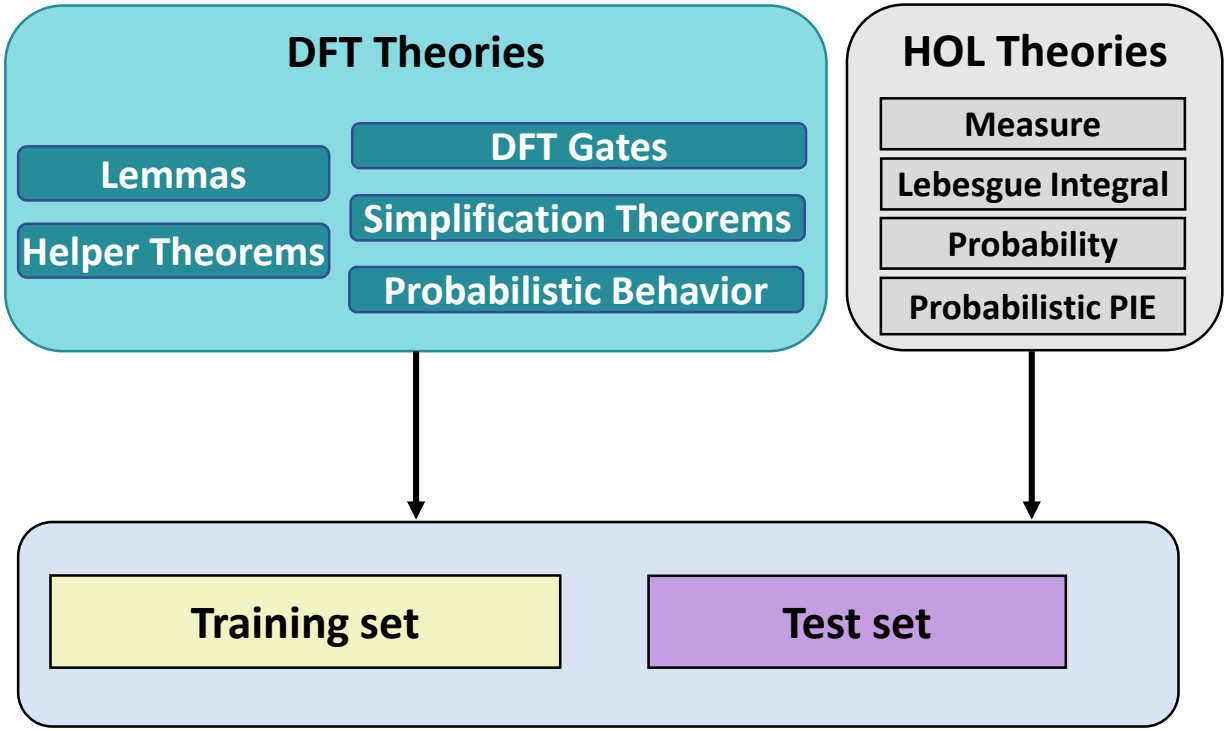
Formalization Summary

- DFT gates and simplification theorems
- Probabilistic behavior of DFT gates
- Utilizing the probabilistic PIE in the quantitative analysis leads to having many subgoals
- **Intermediate lemmas are verified that follow the same pattern**

