ACCELERATION OF PACKET CLASSIFICATION USING ADJACENCY LIST OF RULES

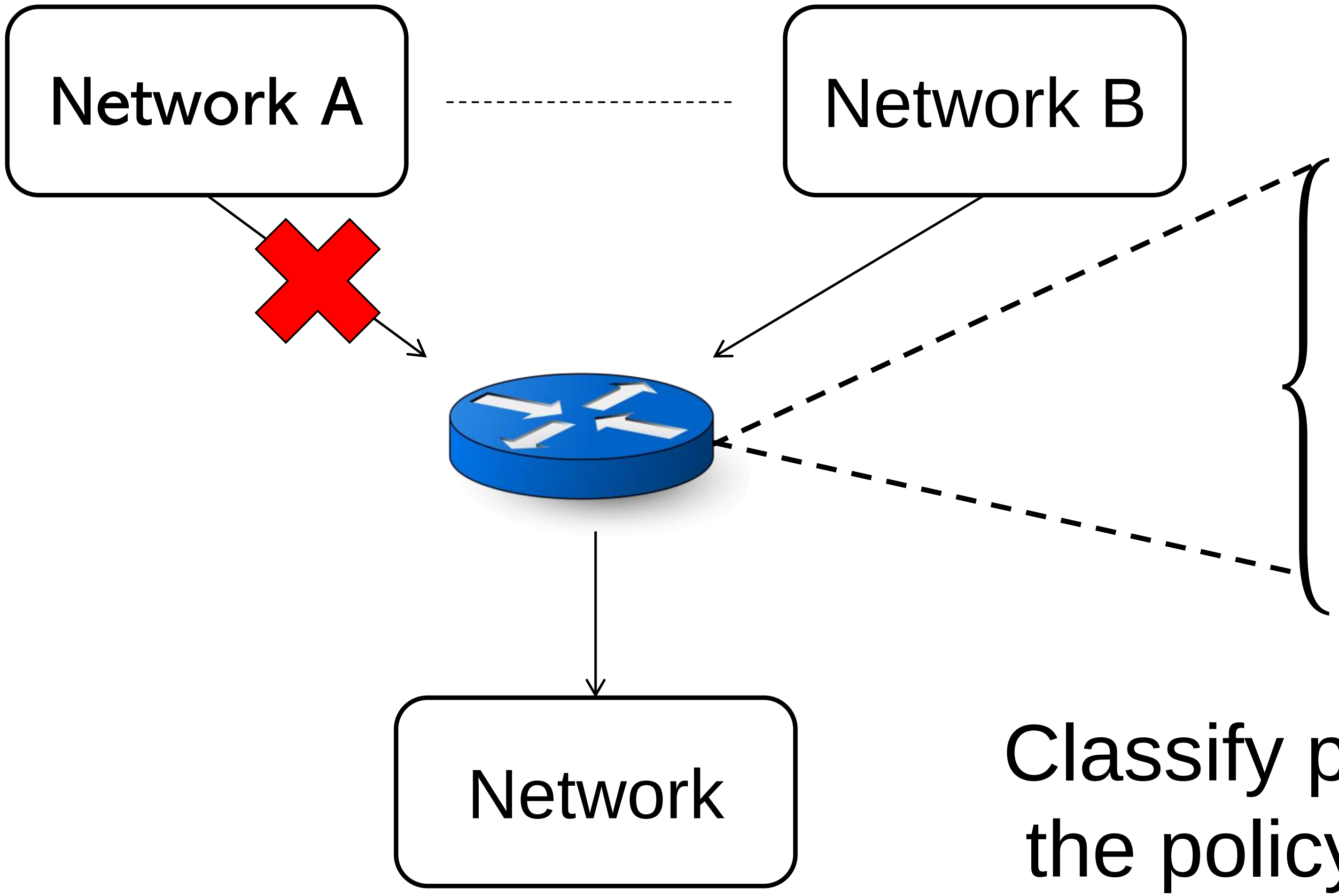
TAKASHI FUCHINO†, TAKASHI HARADA††, KEN TANAKA† AND KENJI MIKAWA†††

† Kanagawa University, Japan
†† Kochi University of Technology, Japan
††† Niigata University, Japan

TABLE OF CONTENTS

[2]A. Tapdiya and E. Fulp, "Towards optimal firewall rule ordering utilizing directed acyclical graphs," in Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on, Aug 2009, pp. 1–6.

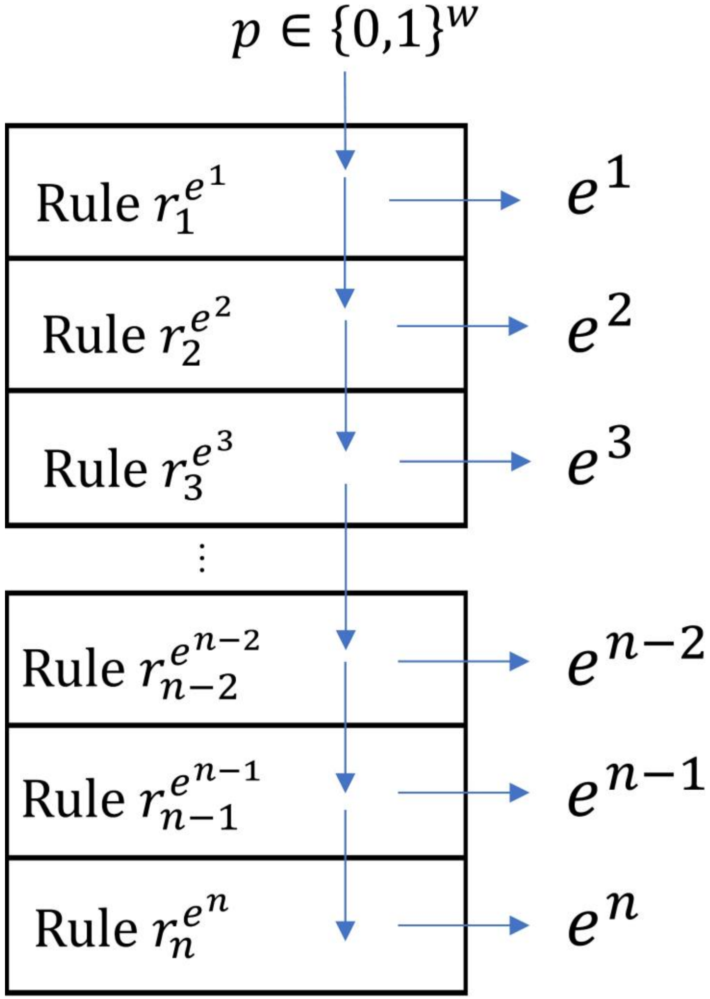
PACKET CLASSIFICATION



$$
\begin{cases}
Network\ \mathrm{A} \mapsto Drop \\
Network\ \mathrm{B} \mapsto Network\mathrm{C} \\
Network\ \mathrm{D} \mapsto Network\mathrm{R} \\
\vdots \\
Network\ \mathrm{Z} \mapsto Drop
\end{cases}
$$

Classify packets according to the policy

MODEL OF PACKET CLASSIFICATION



e.g. $w = 5, p = 11000$

$$r_2^{e^2} = * 1 * 00$$

Make a rule list according to the policy

Each packet is compared with each rule in order.

Assign the evaluation type of the first matched rule.

PACKET CLASSIFICATION ON A RULE LIST

e.g.
$e \in \{P, D\}$
$p = 01000$,
P is assigned to p.

$R(p)$ is denoted an evaluation type for p as the classification result.
$R(01000) = P$

Classifier R
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_4^D = 0 1 0 1 *$
$r_5^D = 0 1 1 1 *$
$r_6^P = 0 1 * * *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$

CLASSIFICATION LATENCY $L(R_\sigma, F)$

Regard a comparison of a packet with a rule as the latency 1

$$L(R_\sigma, F) \equiv \sum_{i=1}^{n-1} i|E(R_\sigma, \sigma^{-1}(i))|_F + (n-1)|E(R_\sigma, \sigma^{-1}(n))|_F$$

where, R is a rule list, F is a packet arrival distribution and σ is an order of rules.

CLASSIFICATION LATENCY $L(R_\sigma, F)$

00000 ↦10	00001 ↦50	00010 ↦17	00011 ↦23
00100 ↦20	00101 ↦60	00110 ↦8	00111 ↦8
01000 ↦200	01001 ↦5	01010 ↦20	01011 ↦35
01100 ↦200	01101 ↦27	01110 ↦15	01111 ↦40
10000 ↦8	10001 ↦2	10010 ↦12	10011 ↦13
10100 ↦6	10101 ↦2	10110 ↦12	10111 ↦28
11000 ↦1	11001 ↦13	11010 ↦2	11011 ↦1
11100 ↦3	11101 ↦3	11110 ↦7	11111 ↦2

Classifier R	$\|E(R,i)\|_F$
$r_1^P = 0 * 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0 *$	60
$r_3^D = 0 * * 0\ 1$	5
$r_4^D = 0\ 1\ 0\ 1 *$	55
$r_5^D = 0\ 1\ 1\ 1 *$	55
$r_6^P = 0\ 1 * * *$	400
$r_7^P = 0\ 0 * * *$	60
$r_8^P = 1\ 0 * 1 *$	65
$r_9^D = * * * * *$	50

$$L(R,F) = 1 \cdot 87 + 2 \cdot 60 + 3 \cdot 5 + 4 \cdot 55 + 5 \cdot 55 + 6 \cdot 400 + 7 \cdot 60 + 8 \cdot 65 + 8 \cdot 50 = 4684$$

