UNIREMINGTON
CORPORACIÓN UNIVERSITARIA REMINGTON
RES. 2661 MEN JUNIO 21 DE 1996

Mundo Organizacional
Grupo de Investigación Uniremington

Ciin
INVESTIGACIÓN E INNOVACIÓN

DEPT. DE INVESTIGACIÓN
VIDOCQ

CIDE

CET-BOLIVIA

VI CONGRESO LATINOAMERICANO DE INGENIERIA DE SISTEMAS E INFORMATICA

14,15 y 16 Octubre de 2020 - 100% online

Uso de la terquedad mutante en el problema de priorización y minimización de casos de prueba

Dra. Ana Loreto González Hernández

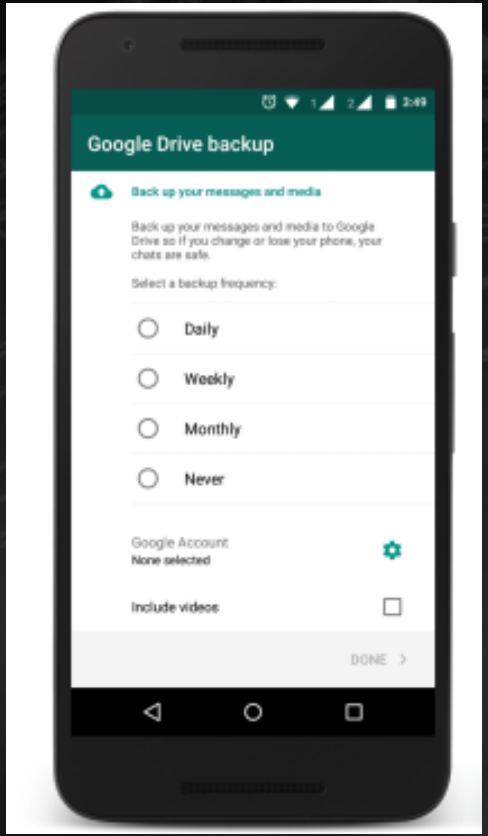
loreto@gmu.edu

14, 15 y 16 de Octubre 2020

Introducción – Software Testing

Las pruebas de software (Software Testing) se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación.

Daily, ☑ Include videos
Weekly, ☑ Include videos
Monthly, ☑ Include videos
Never



Introducción

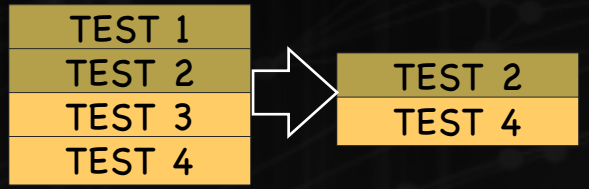
- Se suelen crear conjuntos de prueba grandes
- No se cuenta con el tiempo para ejecutarlos todos

Algunas soluciones:

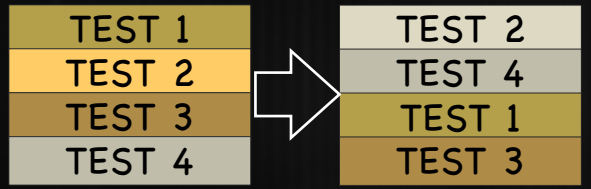
- ✓ Minimizar o reducir el tamaño del conjunto (TCM)
- ✓ Priorizar cuáles pruebas ejecutar primero (TCP)

Problema de Investigación

TCM intenta reducir el número de casos de prueba usando alguna noción de redundancia, manteniendo la satisfacción de los requisitos



TCP es un proceso para decidir el orden de ejecución de los casos de prueba en función de alguna estrategia de priorización



Pruebas de Mutación

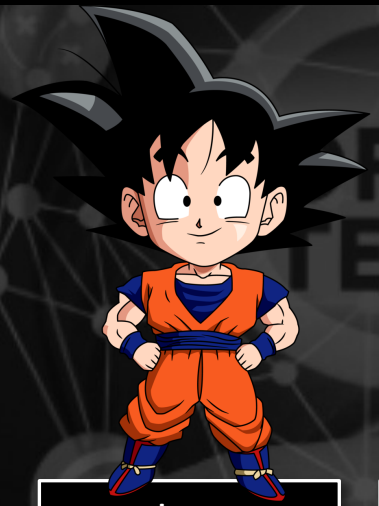
```
int max(int x, int y){
    int mx;
    mx = (x>y)?x:y;
    return mx;
}
```


```
int max(int x, int y){
    int mx;
    mx = (x<y)?x:y;
    return mx;
}
```


```
int max(int x, int y){
    int mx;
    mx = (x==y)?x:y;
    return mx;
}
```