Proposed Methodology



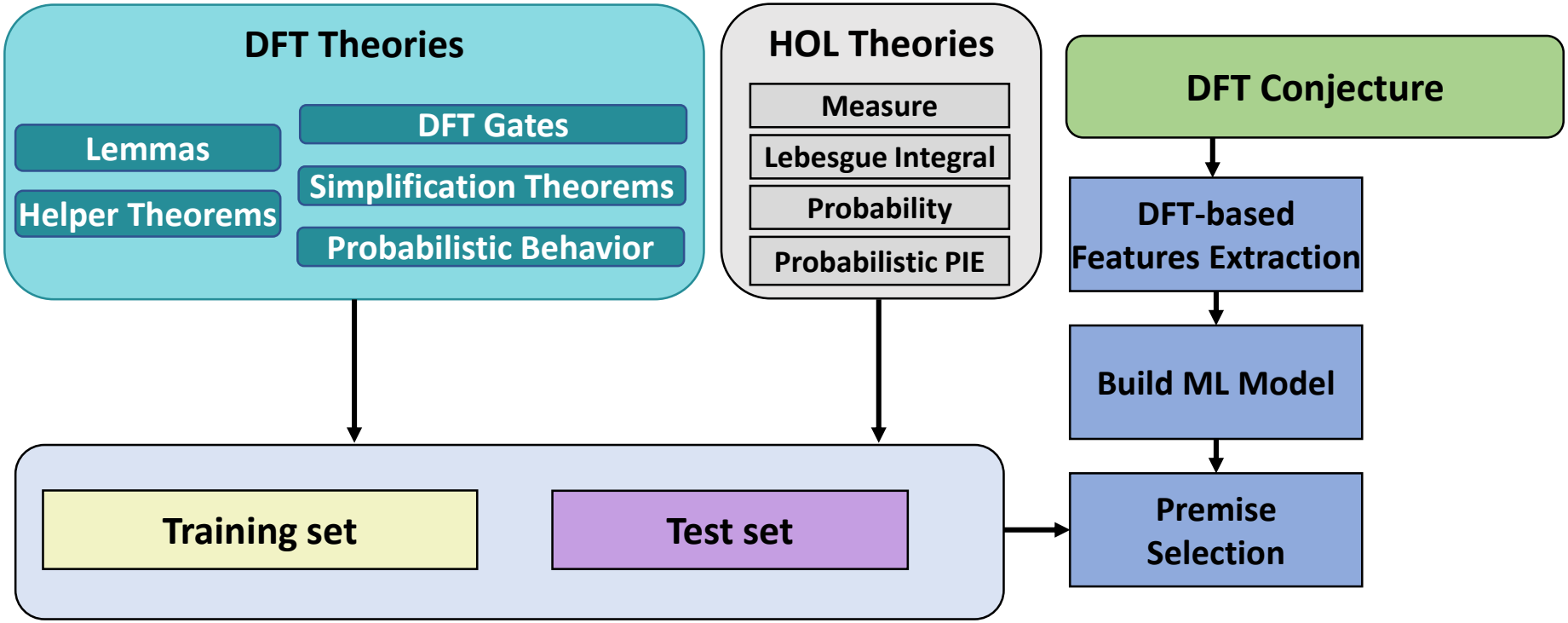
Proposed Methodology



- Divide the existing theories into training and test sets, similar to Holstep[1] , based on certain features, such as input statements

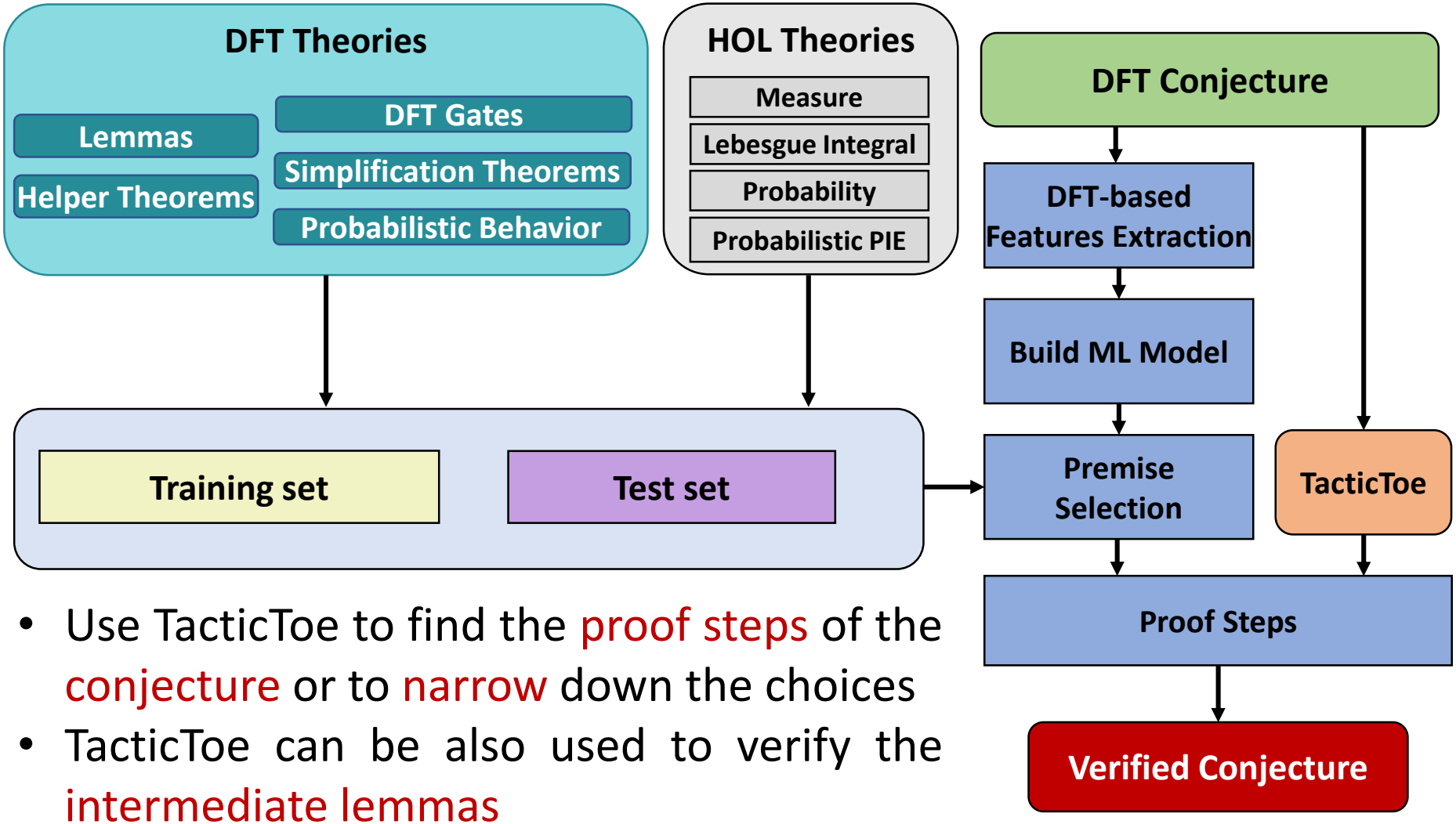
1. C. Kaliszyk, F. Chollet, and C. Szegedy. "*Holstep: A machine learning dataset for higher-order logic theorem proving*", 2017.