POLICY AND REORDERING RULES

Classifier R	$\|E(R,i)\|_F$
$r_1^P = 0 * 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0 *$	60
$r_3^D = 0 * * 0\ 1$	5
$r_4^D = 0\ 1\ 0\ 1 *$	55
$r_5^D = 0\ 1\ 1\ 1 *$	55
$r_6^P = 0\ 1 * * *$	400
$r_7^P = 0\ 0 * * *$	60
$r_8^P = 1\ 0 * 1 *$	65
$r_9^D = * * * * *$	50
$L(R,F) = 4684$	

Classifier R_σ	$\|E(R_\sigma,i)\|_F$
$r_4^D = 0\ 1\ 0\ 1 *$	55
$r_1^P = 0 * 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0 *$	60
$r_3^D = 0 * * 0\ 1$	5
$r_5^D = 0\ 1\ 1\ 1 *$	55
$r_6^P = 0\ 1 * * *$	400
$r_8^P = 1\ 0 * 1 *$	65
$r_7^P = 0\ 0 * * *$	60
$r_9^D = * * * * *$	50
$L(R_\sigma,F) = 4439$	

R and R_σ denote the same policy

POLICY VIOLATION

$R(01010) = R(01011) = D$

If r_4 and r_6 interchange,
$R(01010) = R(01011) = P$

Policy violation occurs

Classifier R
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_4^D = 0 1 0 1 *$
$r_5^D = 0 1 1 1 *$
$r_6^P = 0 1 * * *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$

Classifier R
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_6^P = 0 1 * * *$
$r_5^D = 0 1 1 1 *$
$r_4^D = 0 1 0 1 *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$

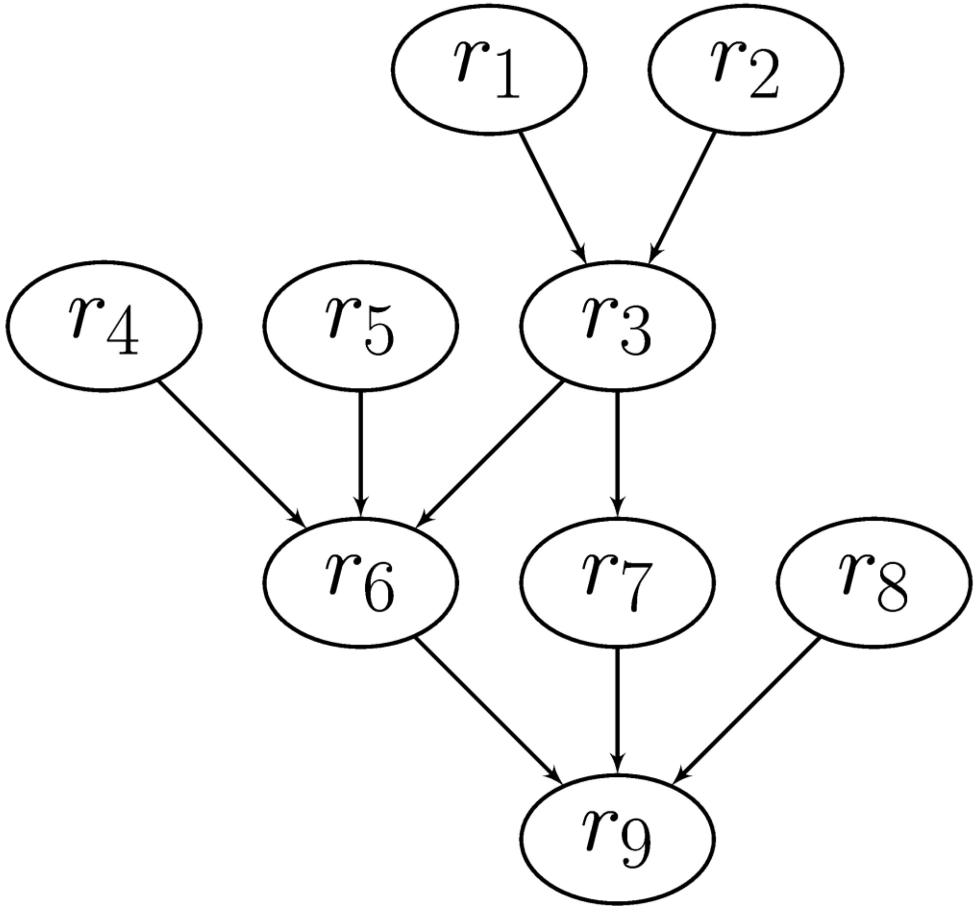
OPTIMAL RULE ORDERING

Optimal Rule Ordering (ORO)	
Input	Rule list R and packet arrival distribution F
Output	Order of rules σ that minimizes $L(R_\sigma, F)$ s.t. σ hold the classification policy.

We need to know which pair of rules causes policy violation when interchanging.

DEPENDENT GRAPH

Classifier *R*
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_4^D = 0 1 0 1 *$
$r_5^D = 0 1 1 1 *$
$r_6^P = 0 1 * * *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$



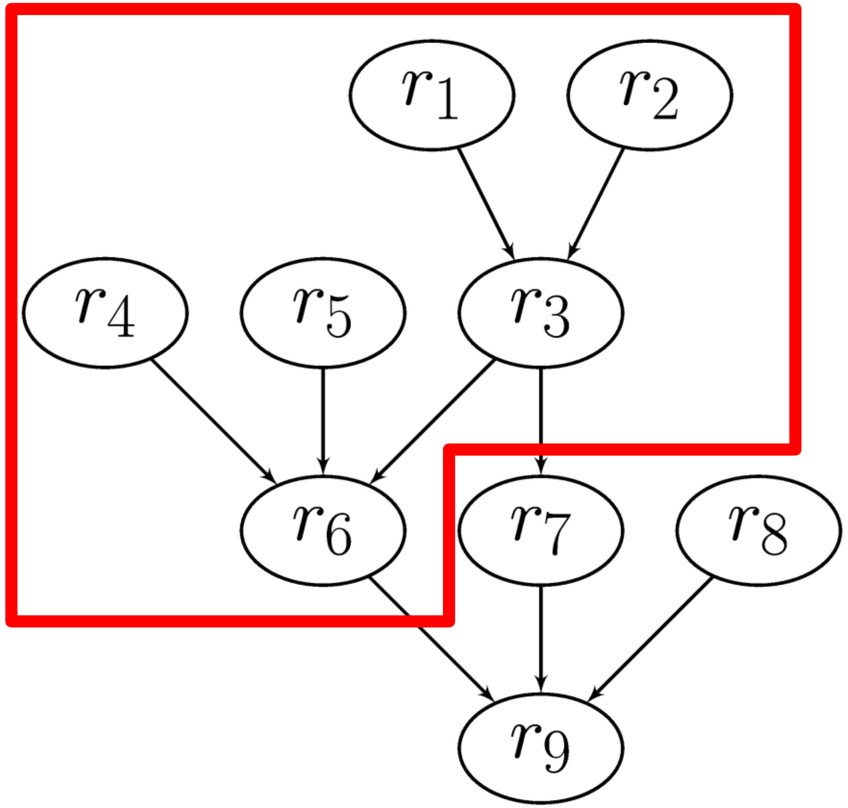
Dependent graph

Draw arrows from the precedent rule to the dependent rules.

To hold the policy, it is necessary to keep the order of the arrows.

SUB GRAPH MERGING(SGM)[2]

Classifier R
$r_1^P = 0 * 1\,0\,1$
$r_2^P = 0\,0\,0\,0\,*$
$r_3^D = 0 * * 0\,1$
$r_4^D = 0\,1\,0\,1\,*$
$r_5^D = 0\,1\,1\,1\,*$
$r_6^P = 0\,1 * * *$
$r_7^P = 0\,0 * * *$
$r_8^P = 1\,0 * 1\,*$
$r_9^D = * * * * *$



$G(r_i)$ is the reachable rule set from rule i and the rule i

$G(r_1) = \{r_1\}$
$G(r_2) = \{r_2\}$
$G(r_3) = \{r_3\}$
$G(r_4) = \{r_4\}$
$G(r_5) = \{r_5\}$
$G(r_6) = \{r_1, r_2, r_3, r_4, r_5, r_6\}$
$G(r_7) = \{r_1, r_2, r_3, r_7\}$
$G(r_8) = \{r_8\}$
$G(r_9) = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9\}$

[2]A. Tapdiya and E. Fulp, "Towards optimal firewall rule ordering utilizing directed acyclical graphs," in Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on, Aug 2009, pp. 1–6.

