OOPSLA 2019

Athens, Greece

Junhee Lee
Software Analysis Laboratory
Korea University

19.10.21 - 19.10.25

1 ″

OOPSLA'19 Y Œ Ð ì ¤ 0 \ | X 0 t ä @ T ä . t ˆ Y Œ 8 @ d 0 | X à ″ p , ø t ″ OOPSLA ″ PL Y Œ t 0 L 8 Ð D · \ „ | X ð l | Î t ü ˆ D ƒ X à , ¬ ð \ 8 ¤ ä ° L ä \ Ì \ | 8 D Î t ü ˆ D ƒ X ä . ° ` € 0 Ð X • t , Y Œ ´ ! t Ð ″ p È L Ð ä À J X À Ì , 0 ¸ ƒ t Î @ u \ Y Œ ä .

2 포¤터 \

포¤터 \ ì ¤ 0 \ | X t 0 ´ @ ù ð \ t | 0 À Ì ã ″ ¬ Œ Ð Þ ° $ … D X ″ Œ ¥ ¨ ü < \ ì ` ˆ ä ″ ƒ D 0 à ä . ˜ L Ð \ ` L ì ¤ 0 Ð Œ motivation ½ € 0 œ œ ˆ $ … D X à ˆ ″ p , ã ″ ¬ Œ ¼ t D ô È L p À è t X T È goal $ … D ã T È à Ù ä à X à „ 8 ä . ø ä L Ð ″ € à • X ″ 8 Ì ˜ $ … X à \ D t ´ ‘ ä Ø ° ü | ô ì ü • à È L D 9 È ″ p , t ˆ Ð ″ 8 $ … D ã T È ˜ » (X | X ″ | ¬ ¤ t ” \ t Œ (X X ä ″ p p ″ ” | à t \ Ý € • ä . t ¬ Œ @ T ¨ ¬ t $ X | ˜ ° | L Œ Ð Œ \ (X X t H ä ″ ƒ D ° • ä . ø ˜ 8 € 0 (ü ˆ $ … ˆ ä . ø ˜ ø ä L € 0 ″ Q D ô à $ … X • t 8 $ … X à , ˜ L D ä < t ä Ø ° ü , à 0 t X t D t ´ | $ … ˆ ″ p ˜ „ . X X ƒ ä . < ` $ • È • D t ´ € 0 $ … X | ″ ¬ Œ Ä ˆ È à , 8 | ã T È ° ü | \ < ´ ô à D t ´ | t ô ¬ Œ Ä ˆ È ä .

코멘트 ° ¬ ð l Ð t l ¬ ð ì \ È 8 D X Ä ¬ Œ @ Æ È ä . < ` ø ð X 8 D ″ ¬ Œ ä @ \ | ä < ì $ À J ″ x Ä ˆ ″ p , < \ memory-leak Ð \ ° ¬ ð l ″ è t l Œ Æ ´ Q t ä ° ‹ X ä . | 0 „ t ‹ X X T X ¸ ″ ” ¸ t 0 D 써 ü ˆ Ð ?” ä . , | 8 < \ ˆ Ð à ; à ICSE Ð È ä à X È L ‹ @ ° ü | » D ˆ D p | à Ð \ T X ¸ Ä ‹ X ä .

선점 < ì ¤ 0 Ð ¥ ø a ˆ X t | t , ° ¬ ð l ´ ¤ 8 | x ″ ð l x À | € T \ Ð ô t À J X ä ″ t ä . < ì ¤ 0 © Ð ”for C” | ` (° t ä . t x # ″ Œ ø Œ ″ \ ö ´ H # È ″ p , ì ¤ 0 | X t ¥ < ä È X È 8 t ”t ð l C X T ¨ ¬ - D à X ″ ƒ t Ð ?” ä . Ý ô ä Java X T ¨ ¬ - D ð l X ″ ¬ Œ t Î @ ƒ X ä . Managed language Ð T ¨ ¬ | ¨ ü < \ ¬ X ″ 8 p , Ü \ ü D Ì ö È ″ p ″ ˜ Ä Î t X ″ À € „ t È 8 D t T ä . , ° ¬ € à • X ″ 8 | … U ˆ Ü ì ´ ″ 8 ¥ t ˜ ø ¼ t Æ È ä . Ü ¤ \ overview ‘ motivation Ð ä $ … D X È L . À J D L ˆ ″ p , ä \ $ … X $ à ô È L motivation ¸ @ overview | T ä ä X t $ … X Œ È ä . Ð ˜ L $ Ä 8 | x ″ ƒ t D È | t , 8 X input output ″ U ä X Œ Ü ì ´ ″ ƒ t ‹ D ƒ ä .

3 OOPSLA \

3.1 새로 L Œ ð l ä

A Path to DOT: Formalizing Fully Path-Dependent Types t ð l ″ 8 t °) Ý t Ð Ä À X à \ ˆ ä ″ x Á t J Œ ¨ @ ð l t ä . t ð l ″ Scala ¸ ´ Ð path-dependent type D È © X À Ì ¬ X À … Ü ¤ \ Dependent Object Type (DOT) t t | \ formalize X ″ » X ″ D À \ ä . ø ¬ à , t 8 | t ° X 0 t DOT D U ¥ X ì path dependent type D À Ð X ″ pDOT D Ü \ ä .

