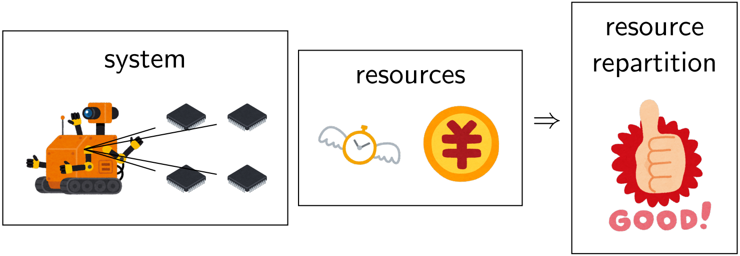
Architecture-Guided Test Resource Allocation Via Logic

Clovis Eberhart[1,2] Akihisa Yamada[3] Stefan Klikovits[1]
Shin-ya Katsumata[1] Tsutomu Kobayashi[4,1] Ichiro Hasuo[1]
Fuyuki Ishikawa[1]

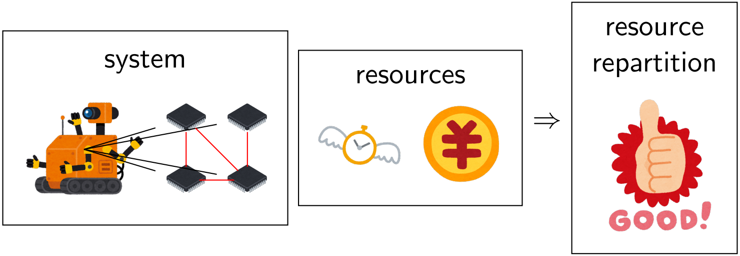
[1]National Institute of Informatics, Japan
[2]Japanese-French Laboratory for Informatics, Japan
[3]National Institute of Advanced Industrial Science and Technology, Japan
[4]Japan Science and Technology Agency, Japan
eberhart@nii.ac.jp

TAP 2021, June 21–22

Test Resource Allocation Problem



Test Resource Allocation Problem



Architecture

- more or less critical modules
- independent of reliability

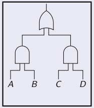
Architecture:

- should influence TRA
- not taken into account in most approaches

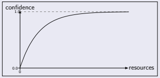
Our approach to the TRAP

Our approach

$$\dfrac{\dfrac{\overline{\Gamma \vdash A} \quad \overline{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \qquad \dfrac{\overline{\Gamma \vdash C} \quad \overline{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$



- system: represented by QCL proof (e.g. via a fault tree)
- reliability of each module: given by confidence functions
- (limited) resources



Our solution

Solve an optimisation problem

Validation of approach: experimental results (`Astrahl`)

Applications

Applications

- complex systems (complex architecture)
- heterogeneous systems (different types of components)
- continuous development
- product line development

Disclaimer

- QCL: general framework for confidence
- this work: preliminary results
- needs more experimental results

Quantitative Confidence Logic Formulas

Quantitative Confidence Logic (QCL)

- confidence (not truth)
- positive and negative

Formulas: $\varphi ::= A \mid \top \mid \bot \mid \varphi \Rightarrow \varphi$ $(\neg\varphi,\ \varphi \wedge \varphi,\ \varphi \vee \varphi)$

Formula with confidence

$\varphi\colon (t, f)$ with $(t, f) \in \left\{(t, f) \in [0, 1]^2 \,\middle|\, t + f \leq 1\right\}$

- t: positive confidence
- f: negative confidence

Examples

- $\varphi\colon (0, 0)$: totally unknown
- $\varphi\colon (1, 0)$: true with total confidence
- $\varphi\colon (1/2, 1/2)$: total confidence in truth with probability $1/2$

QCL Proof Rules

$$\frac{}{\Gamma, \varphi\colon (t,f) \vdash \varphi\colon (t,f)}\ (ax) \qquad \frac{}{\Gamma \vdash \varphi\colon (0,0)}\ (unk)$$

$$\frac{}{\Gamma \vdash \top\colon (1,0)}\ (\top_I) \qquad \frac{}{\Gamma \vdash \bot\colon (0,1)}\ (\bot_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t,f) \qquad \Gamma \vdash \psi\colon (t',f')}{\Gamma \vdash \varphi \Rightarrow \psi\colon (f + t' - ft', tf')}\ (\Rightarrow_I)$$

