



SPLASH 2024

California, United States of America
2024.10.20. - 2024.10.25.

JOONGHOON LEE

Software Analysis Laboratory
Korea University

1. ä어 p

ø m ~ -i È DÐ ò ° O O P S L A Ð | 8 D \ X ä . « Y E 8 t | q Ä Î X
À Ì ä % ^ ~ È 4 - X à Ì D T ä . t ^ S P L A S H 2 0 2 4 X 3 | Û H O O P S L A Ä % È à ,
S A S Ä ñ] X i ^ 3 | ä x Y E ä Ä h Ø 8 X ä . Y E O » @ ½ Ø D ø D µ t
õ X à • \ ä .

2. 논8 \

ì ì - E t ' Ð D ½ - \ ä " f @ Ð \$ à ð ì ' | t ä . D Ì ' € , ä Ä ^ Ä
ì Ì q ð ì ' \ ä .

D ü 정 . Ð ~ Ä • @ æ m Ð \ D | D ^ È X à \ t Ä 쫘 € O è X E ð µ
X \$ ^ ä . † ü € O , Ì O Ø Ä Ä D D \ • Ì | Ì ä È à , ü p ð Ì ä O Ø ä \
• Ì @ æ Ð Y E \ Ä Ä Î t > D ô X ä . ' • Ä ð x , ' © t ° 8 æ m p `
P Ø Ø ç E < Ü 1 € Á Ü 8 " p , 4 • 8 \ ' © @ | à € T high-level Ð \ t Ä
. D f t | p , t ü h ä . \ Ü t ç D ' © D € ì X O " ' \$, f t p , Y E ¥ t
ì Ì \ ~ X ' ^ ' ì ^ " - E ä ü \ , f t | " t Ð ä . ° m \ • Ì | Î t
t O \ È L 9 È à , Ü t € q X ä à è t " D % O Ð ~ \$ Ð Ä E Ð Ä \
ð µ D ^ ä . S P L A S H « Y E ¥ Ð Ä) X • È • \ | @ ' \ Ä % " % - ¥ D ø -
ø f t È ä . \ ¥ ' O X ~ Ì p ñ D ø - U x t P t t ø Ä , t Ý Ð Î t Ä Ä O
L 8 t ä . ä Ì ð Ì ° Ä Ì à X • Ä Ð Î D Ý ó ä Ü " Î t ø à , \ ð µ Î t t |
ä à ä Ð X ä . ø ~ x Ä T 4 ¥ È à æ ¥ O ' ' Ü (Q D » X à È ½ Ð h '
^ È X f ä .

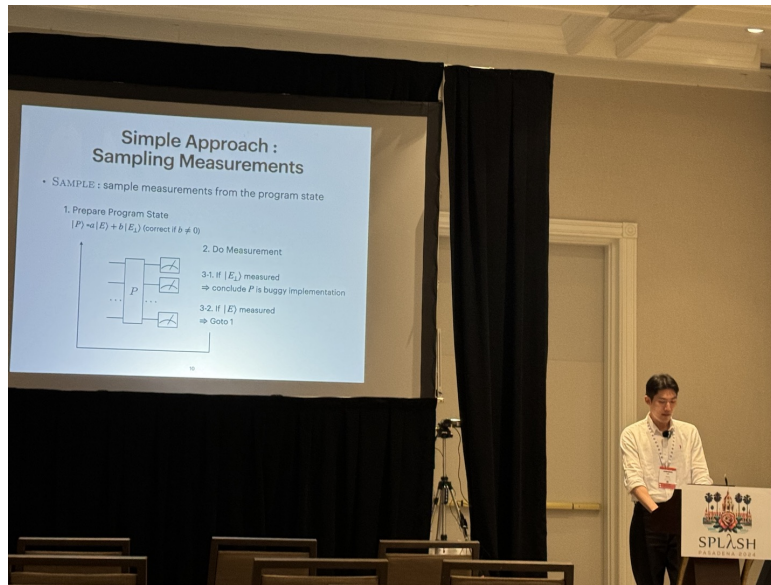


Figure 1: Sampling Measurements

$\mathcal{P} = \{P\}$ is a set of program states. $|P\rangle = a|E_1\rangle + b|E_2\rangle$ is a superposition of two states. The circuit in Figure 1 shows a box P that takes an input and produces an output. The output is measured at two points, E_1 and E_2 . If the measurement at E_1 returns a non-zero value, it indicates a buggy implementation. If the measurement at E_2 returns a non-zero value, the process loops back to the start.