COMPUTE THE AVERAGE OF WEIGHTS

Classifier R	$\lvert E(R,i)\rvert_F$
$r_1^P = 0 * 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0\ *$	60
$r_3^D = 0 * * 0\ 1$	5
$r_4^D = 0\ 1\ 0\ 1\ *$	55
$r_5^D = 0\ 1\ 1\ 1\ *$	55
$r_6^P = 0\ 1 * * *$	400
$r_7^P = 0\ 0 * * *$	60
$r_8^P = 1\ 0 * 1 *$	65
$r_9^D = * * * * *$	50

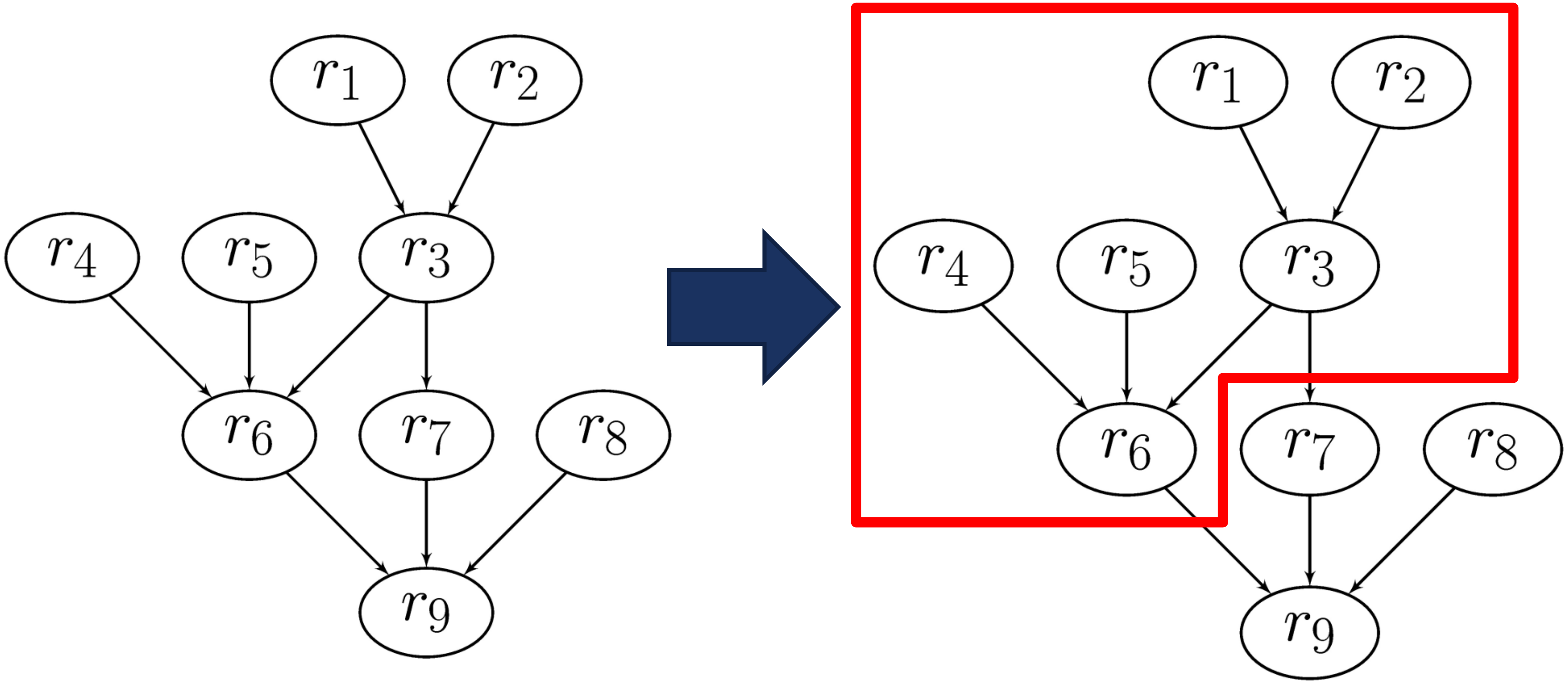
$$G(r_6) = \{r_1, r_2, r_3, r_4, r_5, r_6\}$$

$$|G(r_6)| = \frac{|r_1| + |r_2| + |r_3| + |r_4| + |r_5| + |r_6|}{6}$$

$$= \frac{87+60+5+55+55+400}{6}$$

$$= 110.33$$

COMPARE RULES



$|G(r_1)| = 87$

$|G(r_2)| = 60$

$|G(r_3)| = 50.66$

$|G(r_4)| = 55$

$|G(r_5)| = 55$

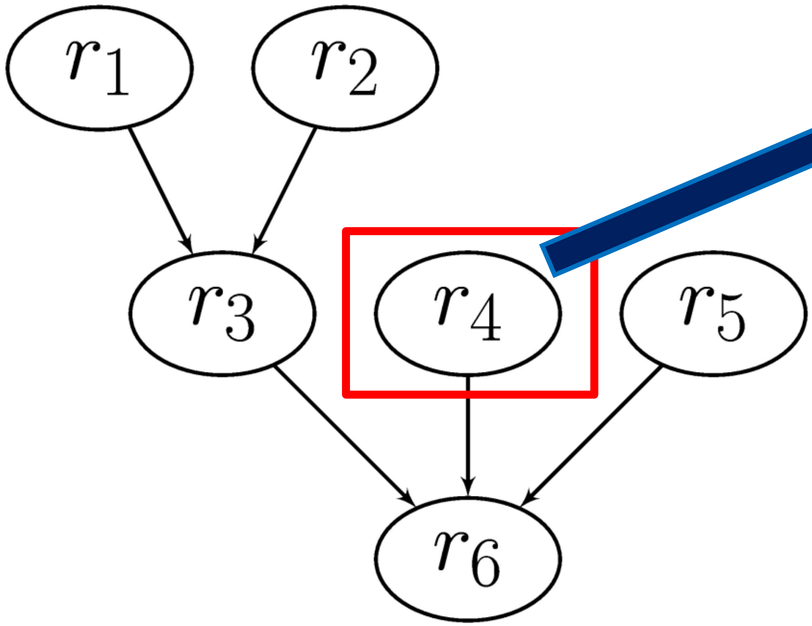
$|G(r_6)| = 110.33$

$|G(r_7)| = 53$

$|G(r_8)| = 65$

$|G(r_9)| = 93$

ADD TO SORTED LIST



Sorted List
r_4

$|G(r_3)| = 50.66$

$|G(r_4)| = 55$

$|G(r_5)| = 55$

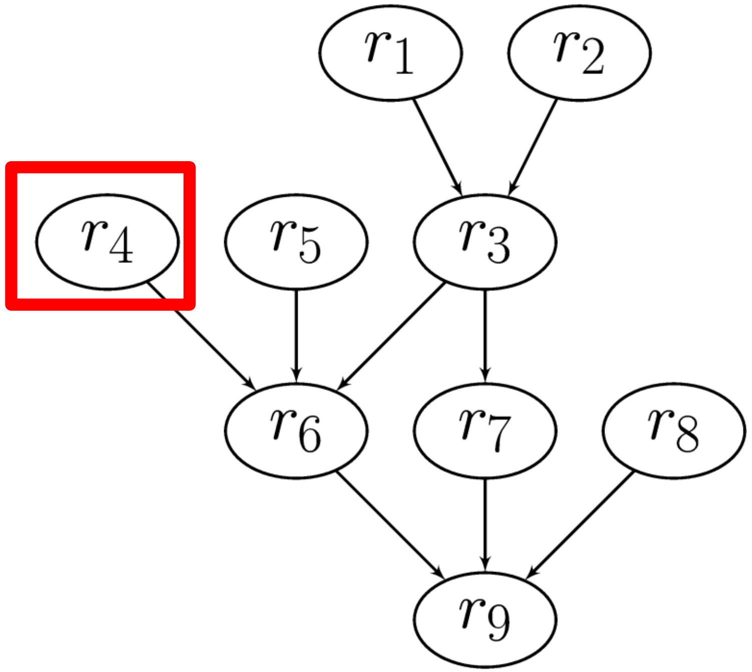
RESULT OF REORDER BY SGM

Sorted List
r_4
r_5
r_1
r_2
r_3
r_6
r_8
r_7
r_9

Sorted	$\lvert E(R,i)\rvert_F$
$r_4^D = 0\ 1\ 0\ 1\ *$	55
$r_5^D = 0\ 1\ 1\ 1\ *$	55
$r_1^P = 0\ *\ 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0\ *$	60
$r_3^D = 0\ *\ *\ 0\ 1$	5
$r_6^P = 0\ 1\ *\ *\ *$	400
$r_8^P = 1\ 0\ *\ 1\ *$	65
$r_7^P = 0\ 0\ *\ *\ *$	60
$r_9^D = *\ *\ *\ *\ *$	50
$L(R,F) = 4457$	

Classifier R	$\lvert E(R,i)\rvert_F$
$r_1^P = 0\ *\ 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0\ *$	60
$r_3^D = 0\ *\ *\ 0\ 1$	5
$r_4^D = 0\ 1\ 0\ 1\ *$	55
$r_5^D = 0\ 1\ 1\ 1\ *$	55
$r_6^P = 0\ 1\ *\ *\ *$	400
$r_7^P = 0\ 0\ *\ *\ *$	60
$r_8^P = 1\ 0\ *\ 1\ *$	65
$r_9^D = *\ *\ *\ *\ *$	50
$L(R,F) = 4684$	

SGM[2] FALLS INTO INFINITE LOOPS

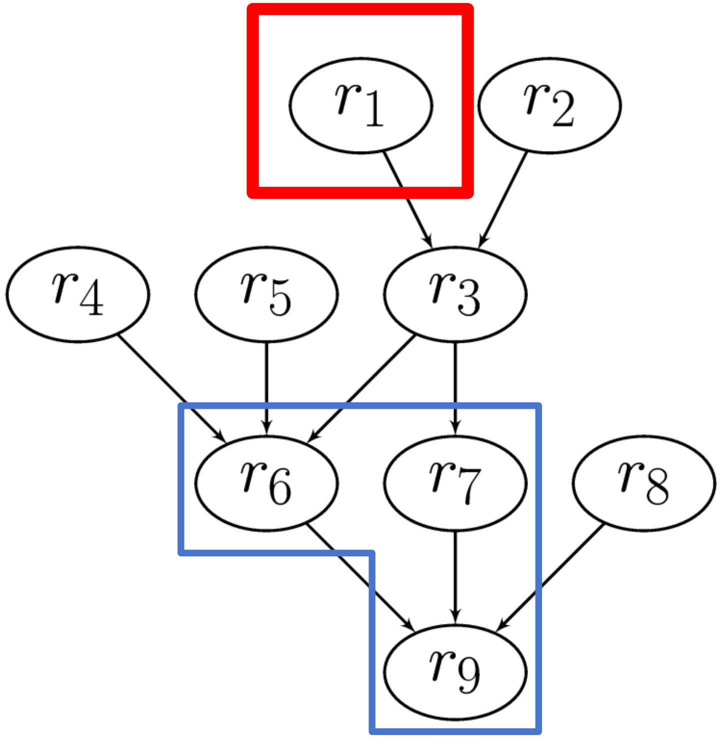


SGM keeps comparing rules until the rule set is singleton. So, the algorithm needs the information on how many rules should be preceded in each rule.
And after the rule are deleted in the rule list, the count has to be updated.