$$\frac{\Gamma \vdash \varphi \Rightarrow \psi\colon (t,f) \qquad \Gamma \vdash \varphi\colon (t',f')}{\Gamma \vdash \psi\colon \left(1 - \frac{1-t}{t'}, \frac{f}{1-f'}\right)}\ (\Rightarrow_{E,l}) \text{ if } t' \neq 0 \text{ and } f' \neq 1$$

$$\frac{\Gamma \vdash \varphi \Rightarrow \psi\colon (t,f) \qquad \Gamma \vdash \psi\colon (t',f')}{\Gamma \vdash \varphi\colon \left(\frac{f}{1-t'}, 1 - \frac{1-t}{f'}\right)}\ (\Rightarrow_{E,r}) \text{ if } t' \neq 1 \text{ and } f' \neq 0$$

Derivable Rules

$$\frac{\Gamma \vdash \varphi\colon (t, f)}{\Gamma \vdash \neg\varphi\colon (f, t)}\ (\neg_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \wedge \psi\colon (tt', f + f' - ff')}\ (\wedge_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \vee \psi\colon (t + t' - tt', ff')}\ (\vee_I)$$

Derivable Rules

$$\frac{\Gamma \vdash \varphi\colon (t, f)}{\Gamma \vdash \neg\varphi\colon (f, t)}\ (\neg_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \wedge \psi\colon (tt', f + f' - ff')}\ (\wedge_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \vee \psi\colon (t + t' - tt', ff')}\ (\vee_I)$$

Derivable Rules

$$\frac{\Gamma \vdash \varphi\colon (t, f)}{\Gamma \vdash \neg\varphi\colon (f, t)}\ (\neg_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \wedge \psi\colon (tt', f + f' - ff')}\ (\wedge_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \vee \psi\colon (t + t' - tt', ff')}\ (\vee_I)$$

Derivable Rules

$$\frac{\Gamma \vdash \varphi\colon (t, f)}{\Gamma \vdash \neg\varphi\colon (f, t)}\ (\neg_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \wedge \psi\colon (tt', f + f' - ff')}\ (\wedge_I)$$

$$\frac{\Gamma \vdash \varphi\colon (t, f) \qquad \Gamma \vdash \psi\colon (t', f')}{\Gamma \vdash \varphi \vee \psi\colon (t + t' - tt', ff')}\ (\vee_I)$$

$$\frac{\Gamma \vdash \varphi\colon \color{red}{(t, f)} \qquad \dfrac{}{\Gamma \vdash \psi\colon (0, 0)}\ (unk)}{\Gamma \vdash \varphi \vee \psi\colon (t + 0 - t \cdot 0, f \cdot 0) = \color{red}{(t, 0)}}\ (\vee_I)$$

QCL Proof Trees

Example

Simple CPS:

$$\dfrac{\dfrac{}{\Gamma \vdash \text{software}}\,(ax) \qquad \dfrac{}{\Gamma \vdash \text{hardware}}\,(ax)}{\Gamma \vdash \text{software} \wedge \text{hardware}}\,(\wedge_I).$$

QCL Proof Trees

Example

Simple CPS:

$$\dfrac{\dfrac{}{\Gamma \vdash \text{software}\colon (0.5, 0.2)}\,(ax) \qquad \dfrac{}{\Gamma \vdash \text{hardware}\colon (0.3, 0.01)}\,(ax)}{\Gamma \vdash \text{software} \wedge \text{hardware}\colon (0.15, 0.208)}\,(\wedge_I).$$

QCL Proof Trees

Example

Simple CPS:

$$\dfrac{\dfrac{}{\Gamma \vdash \text{software}\colon (0.5, 0.2)}\,(ax) \quad \dfrac{}{\Gamma \vdash \text{hardware}\colon (0.3, 0.01)}\,(ax)}{\Gamma \vdash \text{software} \wedge \text{hardware}\colon (0.15, 0.208)}\,(\wedge_I).$$

QCL:

- not about truth
- flow of confidence from hypotheses to conclusion

QCL, Dempster-Shafer Theory, and Fuzzy Logic

Dempster-Shafer Theory

- theory of belief
- major difference to Bayesian approaches: $t + f \leq 1$ (rather than $t + f = 1$)

Fuzzy logical features:

- product T-norm (interpretation of $\wedge$)
- probabilistic sum T-conorm (interpretation of $\vee$)
- involution (interpretation of $\neg$)

QCL vs Fuzzy Logic

- φ: t with $t \in [0, 1]$, equivalent: φ: (t, f) with $f = 1 - t$
- φ: (t, f) with $t + f \leq 1$, equivalent: φ: (t, u, f) with $u = 1 - t - f$

Probabilistic Interpretation of QCL

Interpretation in probability spaces: $[\![\varphi]\!]_\rho$ probability that φ holds (ρ gives probability of atomic variables).
$\rho \vDash \varphi\colon (t, f) \iff [\![\varphi]\!]_\rho \in [t, 1 - f]$

Lemma

For all rules, formulas φ and ψ that share no atomic propositions, and contexts ρ, if the premise sequents hold for ρ, then so does the conclusion.