Subsumir

Se va a formar un equipo de trabajo para un Proyecto:



- Phyton
- HTML
- Analítico

- Phyton
- javascript

- Phyton

- Phyton
- HTML

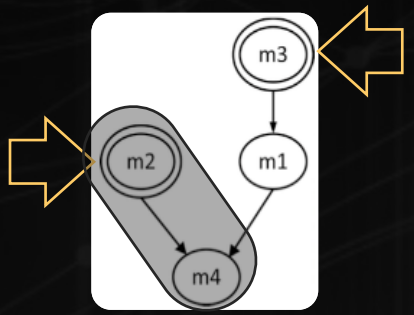
Conocimiento en:

- Phyton
- javascript
- HTML
- Analítico

Mutantes subsumidos

- Algunos mutantes son subsumidos por otros
- Un caso de prueba que mata al mutante m_a también mata a m_b

	m_1	m_2	m_3	m_4
t_1	✓	✓		✓
t_2	✓		✓	✓
t_3				✓
t_4		✓		✓



Si elegimos aquellos casos de prueba de cada nodo raíz que matan el número máximo de mutantes, el número de casos de prueba necesarios para eliminar todos los mutantes es mínimo.

Figura: Kurtz B., Ammann P., Delamaro M., Offutt J. and Deng L. "Mutant Subsumption Graphs". Mutation 2014, Cleveland, OH, March 2014.

Terquedad Mutante

Matado por

2 3 2 1

¿Qué tan difícil es matar un mutante?

TEST 1

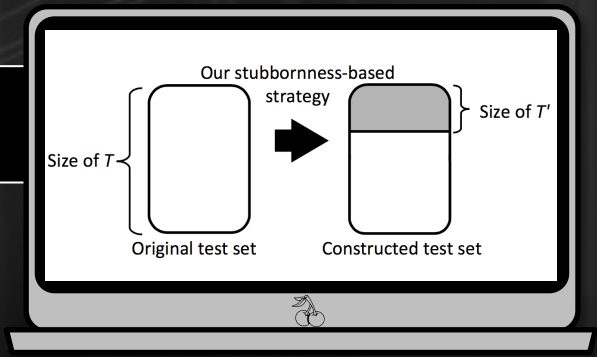
TEST 2

TEST 3

Metodología

1. Usar la terquedad mutante para **asignar puntaje a los mutantes**
2. **Asignar puntaje a cada caso de prueba** con base en el número de "mutantes tercos" que mata
3. Seleccionar los casos de prueba **con la mejor puntuación**

Todos los casos de prueba del conjunto original de mantienen
Tamaño T' = posición del1er test que satisfice el "mutation score"



Estrategias basadas en la terquedad mutante

1. Rank-Stubbornness Model [RSM]
2. Weight-Stubbornness Model [WSM]
3. Hybrid-Stubbornness Model [HSM]

Rank-Stubbornness Model

- El RSM asigna un rango a cada mutante con base en el número de casos de prueba que lo matan
- Se selecciona el caso de prueba que mata el mayor número de mutantes con el rango más alto

Tests	m_1	m_2	m_3	m_4	m_5	m_6	R_1	R_2	R_3
t_1	x	x		x	x		0	3	1
t_2	x	x				x	1	1	1
t_3	x		x	x	x		1	2	1
Rank	3	2	1	2	2	1			

Criterio de desempate: avanzar al siguiente rango

Weight-Stubbornness Model

- El WSM asigna un peso de terquedad ω a cada mutante (1/Rank)
- A cada conjunto de prueba se le asigna un puntaje W derivado de la suma de peso de cada mutante que mata

Tests	m_1	m_2	m_3	m_4	m_5	m_6
t_1	x	x		x	x	
t_2	x	x				x
t_3	x		x	x	x	
Rank	3	2	1	2	2	1
ω	0.3	0.5	1.0	0.5	0.5	1.0

Tests	m_1	m_2	m_3	m_4	m_5	m_6	W_i
t_1	0.3	0.5		0.5	0.5		1.8
t_2	0.3	0.5				1.0	1.8
t_3	0.3		1.0	0.5	0.5		2.3

Hybrid-Stubbornness Model

1. El HSM utiliza primero la estrategia RSM hasta que todos los mutantes de rango 1 son aniquilados

Tests	m_1	m_2	m_3	m_4	m_5	m_6	R_1	R_2	R_3
t_1	x	x		x	x	x	0	4	1
t_2	x	x				x	0	2	1
t_3	x		**x**	x	x		1	2	1

2. Posteriormente usa el WSM hasta matar a todos los mutantes

Criterio de desempate:
Se elige un caso de prueba al azar

Tests	m_1	m_2	m_3	m_4	m_5	m_6	W_i
t_1	0.3	0.5		0.5	0.5	0.5	1.0
t_2	0.3	0.5				0.5	1.0
t_3	0.3		1.0	0.5	0.5		--

Diseño experimental

Se utilizaron 7 programas del repositorio Siemens, usados también por Ammann et al.