But, SGM[2] can't update them exactly. So, the algorithm often falls into infinite loop.

[2]A. Tapdiya and E. Fulp, "Towards optimal firewall rule ordering utilizing directed acyclical graphs," in Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on, Aug 2009, pp. 1–6.

FIX THE SGM IN [2]



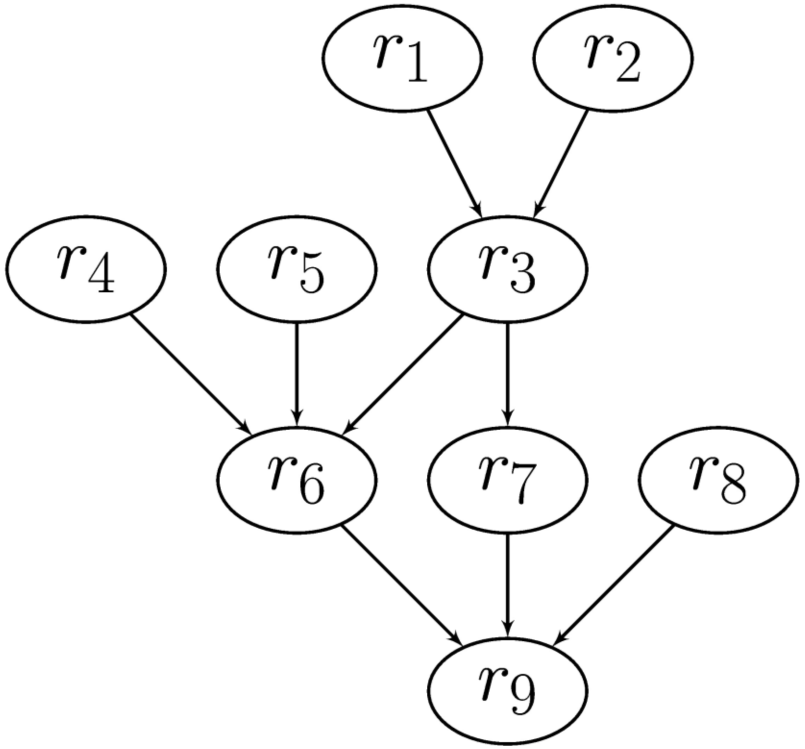
$C[1] = 1$
$C[2] = 1$
$C[3] = 3$
$C[4] = 1$
$C[5] = 1$
$C[6] = 6$
$C[7] = 4$
$C[8] = 1$
$C[9] = 9$

→

$C[1] = 0$	$C[1] = 0$
$C[2] = 1$	$C[2] = 1$
$C[3] = 2$	$C[3] = 2$
$C[4] = 1$	$C[4] = 1$
$C[5] = 1$	$C[5] = 1$
$C[6] = 6$	$C[6] = 5$
$C[7] = 4$	$C[7] = 3$
$C[8] = 1$	$C[8] = 1$
$C[9] = 9$	$C[9] = 8$

[2]A. Tapdiya and E. Fulp, "Towards optimal firewall rule ordering utilizing directed acyclical graphs," in Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on, Aug 2009, pp. 1–6.

SGM[2] USING THE ADJACENCY LIST



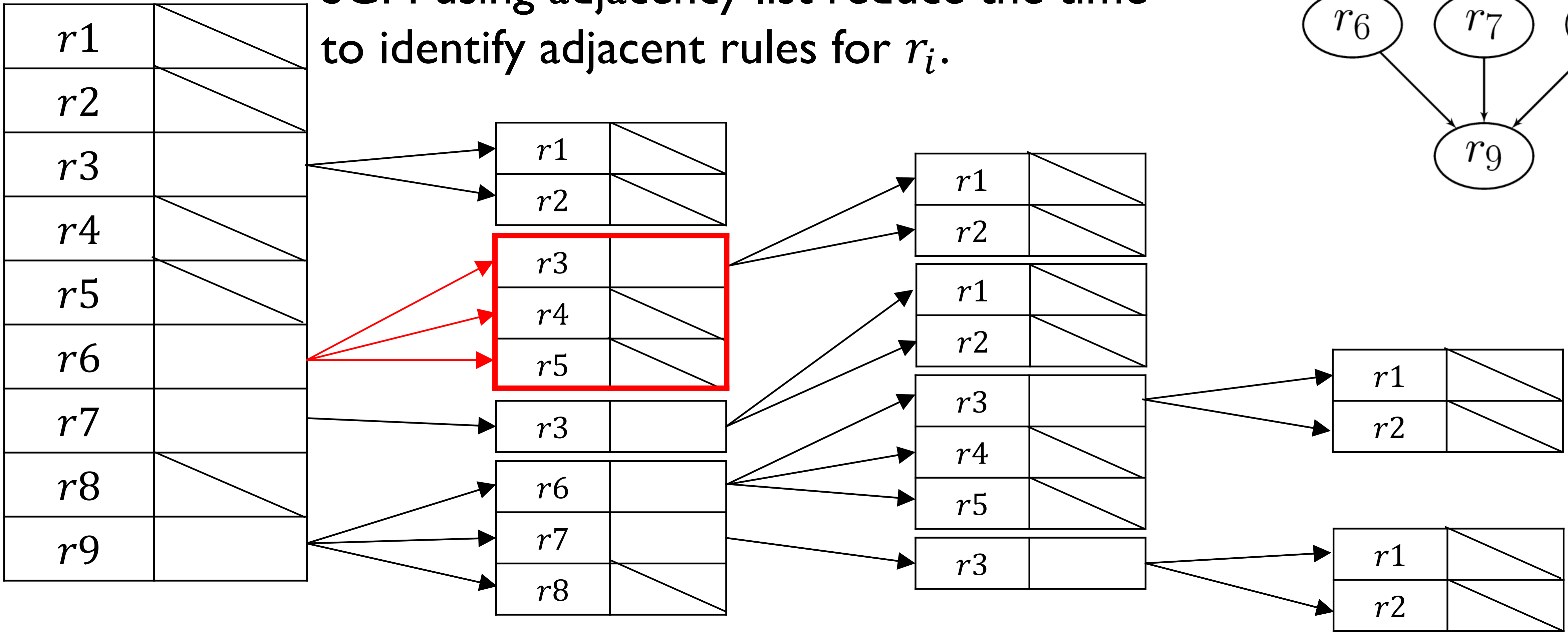
The preceding relation
with two-dimensional array: DEP[][]

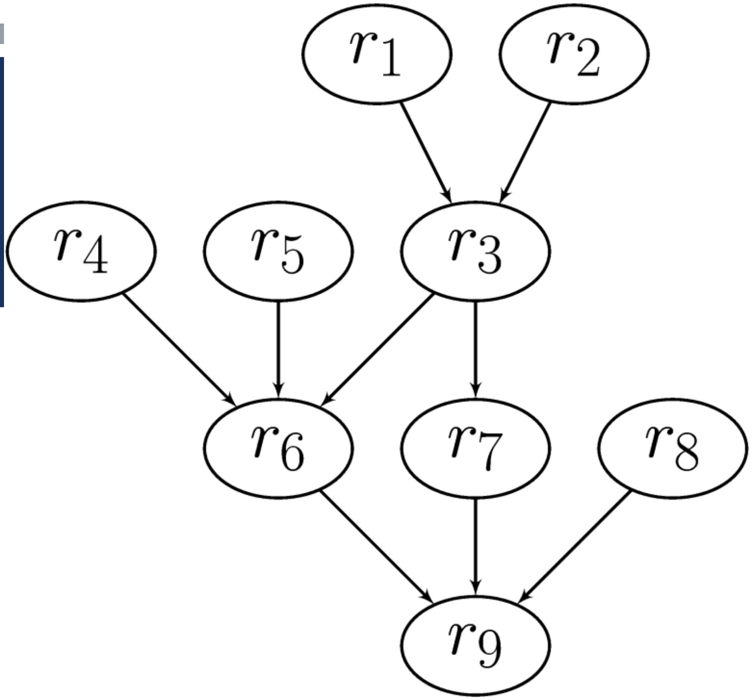
```
   1 2 3 4 5 6 7 8 9
1 {0,0,0,0,0,0,0,0,0,0,0},
2 {0,0,0,0,0,0,0,0,0,0,0},
3 {0,1,1,0,0,0,0,0,0,0,0},
4 {0,0,0,0,0,0,0,0,0,0,0},
5 {0,0,0,0,0,0,0,0,0,0,0},
6 {0,0,0,1,1,1,0,0,0,0,0},
7 {0,0,0,1,0,0,0,0,0,0,0},
8 {0,0,0,0,0,0,0,0,0,0,0},
9 {0,0,0,0,0,0,1,1,1,0,0}
```

Identifying adjacent rules for ri with DEP[][] requires at most n steps.