Type syetem „ | X 8 | ø Ù H ˜ ° • 0 L 8 Ð p T • 8 ˆ 0 \ ä . \ à X \ ø ˜ ¸ ´ ä @ ″ Á T À … D X X à (e.g., tree) t À … D t © X ì ä ‘ \ instance X type D È \ X ` ˆ Œ T Õ ″ ä (e.g., red-balck tree). t ä @ \ ø ˜ D ´ Ä t à ¨ ( < \ ` ˆ Œ Õ ″ ä . t L , t Œ õ ¡ \ type system D ˜ formalize X ″ calculus (formal system) | • x X ″ ƒ t ä ° ″ X ä à \ ä . t | µ t type system X $ X | > p ˜ , È \ ´ ¸ ´ | • x X ″ p Ä À D ä à \ ä . t Ð t ì \ calculus | ˜ • x X À » X t , unsound type system Ð \ ø ˜ D X Œ p ˜ ä ° \ x (4 Ì t ¥ \ ¸ ´ | ¬ © X Œ ä .

´ \ X | ˜ t t X À » ˆ À Ì ø ˜ Ä 8 | ˜ € È ä à Ý t Ü ″ @ l ´ t à motivating t ″ © \ @ t | t ° X 0 \ ´ \ X | 0 t ð l @ J „ Æ t D P X à non-trivial \ € „ D Ä • Ü ì ´ t ì ˆ 0 L 8 t | à Ý \ ä . ˜ Ð type D \ õ € X Œ t Å t ü Ì \ ð l x ƒ ä .

Duet: An Expressive Higher-Order Language and Linear Type System for Statically Enforcing Di erential Privacy Differential privacy ″ ˜ Ð security ½ Ð ″ X à Ü Ü À | ` P t | à \ ä . Differnetial privacy 4 Ç t Ð t , x ô ì h dataset D D „ X ì © \ ô | ü , t ù ô Ð x ô Ü ì ˜ À J Œ T X ″ Ü ¤ \ D Ð \ ä . © ´ Ð differential t ä ´ t ″ t P t Y < \ differential \

X ´ ˆ 0 L 8 t ä. Informal X Œ Ð X t, \ x X privacy Ì t ä x P p t 0 K X „ ° ü p X Ù | ` L differential privacy | À ¨ ä à Ð \ ä à \ ä.

t ð l Ð À X ″ 8 ″ differential privacy ä ° ″ h Ð Ä ¹ \ ø ¨ t differential privacy | õ X ″ À ″ 8 Á U x \ ä ″ ƒ t ä. t | 8 t Ü X ″ t ° E @ differential privacy | • … ` ˆ ″ language | • x \ ƒ t ä. t language X ¹ t \ t | t, < differential privacy numeric \ X t 0 L 8 Ð linear type system D D Ý \ ü privacy @ sesntivity | Ä ° X ″ P ¸ ´ | Ü \ t ä.

t ù | 8 X • 8 \ ´ © @ differential privacy @ t | l X ″ machine learning, • … Ð ð x type system ñ ø ¬ L D l | ` ´ © t Î D • 8 ˆ E @ X À » ˆ À Ì, e ø m à ì ˆ Œ À ü ü x ƒ @ U ä \ ƒ ä. t | 8 \ ˜ Ð Å X $ à \ ä.

3.2 ¬ ㅁ ˆ 는 ð l ä

BDA: Practical Dependence Analysis for Binary Executables by Unbiased Whole-Program Path Sampling and Per-Path Abstract Interpretation t ð l ″ t ¬ Ð t dependency | „ X ″ ð l t ä. t | 8 t X à • \ ƒ @ è X ä. t ¬ Ð t U X t Ä sound \ „ 0 | Ì Ü ″ ƒ t ä ° ´ 5 0 L 8 Ð, (¤ Ä \ „ X ä ″ ƒ t ä. t L ¹ (¤ Ð X ° X t H À \, ¥ \ ä ‘ \ (¤ | à t Ä] t Ü X ″ ƒ t t | 8 X u ì t ä.

t ð l X ¤ ì x ¸ ″ ‹ @ 8 $ x ƒ ä. (¤ Ä \ „ X t ù ð ˆ U \ „ ° ü ˜ $ À Ì, | < \ ″ t | è statement coverage | , ¬ ″) Ý < \ „ D \ ä à t „ D ˜ ˆ ä à Ð ` Æ ä. ø À Ì, 8 | ô | ì \ T | \ data dependency | „ X ″ 8 \ $ \ ä t, statement coverage | ˜ ì 1 X ″ Ä \ Ä © „ \ ô | » 0 Ð € q X À J Œ ä.