ID	Program	Original sets #Tests	Original sets #Mutants	Mutation adequate #Tests	Mutation adequate #Mutants
P1	print_tokens	512	4322	512	3711
P2	print_tokens2	512	4734	512	4047
P3	replace	512	11080	509	8783
P4	schedule	512	2108	512	1838
P5	schedule2	512	2626	512	2131
P6	tcas	512	2384	506	1957
P7	totinfo	512	6693	512	5821

Métricas

Minimización

Tamaño T'

Priorización

APMK *

Tiempo

Tiempo de ejecución

*Average Percentage of Mutants Killed

- Se crearonn 30 ordenaciones para cada conunto de prueba
- Se utilizaron pruebas estadísticas con $\alpha = 0.05$ para verificar si existía diferencia significative en el desempeño de las estrategias

Minimización

Promedio de número de casos de prueba en el conjunto minimizado

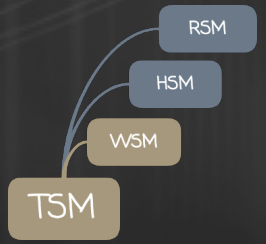
ID	TSM	RSM	WSM	HSM	OPT-GDY	ARND	SBM	RND
P1	12.4	**9.0**	10.0	**9.0**	**9.0**	28.2	361.0	399.5
P2	12.1	**8.0**	9.0	9.0	9.0	28.7	155.0	331.7
P3	45.0	**40.0**	44.0	41.0	44.4	86.2	446.0	488.4
P4	14.5	13.0	**12.0**	**12.0**	14.0	28.5	470.0	392.9
P5	17.1	**14.0**	**14.0**	**14.0**	16.0	30.8	503.0	422.0
P6	41.4	**39.0**	**39.0**	**39.0**	41.2	65.8	374.7	471.1
P7	13.3	**12.0**	13.0	13.0	16.0	30.9	362.0	421.6

Strategy	Mean rank
RSM	**1.64**
HSM	2.00
WSM	2.36
TSM	4.00

Resutados de las pruebas estadísticas

κ	Strategy	$z = (R_0 - R_\kappa)/SE$	ρ	α'	SD
1	WSM	(4.00-2.36)/0.69 = 2.38	0.017	0.017	χ
2	HSM	(4.00-2.00)/0.69 = 2.90	0.004	0.025	✓
3	RSM	(4.00-1.64)/0.69 = 3.42	0.001	0.050	✓

TSM es el algoritmo de control



Priorización

Promedio de APMK de cada estrategia

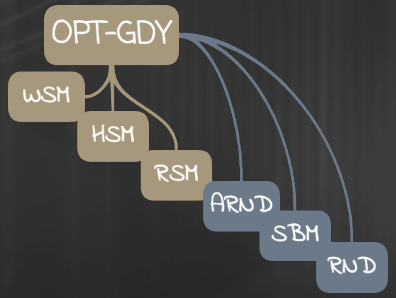
ID	Our proposal			Previous strategies			
	RSM	WSM	HSM	OPT-GDY	ARND	SBM	RND
P1	0.997	0.998	0.998	**0.999**	0.993	0.995	0.979
P2	0.998	**0.999**	0.998	**0.999**	0.994	0.997	0.992
P3	0.992	0.996	0.992	**0.997**	0.984	0.984	0.972
P4	0.994	0.998	0.995	**0.999**	0.995	0.969	0.986
P5	0.996	0.997	0.996	**0.998**	0.994	0.984	0.979
P6	0.985	0.991	0.986	**0.993**	0.976	0.877	0.949
P7	0.998	0.998	0.998	**0.999**	0.995	0.991	0.990

Strategy	Mean rank
OPT-GDY	**1.07**
WSM	2.14
HSM	3.21
RSM	3.79
ARND	5.14
SBM	5.93
RND	6.71