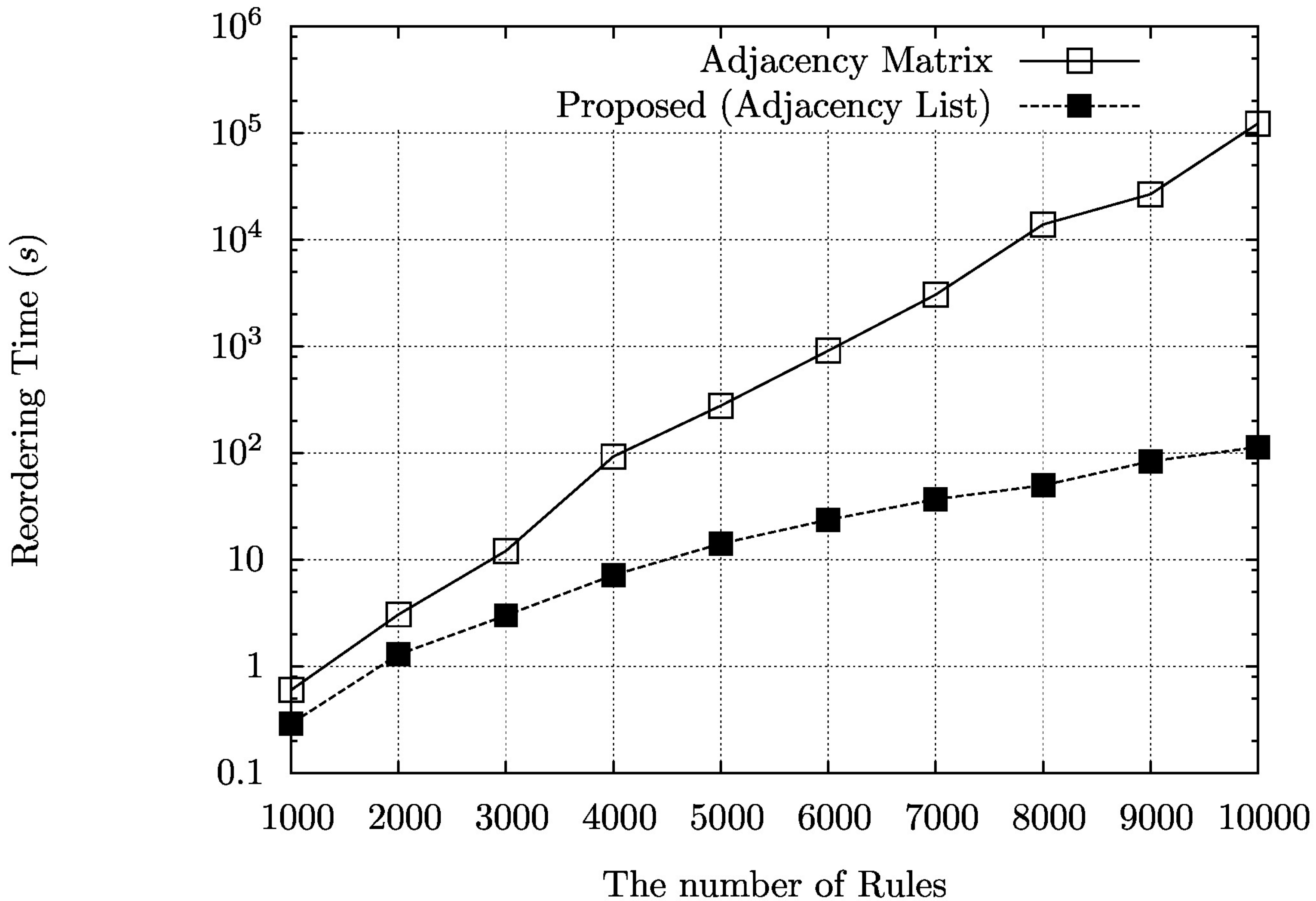
ADJACENCY LIST

SGM using adjacency list reduce the time to identify adjacent rules for r_i.





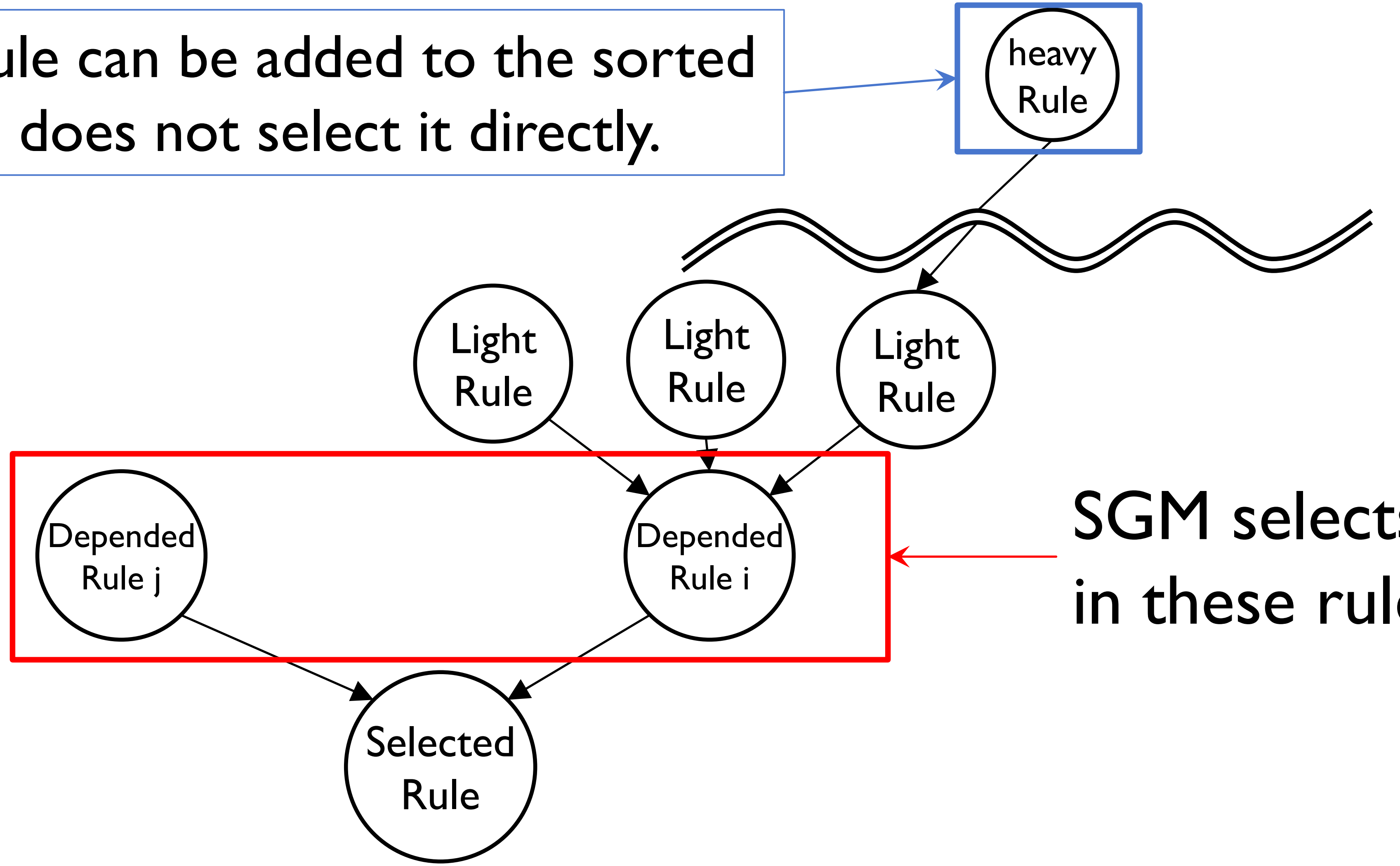
RESULT OF EXPERIMENT ON REORDERING TIME(ACL)



THE METHOD OF RULE SELECTION IN SGM

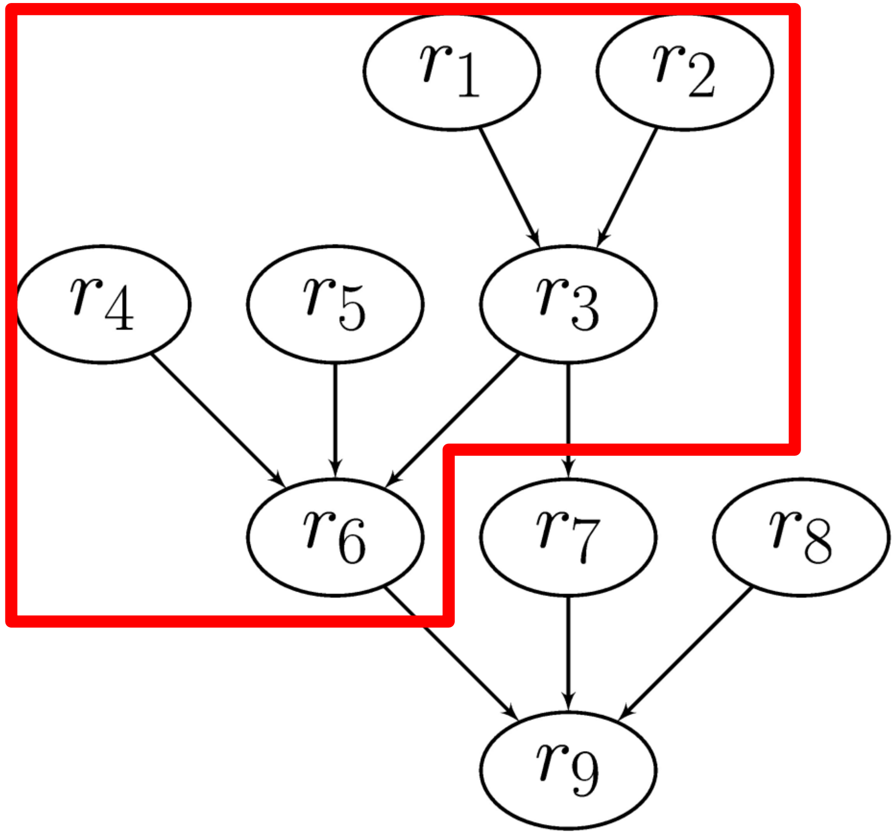
The heavy rule can be added to the sorted list but SGM does not select it directly.