1. $\mathcal{P} = \{P\}$ is a set of program states. $|P\rangle = a|E_1\rangle + b|E_2\rangle$ is a superposition of two states. The circuit in Figure 1 shows a box P that takes an input and produces an output. The output is measured at two points, E_1 and E_2 . If the measurement at E_1 returns a non-zero value, it indicates a buggy implementation. If the measurement at E_2 returns a non-zero value, the process loops back to the start.

H < \ Đ ` ^ ä.

2. \ 초 t 요하ä. ì ì \ | Á ä < p • 쫘" p, \ ^ € , 8 \$...t © , ^
 ~ ì à eømE ä \$ | \ Đ Ñ ` ^ " f ä .15 , \ pX ^ 5 , HD
 \ 1 (° " ì Xä . ' (< ø , | X 8 DÈt • 8 \ ' © @ O | O
 ... (ä . ì ì \ | ä < t P ØØ ühX p , t 3XLD 1 ^ • 쫘ä . ì } 또
 \ XE ä t \ | 1DT à ½ 쓰à 8 \$...LÀ Ü DT ` ` ` f t ä .

3. È8 내용@ 대체로 D. 하ä. -ä ¥ ° \$ ^ X € , t ÈXQö Ü t Èä . ' »
 LDäp~ ö » Xt ' LÀ X" q t ^ È" p , -ÆÝ @ pO pO | XÀ J
 " .äx \ äĐ Ä ' • \ ' © @ € , äx -Æt < È8XOÄ ^ à , '
 \ Đ ä ' (È8äÄ ' • Ä" Á" HD ^ Èä . 1 ^ , ' • ðl XÜ" P Ø
 Ø ühXÈ8 X~ " xt ^ t -X" X½Đ t) • D © ` ^ " " p ,
 | 8 -ð ü Đ XXÈ8ü pX -t äDýà • ä ^ Æ öÄ ` ^ Èä . äL
 \ | XE ä t è , q \ à öÄ D ` ^ Df ä .

3. \ 참석

ä' \ , | X \ ÜÜĐÄ% ä öÈ Ä • . " äÈp \ | äÈä . xÁ tÈX
 ðl 8 Ä Ä | Æ Xä • \ ä .

Quantum Control Machine: The Limits of Control Flow in Quantum Programming. t ðl Đ
 " ' • \ ø " X control flow | \$...XO t reverse jump Ä...D H\ä . à òèOĐ
 " conditional jump\ - ÄxOĐ ô4 ĐO | äL < \ ä % X | program counter
 Đ \$ ` ^ ä . XÄ ì ' • \ ø " D ä % ` L" ÄxO © ÁÜĐ " t Æ ä .
 % , conditional jump | - © Xt , OÈä U ` < \ äL program counter t ì | Ää . XÄ ì

t ...9' | ' • \ø" Ð " ø \ -©` Æä.' • Æt, O ' • ôè ü Ð
 " à ð° • " ÈO-%, \ \ " p,t" norm-preserving ð° < \ , x' • ÁÜ" ø
 norm t t t ' | \ä.ø-à à x conditional jump\ © ÁÜÐ \ state transition
 D O | t ^ ¥\ norm t ÝXÆ ä. • " t 8 | t° X O t reverse jump
 Ä...D H\ä. ¥ î ^ ÈX ðl à,° ...% ^ Æ \ Ä ~ l ÁXî -ø^ Æ -ø^ Æ
 ä Èä. -ä QRAM(Quantum Random-Access Memory)X Ä...ü (t " î î t ^ D
 ^ Äl , ' Èà O t Ð t -X" ...9' ä \ " \ Ä t -hD ô x x Á x ðl ä.