Resutados de las pruebas estadísticas

κ	Strategy	$z = (R_0 - R_\kappa)/SE$	ρ	α'	SD
1	WSM	(1.07-2.14)/1.16 = -0.93	0.352	0.008	χ
2	HSM	(1.07-3.21)/1.16 = -1.86	0.063	0.010	χ
3	RSM	(1.07-3.79)/1.16 = -2.35	0.019	0.013	χ
4	ARND	(1.07-5.14)/1.16 = -3.53	4E-04	0.017	✓
5	SBM	(1.07-5.93)/1.16 = -4.21	3E-05	0.025	✓
6	RND	(1.07-6.71)/1.16 = -4.89	$\ll$2E-05	0.050	✓

OPT-G es el algoritmo de control



Tiempo de Ejecución

Tiempo promedio de ejecución de cada estrategia

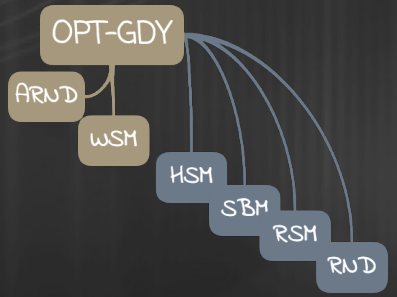
ID	Our proposal			Previous strategies			
	RSM	WSM	HSM	OPT-GDY	ARND	SBM	RND
P1	286.6	282.2	282.9	**267.7**	270.4	280.6	285.0
P2	305.4	303.9	304.4	**287.1**	287.4	295.1	305.3
P3	668.1	656.4	663.2	**639.1**	640.6	674.1	682.4
P4	140.1	137.9	138.4	**131.6**	132.1	141.6	141.3
P5	179.2	178.9	180.3	**169.6**	**169.6**	182.9	184.0
P6	133.5	131.6	133.3	**131.0**	**131.0**	139.0	140.2
P7	441.4	433.3	438.2	**411.1**	412.3	436.1	442.5

Strategy	Mean rank
OPT-GDY	**1.14**
ARND	1.86
WSM	3.29
HSM	4.57
SBM	5.00
RSM	5.57
RND	6.57

Resutados de las pruebas estadísticas

κ	Strategy	$z = (R_0 - R_\kappa)/SE$	ρ	α'	SD
1	ARND	(1.14-1.86)/1.16 = -0.62	0.535	0.008	χ
2	WSM	(1.14-3.29)/1.16 = -1.86	0.063	0.010	χ
3	HSM	(1.14-4.57)/1.16 = -2.97	0.003	0.013	✓
4	SBM	(1.14-5.00)/1.16 = -3.34	0.002	0.017	✓
5	RSM	(1.14-5.57)/1.16 = -3.84	2E-04	0.025	✓
6	RND	(1.14-6.57)/1.16 = -4.70	$\ll$2E-05	0.050	✓

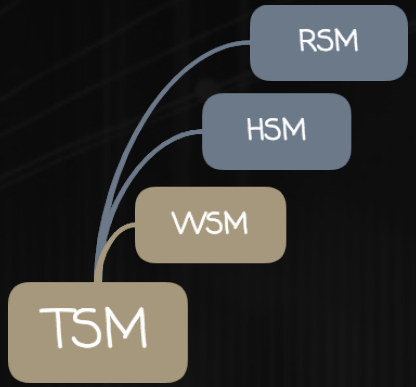
OPT-G es el algoritmo de control



Conclusiones y Contribuciones

- El RSM y el HSM generan conjuntos de prueba más pequeños que el **TSM**
- A diferencia de **TSM**, las estrategias propuestas priorizan la ejecución de los casos de prueba

ID	TSM	RSM	WSM	HSM	OPT-GDY	ARND	SBM	RND
P1	12.4	**9.0**			**9.0**			
P2	12.1	**8.0**			9.0			
P3	45.0	**40.0**			44.4			
P4	14.5	13.0			14.0			
P5	17.1	**14.0**			16.0			
P6	41.4	**39.0**			41.2			
P7	13.3	**12.0**			16.0			



RSM generó conjuntos de prueba más pequeños que **OPT-GDY**

Conclusiones y Contribuciones

- El **APMK** para todas las estrategias propuestas (RSM, WSM, HSM) es tan bueno como el de **OPT-GDY** (la mejor técnica)
- **OPT-GDY** prioriza casos de prueba que matan mutantes redundantes. Las estrategias propuestas priorizan casos de preuaba que matan mutantes no redundantes (tercos) evitando que casos de prueba con redundancia se incluyan en el conjunto

¿Preguntas?

VI CONGRESO LATINOAMERICANO DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Uso de la terquedad mutante
en el problema de priorización y minimización
de casos de prueba

Gracias por su atención

Ana Loreto Gonzalez Hernandez
loreto@gmu.edu
GEORGE MASON UNIVERSITY