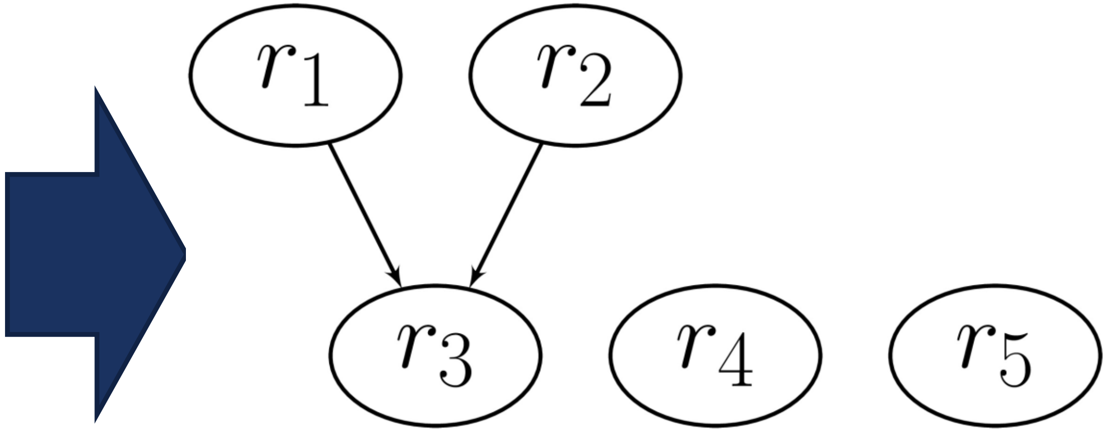
SGM selects some rule in these rules



SGM CAN NOT SELECT r_1

R	$\|E(R,i)\|_F$
r_1	87
r_2	60
r_3	5
r_4	55
r_5	55
r_6	400
r_7	60
r_8	65
r_9	50



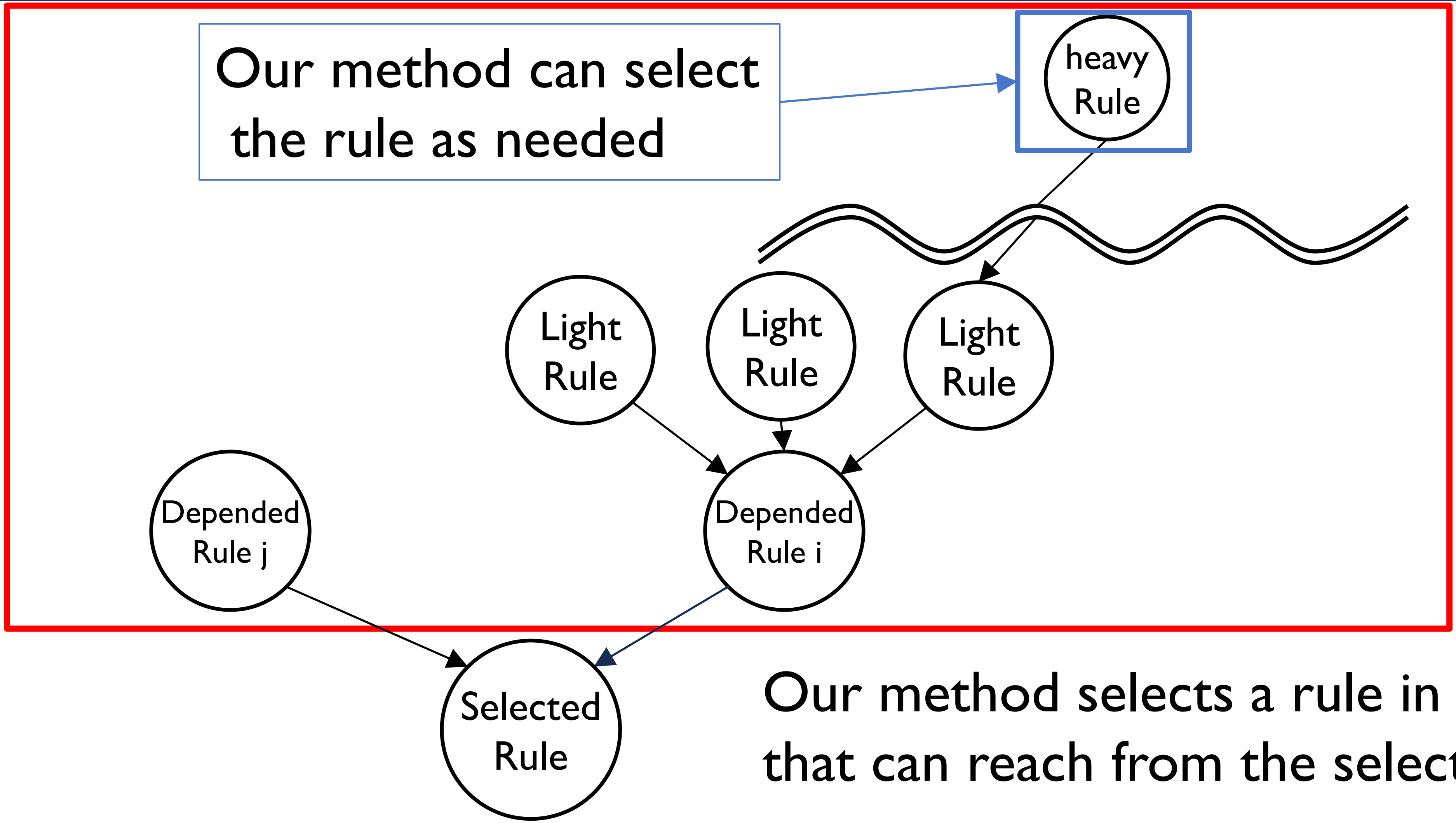


$G(r_3) = 50.66$

$G(r_4) = 55$

$G(r_5) = 55$

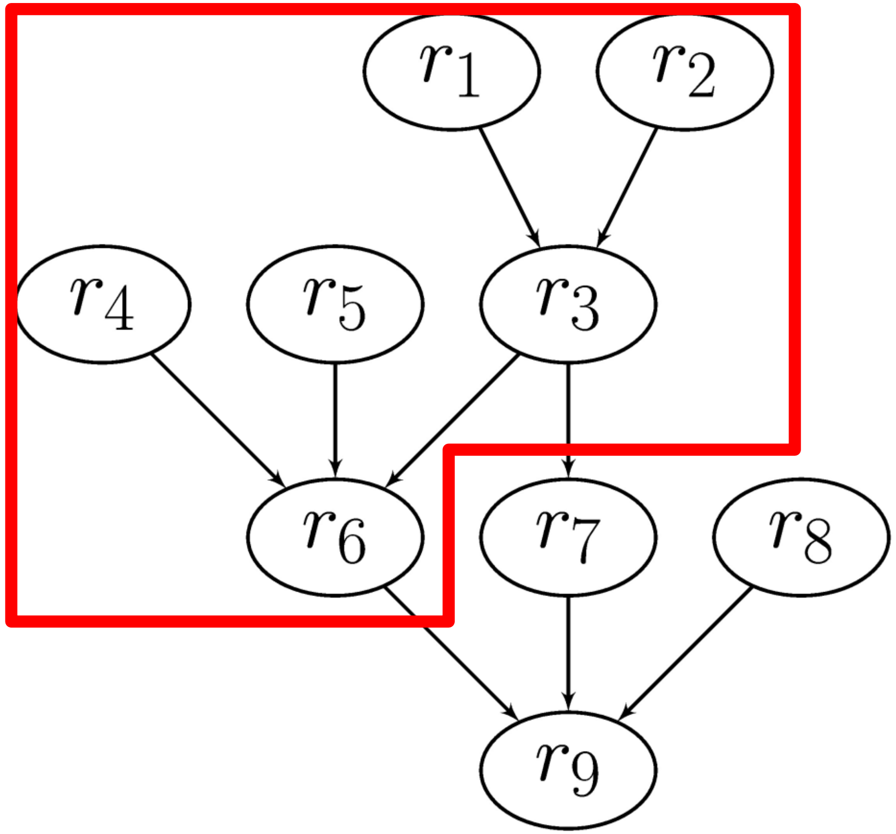
COMPREHENSIVE CONSTRUCTION OF SUB-GRAPHS