Corollary

If φ is linear (each atomic proposition appears at most once) and a proof π of $\Gamma \vdash \varphi\colon c$ only uses base rules and introduction rules, then $\Gamma \vdash \varphi\colon c$ holds for all contexts ρ.

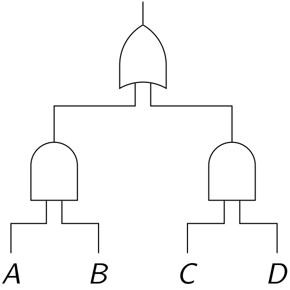
Fault Trees

Fault trees:

- industry standard
- represent fault propagation
- fault at root $\iff$ failure

Quantitative analysis:

- assign fault probabilities to base events
- compute failure probability



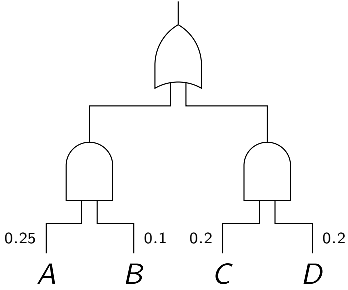
Fault Trees

Fault trees:

- industry standard
- represent fault propagation
- fault at root $\iff$ failure

Quantitative analysis:

- assign fault probabilities to base events
- compute failure probability



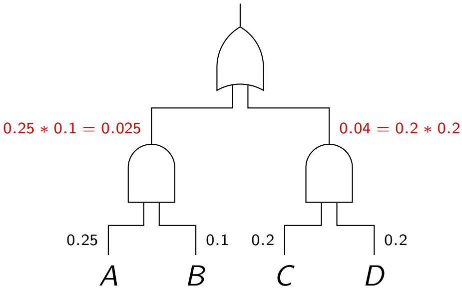
Fault Trees

Fault trees:

- industry standard
- represent fault propagation
- fault at root $\iff$ failure

Quantitative analysis:

- assign fault probabilities to base events
- compute failure probability



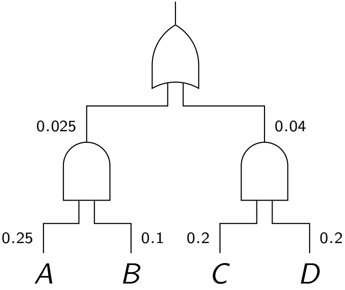
Fault Trees

Fault trees:

- industry standard
- represent fault propagation
- fault at root $\iff$ failure

Quantitative analysis:

- assign fault probabilities to base events
- compute failure probability



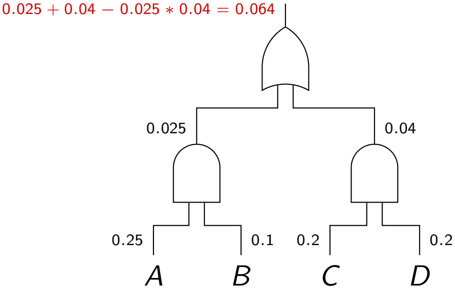
Fault Trees

Fault trees:

- industry standard
- represent fault propagation
- fault at root $\iff$ failure

Quantitative analysis:

- assign fault probabilities to base events
- compute failure probability



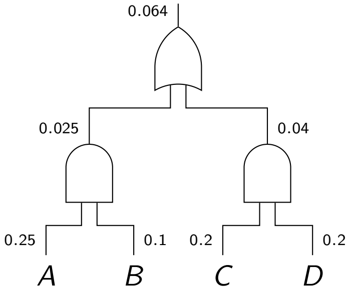
Fault Trees

Fault trees:

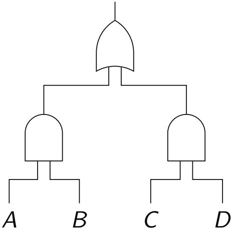
- industry standard
- represent fault propagation
- fault at root $\iff$ failure

Quantitative analysis:

- assign fault probabilities to base events
- compute failure probability



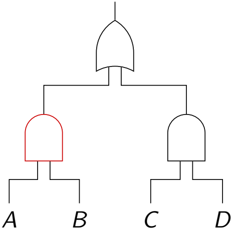
Translation to QCL Proof Trees



$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash A} \quad \dfrac{}{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \qquad \dfrac{\dfrac{}{\Gamma \vdash C} \quad \dfrac{}{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$

- dualisation: propagation of faults $\rightarrow$ confidence
- Γ: contains hypotheses

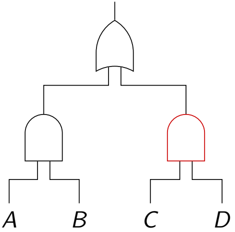
Translation to QCL Proof Trees



$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash A} \qquad \dfrac{}{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \qquad \dfrac{\dfrac{}{\Gamma \vdash C} \qquad \dfrac{}{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$

- dualisation: propagation of faults $\rightarrow$ confidence
- Γ: contains hypotheses

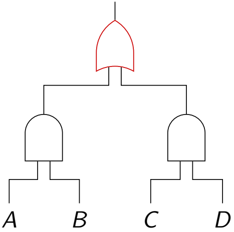
Translation to QCL Proof Trees



$$\dfrac{\dfrac{\overline{\Gamma \vdash A} \quad \overline{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \quad \dfrac{\overline{\Gamma \vdash C} \quad \overline{\Gamma \vdash D}}{\color{red}{\Gamma \vdash C \vee D}}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$

- dualisation: propagation of faults $\rightarrow$ confidence
- Γ: contains hypotheses

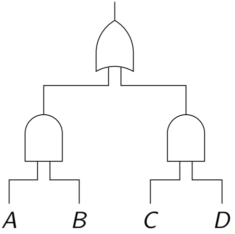
Translation to QCL Proof Trees



$$\dfrac{\dfrac{\overline{\Gamma \vdash A} \quad \overline{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \qquad \dfrac{\overline{\Gamma \vdash C} \quad \overline{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$

- dualisation: propagation of faults $\rightarrow$ confidence
- Γ: contains hypotheses

Translation to QCL Proof Trees



$$\dfrac{\dfrac{\overline{\Gamma \vdash A} \quad \overline{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \quad \dfrac{\overline{\Gamma \vdash C} \quad \overline{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$

- dualisation: propagation of faults $\rightarrow$ confidence
- Γ: contains hypotheses

$$\Gamma \vdash (A \vee B) \wedge (C \vee D) \colon ((t_A + t_B - t_A t_B)(t_C + t_D - t_C t_D), \\ f_A f_B + f_C f_D - f_A f_B f_C f_D)$$

$$\Gamma \vdash \varphi \colon (g_t(c_1, \ldots, c_n), g_f(c_1, \ldots, c_n))$$

Confidence Functions

Confidence function

$c\colon \mathbb{R}_+ \to \mathbb{C}\ \left(= \left\{(t, f) \in [0,1]^2 \,\middle|\, t + f \leq 1\right\}\right)$

- takes resources (time, money, etc.)
- returns confidence

Given:

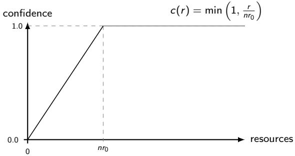
- proof of $\Gamma \vdash \varphi$: (t, f)
- confidence functions c_i's for hypotheses in Γ
- resources r_i spent on hypotheses

positive confidence $t = g_t(c_1(r_1), \ldots, c_n(r_n))$

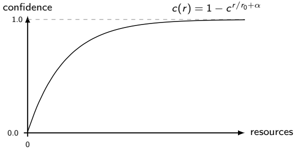
Examples of Confidence Functions

Here: negative confidence $f = 0$.

- complete test suite



- independent test suite



- SRGMs

TRAP as Optimisation Problem

TRAP

Given:

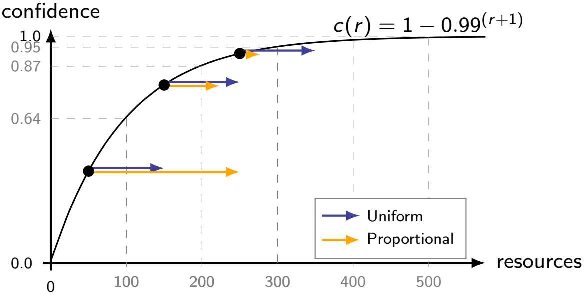
- fault tree
- confidence functions c_i for modules
- resources r_i spent on modules
- resources r to spend

maximise $g_t(c_1(r_1 + r'_1), \ldots, c_n(r_n + r'_n))$ under $\sum_{i=1}^{n} r'_i \leq r$
$\rightarrow$ constrained optimisation problem