Proposed Methodology



- For DFT conjectures, features are extracted to build ML models such as neural networks
- The ML models will be used to find the suitable premises

Proposed Methodology



- Use TacticToe to find the proof steps of the conjecture or to narrow down the choices
- TacticToe can be also used to verify the intermediate lemmas

TacticToe

- TacticToe[1] is used to record part of DFT theories
- TacticToe is tested with a small subset of intermediate lemmas
- Proof steps were determined for small lemmas not complex theorems

$$\forall A1\ A2\ A3\ A4\ A5\ A6\ p.\ prob\_space\ p \wedge A1 \in events\ p \wedge A2 \in events\ p \wedge A3 \in events\ p \wedge A4 \in events\ p \wedge A5 \in events\ p \wedge A6 \in events\ p \Longrightarrow A1 \cap A2 \cap A3 \cap A4 \cap A5 \cap A6 \in events\ p$$

```
RW_TAC bossLib.std_ss [] THEN (MATCH_MP_TAC o REWRITE_RULE
[subsets_def] o Q.SPEC `(p_space p, events p)`) ALGEBRA_INTER
THEN RW_TAC bossLib.std_ss [] THENL [RW_TAC bossLib.std_ss
[EVENTS_ALGEBRA], RW_TAC bossLib.std_ss [EVENTS_INTER]]);
```

1- In collaboration with Cezary Kaliszyk

Generic Lemmas

- Verifying generic lemmas that can facilitate the learning process[1]
- The extreal addition associativity is used with the PIE and other lemmas to reach the final form of the probability of CAS

$$\forall L.\ (\neg \text{MEM PosInf } L) \bigvee (\neg \text{MEM NegInf } L) \Longrightarrow$$
$$(\text{FOLDR } (\lambda a\ b.\ a + b)\ 0\ L = \text{FOLDL } (\lambda a\ b.\ a+b)\ 0\ L)$$

$$\forall A1\ A2\ A3\ A4\ A5\ A6\ A7\ A8\ A9\ A10.$$
$$A1 \neq \text{PosInf} \bigwedge A2 \neq \text{PosInf} \bigwedge A3 \neq \text{PosInf} \bigwedge A4 \neq \text{PosInf} \bigwedge$$
$$A5 \neq \text{PosInf} \bigwedge A6 \neq \text{PosInf} \bigwedge A7 \neq \text{PosInf} \bigwedge A8 \neq \text{PosInf} \bigwedge$$
$$A9 \neq \text{PosInf} \bigwedge A10 \neq \text{PosInf} \Longrightarrow$$
$$(A1 + (A2 + (A3 + (A4 + (A5 + (A6 + (A7 + (A8 + (A9 + A10)))))))) =$$
$$A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10)$$