Quarl: A Learning-Based Quantum Circuit Optimizer. ' • Æ \ \ T Ð \ ðl ä.

PLDIÐ D. \ © \ X ðl \ Quartz@ Queso ^ Èà, rule-based Æ \ Ä X D µ t Û | \ ð°
 D %XÄl , T @ X CNOT Æ t, | -©X" Æ \ | >" ä. t Ð t Quarl@ Y µ
 O X Æ \ \ T) • D Ü \ ä.° rule-based Æ \ Ä X D %X ä ô t • Ð X \ Ä \
 ^ T Æ t Ä X J t Ð \ Ä t - \ ä.%, L ' ‡ ^ X Ä X ¥ ' @ X Æ \ |
 Ý X Æ ä. X Äl t ðl Ð " ø) • t \ t D Ø D p \ ä. è O < \ " \$ ^ \$
 CNOT Æ t, Î D Ä " D " (x Ä X t Äl , \ ... < \ T @ CNOT Æ t, | -©
 X" Æ \ Ð Äl ` Ä ^ ä " f t ä. \ Ð " ä % Ü Ð \ , t Æ ' O \ È 8 ^ "
 p, @ Ü Û H Ä ðl D ¥ < @ ° ü | ô ä à ^ ä. i 1 è Ä Ð greedy X Æ
 Ä X X Ä J " Æ T optimal \ \ ... ° ü Ð Äl ` ^ Ä J D L ö È " p, ä ° ü @ Y µ
 O X \ ü) • D µ t t ° X î e ø m Æ ä @ \ ä.

Enhancing Static Analysis for Practical Bug Detection: An LLM-Integrated Approach. LLM

D t © t , D % \ ä " ' È ' È \ © Ð t L \$ \ Ð 8 X ä. t ðl Ð "
 use-before-initialization , ø | >" p Ñ \ ä. t ø t -X" - x ä UBI , ø | >"
 O 8 ä % , O UBITect | ä \ ä % X t T " - Ü ü \ x t } 60% Äl
 ä , ` ^ ä. %, ~ 8 Ä 40% H Ð false positive , ø ä t t - ` Ä ^ à, t ½° ä D ä

èO t LLMD t ©\ä. | ä´, sscanf h X-4 DDPX" ƙÜ ^ ä t, ´Ä
 ø h X x • ä @ OT ÁÜì | \ä.t ðl ð " tì \ *post-constraint* | t © t è }
 À D > à • \ä. *UBITect* ð undecidable \ ° ½° LLMD è Ä < \ ä % X p \ \
 decided ° ü | Äœ \ ä . l } LLMD ä % \ ° ü U \ è D t h X D " \ ½° ,
 t ù ô | õ X ì ä Ü LLMD ä % X " ñ ‡ è Ä X Á 8 ' © D µ t ° ` D , ä " üt
 x Á t È ä . t ð l ð " t ü D *Progressive Prompt* | ð \ ä . 10 t Á X , ø | > "
 ñ < @ ° ü Ä ~ T à ü) • Ä - " ð l ä .

4. 맺 < p

« Y E 8 x p ä \ Ä t | \ ä " Ý ð q D ð Î t ^ " p , ô ì ^ ð µ \ U
 „ x Ä ì q µ ý E È 4 - ` ^ È ä . 8 Ä ð (ð l • ä ü ô < \ T X X @
 ð x Á J E O µ ð " X à , µ µ \ Ä ^ < \ ð l X " p Î @ Ù O € ì È ä à Ý \ ä .
 1 ^ x O 7 ð ì ô X ð l • ä D Á ì ~ È T ð x ø ì \ à O \ 4 , ^ È ä .
 t ^ ð " 2 • | 8 \ " p , ä L ð " 1 • \ | 8 \ X - | ä ð t ø ä . ð l Ä Ä t ü
 Ü " \$ Y ü P Ø , h Ø ð l X " , l O Ø , ø - à \ D € O Y E œ ¥ L Ä Ä Ä ü à
 ð l ä D ì h \ " à , ä Ø Ä ì < \ - Ü ½ È ä .