Static Analysis with Demand-Driven Value Re nement t ð l ″ 8 | \ € È ä ″ • Œ t ä ´ 0 µ Ð ¨ ″ ä. t ð l X à • X ″ ƒ @ Javascript | U X à ` t Œ „ X ″ ƒ x p, 8 | Ð ˜ € È ä à • | ″ @ concrete \ 8 Á D Õ < \ ´ \ X | Ü \ t ä. U Ä Ý À ″ 8 | | x t ° E (e.g., context-senstivity tuning) D t © X ì t ° X ″ ƒ t D È |, U Ä Ý (¬ ″ Ð x (dynamic property read/write) D E X à t | ˜ Ü ì ´ ″ example D > D t Ð \ t ° E D Ü ˆ ä ″ • Œ t à ä.

t ð l ″ • ¤ l ½ ¸ Ð U Ä Ð p ¥ D | X ″ ƒ t dynamic property read/write | à À X à t | ˜ „ X 0 \ ) • D Ü X ä. | ä t `source[name] = func` t | ″ T Ü ″ `source` À X dynamic field ”name” Ð h `func` D ¥ X ″ statement t ä. ä ‰ Ð X ″ field D } ″ T Ü ″ „ 0 X U Ä | Ý (¬ ″ ü Ð x t ä. t ð l ″ t @ t U Ä Ð p ¥ D ü ″ statement | Ì ˜ t, „ state | func t È ˆ ″ < \ ˜ à, X ½ ° Ð t T U \ D Ä ° X ″) • D Ü X ä.

3.3 PL¤러´ ðl

Language-Integrated Privacy-Aware Distributed Queries t ðl 0µÐ ¨″ t ″ database „|X8 | À à @ 8 | Á €À Jà type system D HXà t| t©Xì 8 | t°\ tä. Ð PL Y Œä´ ðl|à • ä.

t ðl €à•\8 ″äLü ä.ìì pt0 t¤Ð ¹ ü¬Ð õX″8 |Ý tô•.tL, pt0 t¤|´¤ \ ü` À, Å½ <\ 8, À, ¸ join `ÀÐ 0| |ì<¤ Ît (tœäà\ä. ,t|˜»$ Xt privacy hÀ0Ä\ä. t ðlÐ €à• X″8 ″ |t„Ü| ÀXt Ä |ì<¤ ¥‹@ placement |àt″8 tä.t|8@t8 | t°X0 t type system D HXà t| t©Xì 8 | t°\ä.t°)Ý@ \ø¨ i1ü D·Xä: information-flow type system D t© Xì candidate | type-directed)Ý<\ ˜ôXà constraint solving 8 | €´ ¥‹@ placement | àt″ ƒ tä.

4 ø리¤

ø¬¤X „ 0″ ‘DÐ X ¤ètü″ Ît ì•″p, <\ À „Xä″ xÁD Xä.½Ð″X˜ t ™ ´ˆà, Ä- p ð 0µt 8p¬Ð ˆà, •Ù(ä@ 8 ‘½<\ € ü(´ˆÈä. ,DX0Ä Ä- ÎD |xÐ %D 9<t •Øt 9ä ¨4 LÝD DX0 ,|@ 9″ ƒD l½` ˆÈä.$Ð″ À| øðÀ Ît ´PôôXà pÀÄ ÎD p @ Ø\ •Œt äÈä.

ø¬¤Ð ¬ä |0 ˆXƒ@ À ä.l ÀÄ\€Éˆ DL Dl\ô¬¤, t L|à ñÐ\Ì ãXø¬¤ àT@ (à ät˜ <äD ü ˆDL 0 ˆÈ″p, €„ €T8ˆ´ Ît Dlàä.ø¬¤ LÝ@ ˆ|| ¥ Ûˆ Èä.Ltô˜|Ð Ä €D €´ #″ àÐ Ä€ Ä|t| ¹″ø¬¤ µ ¤ÀÄ ˆÈà, l´ DŒ| I¥ˆ •ù x Œ¤Ð x LÝÄ ˆÈ″p tä@ …ÛÐ € ÞÀ JXä. €„ 4 Üà•ù tÈä.ø¬¤Ð ¹tˆX @ Y08\Ì ôXø¬¤ 8•|´ Ð ˜ü ˆÈä″ tä. |ä´, ÀX íD ˜À´″ ST.″ ΣT \ ðìˆà, 8• L àÐ Λ| ð″ ƒÄ ììX JŒ ü ˆÈä.

5 맺L말

<\ Ìm\ YŒ ä. ì%t p ¨´À″ \ä@ ˆÈÀÌ, ¨P 8 | ˜ X Xà .@ Dt ´|´à‹@ °ü|´″ðlätÈä.X8D ¨0″ \äÄ ˆÈ″p, øð \XÈ8Ð″ - 0t ðl\ t 8 t°t À J″À|<Èä. tÐ¬ ″ \Ð″ t 0 t´ LÀ ©t ″À|ü\ ì ″ƒ Xä.äL ðl″ FSE 9@ OOPSLA Ð œXŒ ƒ @p, OOPSLA x <\″ € T ¬øˆD ƒ ä. ÈÀÉ<\, ‹@ ½ØD XŒ tüà P ØØ ¬Ü½Èä.