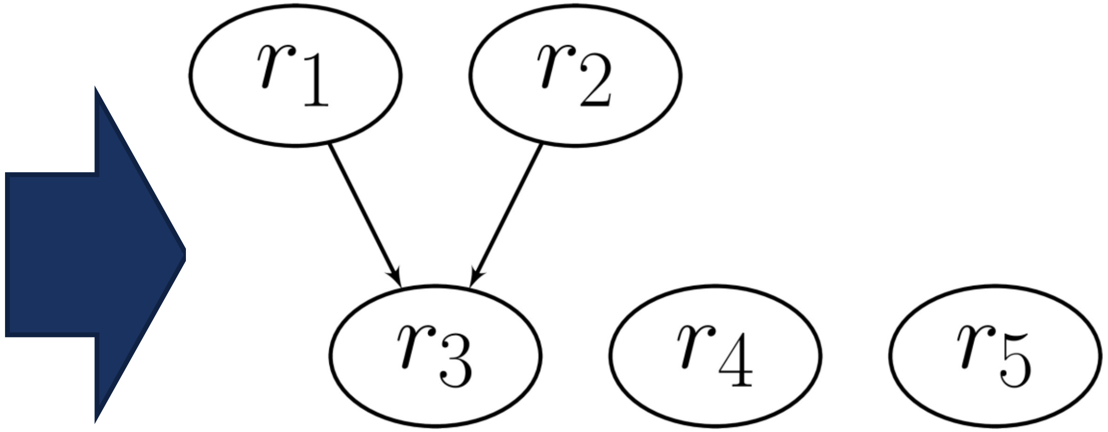


Our method selects a rule in all of rules that can reach from the selected rule

SELECT THE HEAVIEST RULE IN LOCATABLE RULES

R	$\|E(R,i)\|_F$
r_1	87
r_2	60
r_3	5
r_4	55
r_5	55
r_6	400
r_7	60
r_8	65
r_9	50





$G(r_1) = 87$
$G(r_2) = 60$

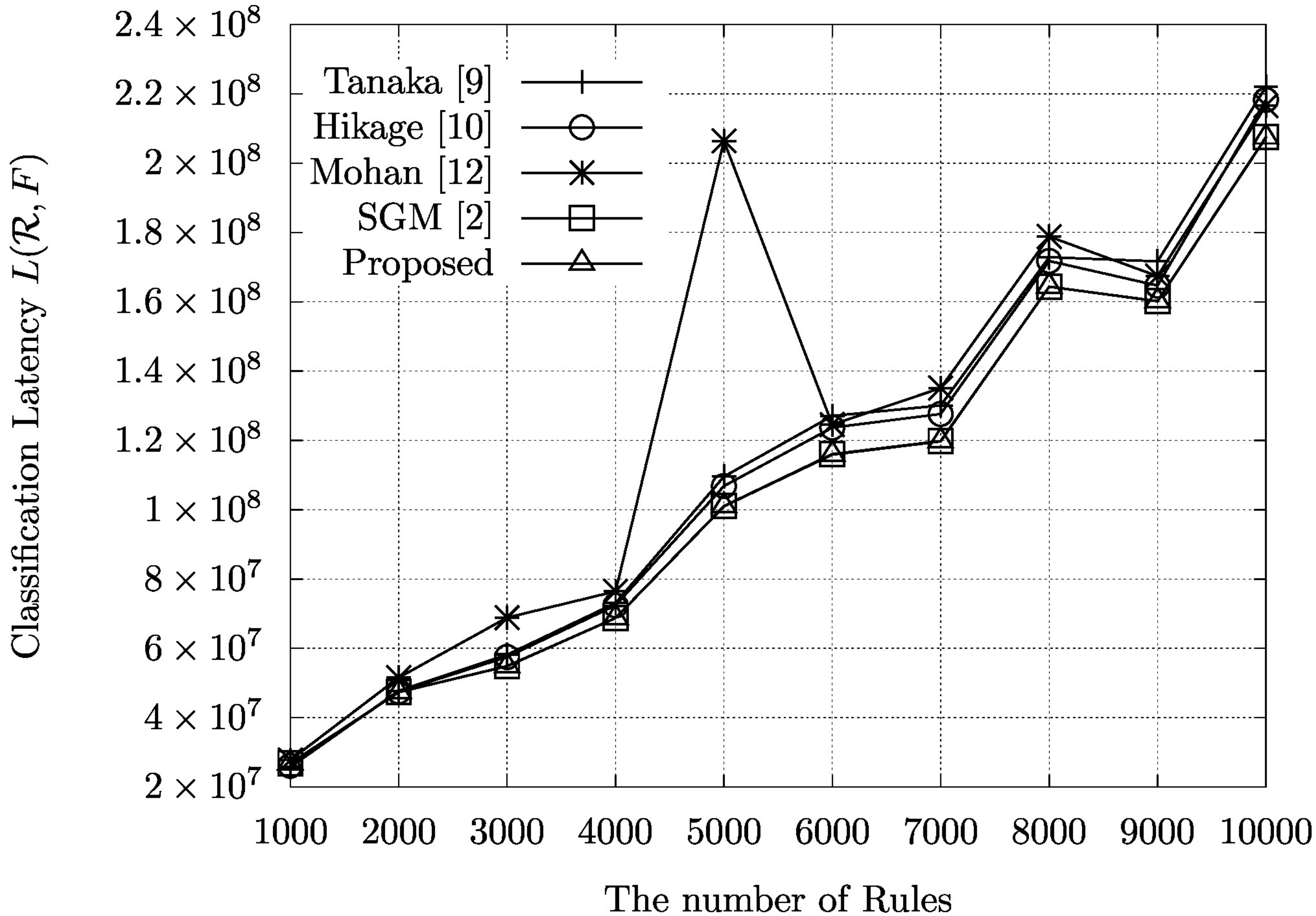
$G(r_3) = 50.66$
$G(r_4) = 55$
$G(r_5) = 55$

THE REORDERING RESULT OF CLASSIFIER R

Proposed	$\lvert E(R,i)\rvert_F$
$r_1^P = 0 * 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0 *$	60
$r_4^D = 0\ 1\ 0\ 1 *$	55
$r_5^D = 0\ 1\ 1\ 1 *$	55
$r_3^D = 0 * * 0\ 1$	5
$r_6^P = 0\ 1 * * *$	400
$r_8^P = 1\ 0 * 1 *$	65
$r_7^P = 0\ 0 * * *$	60
$r_9^D = * * * * *$	50
$L(R,F) = 4349$	

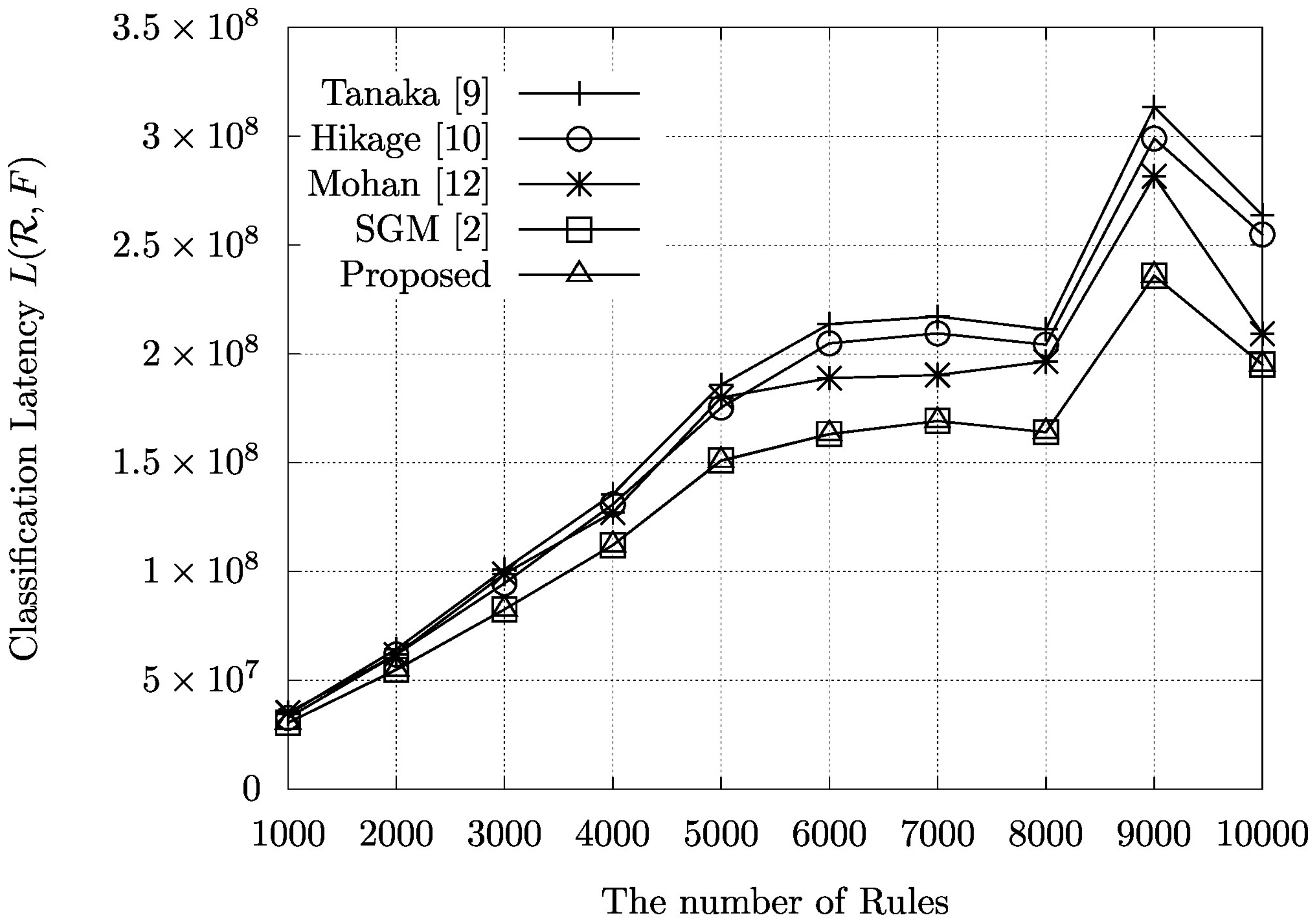
SGM	$\lvert E(R,i)\rvert_F$
$r_4^D = 0\ 1\ 0\ 1 *$	55
$r_5^D = 0\ 1\ 1\ 1 *$	55
$r_1^P = 0 * 1\ 0\ 1$	87
$r_2^P = 0\ 0\ 0\ 0 *$	60
$r_3^D = 0 * * 0\ 1$	5
$r_6^P = 0\ 1 * * *$	400
$r_8^P = 1\ 0 * 1 *$	65
$r_7^P = 0\ 0 * * *$	60
$r_9^D = * * * * *$	50
$L(R,F) = 4457$	