Experimental Competitors

Competitors:

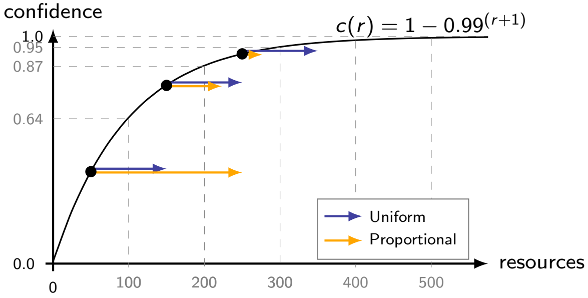
- uniform
- “inverse proportional”



Experimental Competitors

Competitors:

- uniform
- “inverse proportional”

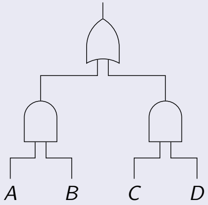


- naive
- architecture unaware

First Experiment

RQ1: how much confidence does our approach gain?

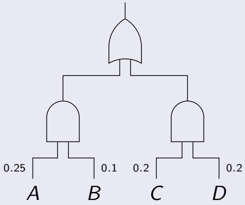
Method



First Experiment

RQ1: how much confidence does our approach gain?

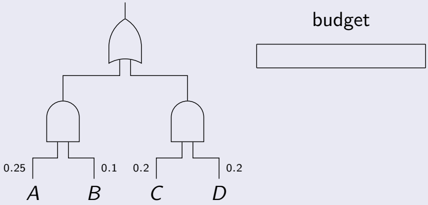
Method



First Experiment

RQ1: how much confidence does our approach gain?

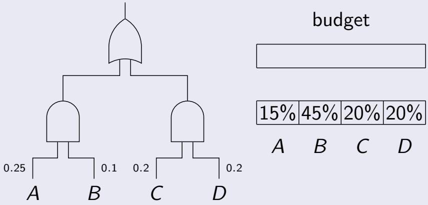
Method



First Experiment

RQ1: how much confidence does our approach gain?

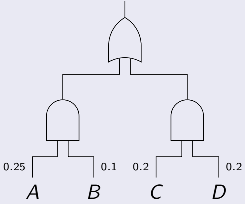
Method

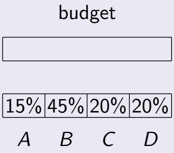


First Experiment

RQ1: how much confidence does our approach gain?

Method





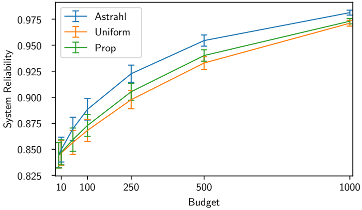
r: `Astrahl`'s score

r': competitor's score

relative difference:

$$\frac{(1-r)-(1-r')}{1-r}$$

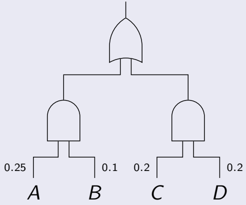
	Astrahl	Uniform		Proportional	
Budget	Score	Score	Diff %	Score	Diff %
1	.8445	.8442	-0.19	.8442	-0.19
10	.8498	.8465	-2.20	.8471	-1.80
50	.8697	.8565	-10.13	.8593	-7.98
100	.8884	.8682	-18.10	.8729	-13.89
250	.9226	.8976	-32.30	.9053	-22.35
500	.9544	.9329	-47.15	.9400	-31.58
1000	.9812	.9711	-53.72	.9730	-43.62



Second Experiment

RQ2: is the gain in confidence linked to a gain in reliability?

Method



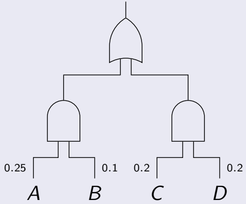
budget

A	B	C	D
15%	45%	20%	20%

Second Experiment

RQ2: is the gain in confidence linked to a gain in reliability?

Method



budget

A	B	C	D
15%	45%	20%	20%

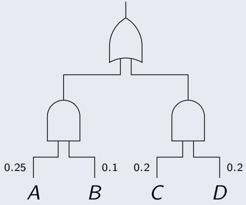
faults

A	B	C	D
4	6	5	4

Second Experiment

RQ2: is the gain in confidence linked to a gain in reliability?