1- In collaboration with Cezary Kaliszyk

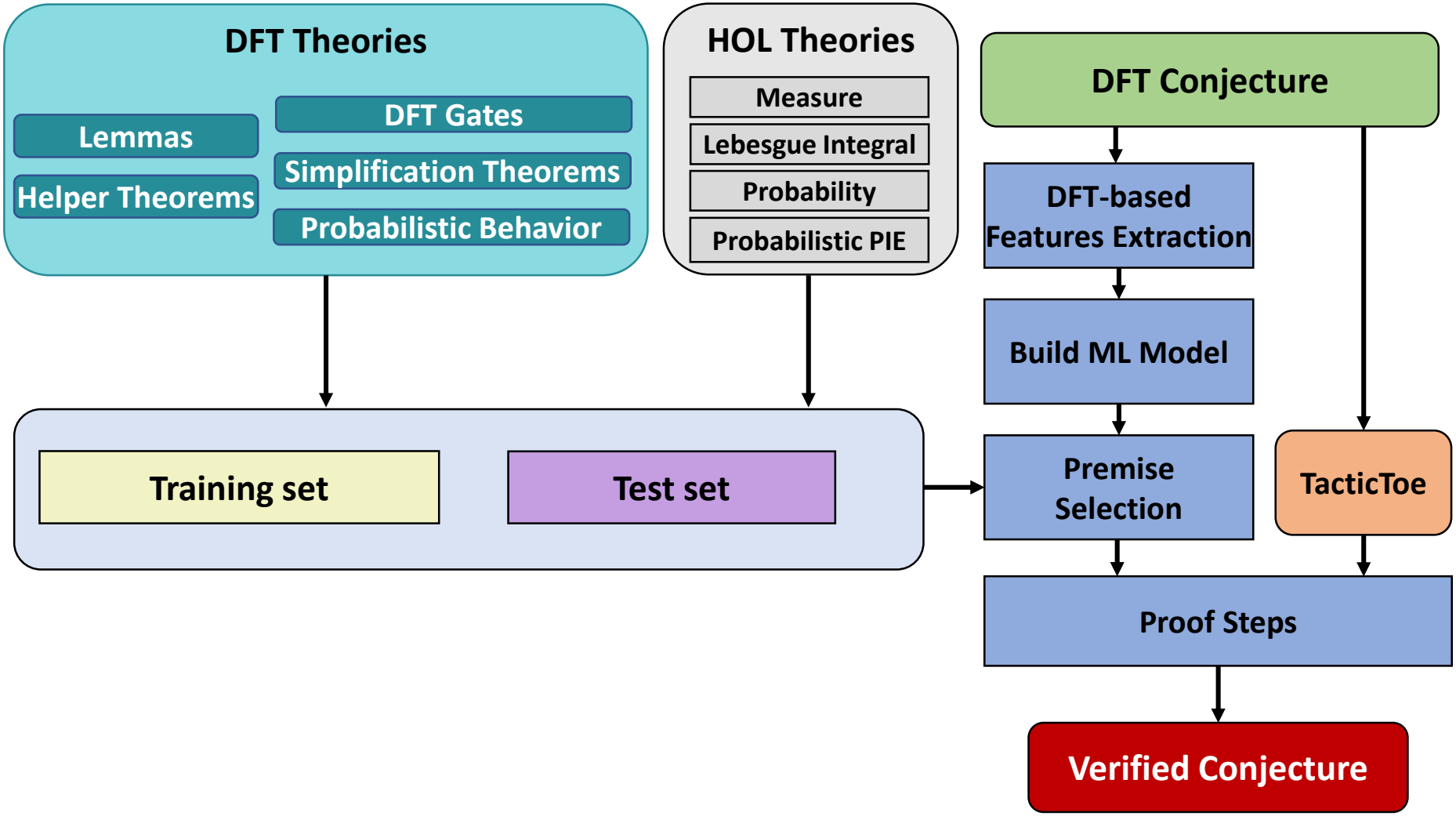
Conclusion

- Verified DFT algebraic analysis using interactive theorem proving
- A methodology to reduce user intervention in the analysis using machine learning techniques
- TacticToe is used with a small subset of DFT theorems

Future Work

- Divide DFT theories into training and testing sets
- Create ML models and use them with the testing set
- Use TacticToe to extract the proof steps that are useful in the proof steps
- Combine both ML models and TacticToe to generate the proof steps required to verify a given conjecture

Future Work


DFT Theories
Lemmas
DFT Gates
Helper Theorems
Simplification Theorems
Probabilistic Behavior
HOL Theories
Measure
Lebesgue Integral
Probability
Probabilistic PIE
DFT Conjecture
DFT-based Features Extraction
Build ML Model
Training set
Test set
Premise Selection
TacticToe
Proof Steps
Verified Conjecture

Concordia University

Hardware Verification Group

Faculty of Engineering and Computer Science

www.hvg.ece.concordia.ca