Method



budget

15%	45%	20%	20%
A	B	C	D

faults

4	6	5	4
A	B	C	D

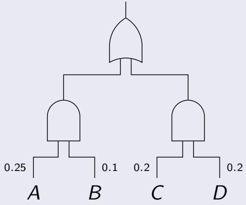
tests

3	9	4	4
A	B	C	D

Second Experiment

RQ2: is the gain in confidence linked to a gain in reliability?

Method



budget

A	B	C	D
15%	45%	20%	20%

faults

A	B	C	D
4	6	5	4

tests

A	B	C	D
3	9	4	4

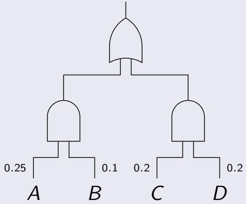
final faults

A	B	C	D
4	3	4	2

Second Experiment

RQ2: is the gain in confidence linked to a gain in reliability?

Method



budget

15%	45%	20%	20%
A	*B*	*C*	*D*

faults

4	6	5	4
A	*B*	*C*	*D*

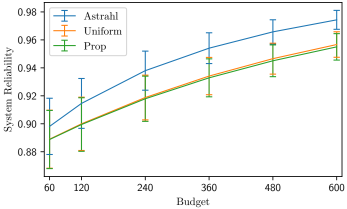
tests

3	9	4	4
A	*B*	*C*	*D*

final faults

4	3	4	2
A	*B*	*C*	*D*

	Astrahl	Uniform		Proportional	
Budget	Score	Score	Diff %	Score	Diff %
60	.8982	.8890	-9.04	.8887	-9.33
120	.9146	.9000	-17.10	.8995	-17.68
240	.9380	.9188	-30.97	.9179	-32.42
360	.9541	.9341	-43.57	.9329	-46.19
480	.9657	.9466	-55.69	.9451	-60.06
600	.9743	.9567	-68.48	.9550	-75.10



Future Experiments

Compare to architecture-aware TRA strategies

- parallel-series architecture $\rightarrow$ fault tree
- using same confidence functions for modules (SRGM)
- same function to optimise

Experiments on larger fault trees

- numerical optimisation less efficient
- should still be better than not taking architecture into account (even better)

Other Frameworks

Other TRAPs

- optimise $t + f$
- optimise resources (for fixed t / fixed $t + f$)

Dynamic TRAP

- take test results into account $\rightarrow$ many faults = loss of confidence
- number of faults unknown $\rightarrow$ not optimisation problem
- use Bayesian reasoning to guess number of faults

Others

- test prioritisation
- ...?

Conclusion

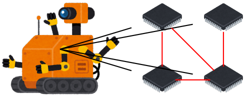
This work

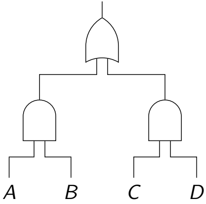
- Quantitative Confidence Logic proof rules
- translation fault tree $\rightarrow$ proof tree
- architecture-aware TRA strategy
- experimental validation

Future work

- logical side:
  - QCL and truth (e.g., $\vdash A \Rightarrow A \colon (1, 0)$)
  - QCL and time
  - equip logic with confidence
- practical side:
  - dynamic TRAP

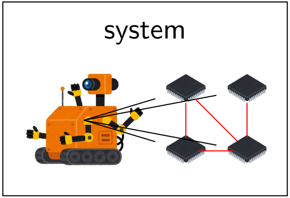
system

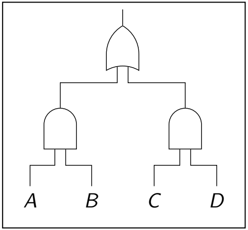




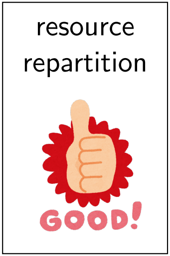
$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash A} \qquad \dfrac{}{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \qquad \dfrac{\dfrac{}{\Gamma \vdash C} \qquad \dfrac{}{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$

resource repartition





$$\dfrac{\dfrac{\overline{\Gamma \vdash A} \quad \overline{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \qquad \dfrac{\overline{\Gamma \vdash C} \quad \overline{\Gamma \vdash D}}{\Gamma \vdash C \vee D}}{\Gamma \vdash (A \vee B) \wedge (C \vee D)}$$



Thank you for your attention!