RESULT OF EXPERIMENT WITH THE PROPOSED(ACL)



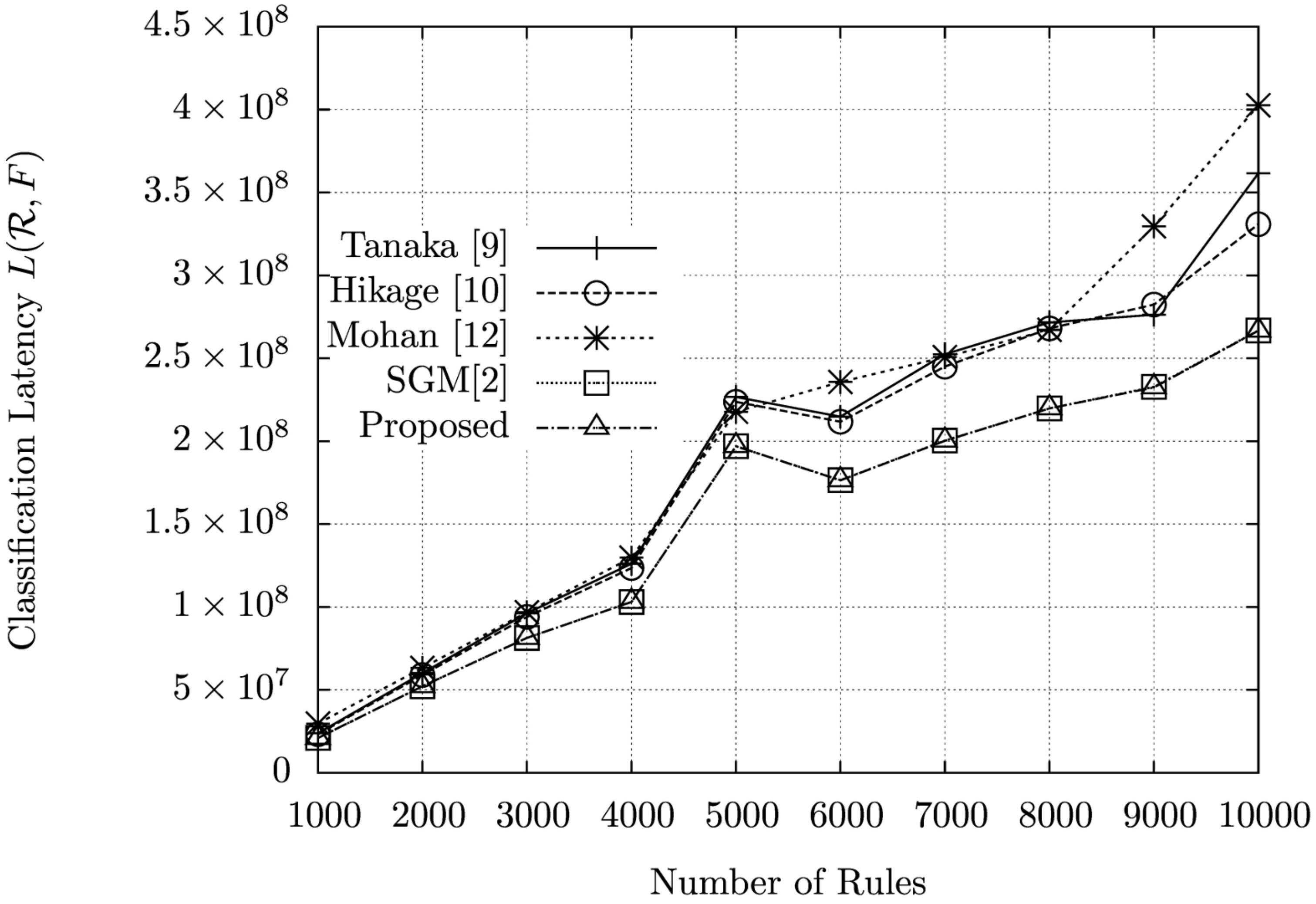
	SGM	Proposed
1000	2.68005×10^7	2.67989×10^7
2000	4.73388×10^7	4.73274×10^7
3000	5.48907×10^7	5.48412×10^7
4000	6.87417×10^7	6.87130×10^7
5000	1.01037×10^8	1.00985×10^8
6000	1.16075×10^8	1.15903×10^8
7000	1.19821×10^8	1.19704×10^8
8000	1.64378×10^8	1.64333×10^8
9000	1.60184×10^8	1.60136×10^8
10000	2.07544×10^8	2.07395×10^8

RESULT OF EXPERIMENT WITH THE PROPOSED(IPC)



	SGM	Proposed
1000	3.0485×10^7	3.0465×10^7
2000	5.4826×10^7	5.4834×10^7
3000	8.2575×10^7	8.2603×10^7
4000	1.1210×10^8	1.1212×10^8
5000	1.5103×10^8	1.5078×10^8
6000	1.6314×10^8	1.6301×10^8
7000	1.6911×10^8	1.6912×10^8
8000	1.6398×10^8	1.6402×10^8
9000	2.3594×10^8	2.3592×10^8
10000	1.9509×10^8	1.9505×10^8

RESULT OF EXPERIMENT WITH THE PROPOSED(FW)



	SGM	Proposed
1000	2.10699×10^7	2.10647×10^7
2000	5.2163×10^7	5.21712×10^7
3000	8.13315×10^7	8.12916×10^7
4000	1.0304×10^8	1.03066×10^8
5000	1.96935×10^8	1.9692×10^8
6000	1.76441×10^8	1.7633×10^8
7000	2.00270×10^8	2.0024×10^8
8000	2.19776×10^8	2.19732×10^8
9000	2.32778×10^8	2.32405×10^8
10000	2.66782×10^8	2.66643×10^8

CONCLUSION AND FUTURE WORK

Conclusion

- Introduced Optimal Rule Ordering Problem
- Fixed SGM in [2]
- Applied the Adjacency List, and showed that the proposed method decreases reordering time compared with original SGM
- Proposed augmented SGM with Comprehensive Construction of Sub-graphs
- The results of our experiments show the proposed method reduces the latency compared to other method

Future Work

- Developing a reordering method in consideration of weight variation
- Reducing reordering time of SGM

SUB SEAT

FORM OF PACKET AND RULE

Packet as a bit string of length w.

e.g. $w = 5,\ \ p = 11000$

Condition of rule as a string on $\{0,1,*\}^w$.

$$r_i^e = b_1 b_2 \cdots b_w \quad \begin{pmatrix} b_k \in \{0,1,*\}, \\ e \in \{A_1, A_2, \ldots, A_m\} \end{pmatrix}$$

e.g. $w = 5,\qquad r_2^{e_2} = *\,1*00$

PACKET CLASSIFICATION

00000 $\mapsto P$	00001 $\mapsto P$	00010 $\mapsto P$	00011 $\mapsto P$
00100 $\mapsto P$	00101 $\mapsto P$	00110 $\mapsto P$	00111 $\mapsto P$
01000 $\mapsto D$	01001 $\mapsto D$	01010 $\mapsto D$	01011 $\mapsto D$
01100 $\mapsto D$	01101 $\mapsto D$	01110 $\mapsto D$	01111 $\mapsto D$
10000 $\mapsto D$	10001 $\mapsto D$	10010 $\mapsto P$	10011 $\mapsto P$
10100 $\mapsto D$	10101 $\mapsto D$	10110 $\mapsto P$	10111 $\mapsto P$
11000 $\mapsto D$	11001 $\mapsto D$	11010 $\mapsto D$	11011 $\mapsto D$
11100 $\mapsto D$	11101 $\mapsto D$	11110 $\mapsto D$	11111 $\mapsto D$

Classifier R
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_4^D = 0 1 0 1 *$
$r_5^D = 0 1 1 1 *$
$r_6^P = 0 1 * * *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$

The table on the right shows the policy on the left.

$|P|_F$

Let P be a set of packets and F be a packet arrival distribution.

$$|P|_F \equiv \sum_{p \in P} F(P)$$

$00000 \mapsto 10$	$00001 \mapsto 50$	$00010 \mapsto 17$	$00011 \mapsto 23$
$00100 \mapsto 20$	$00101 \mapsto 60$	$00110 \mapsto 8$	$00111 \mapsto 8$
$01000 \mapsto 200$	$01001 \mapsto 5$	$01010 \mapsto 20$	$01011 \mapsto 35$
$01100 \mapsto 200$	$01101 \mapsto 27$	$01110 \mapsto 15$	$01111 \mapsto 40$
$10000 \mapsto 8$	$10001 \mapsto 2$	$10010 \mapsto 12$	$10011 \mapsto 13$
$10100 \mapsto 6$	$10101 \mapsto 2$	$10110 \mapsto 12$	$10111 \mapsto 28$
$11000 \mapsto 1$	$11001 \mapsto 13$	$11010 \mapsto 2$	$11011 \mapsto 1$
$11100 \mapsto 3$	$11101 \mapsto 3$	$11110 \mapsto 7$	$11111 \mapsto 2$

e.g.

$$P = \{00011, 01101\}$$

$$|P| = 23 + 27 = 50$$

OVERLAP RELATION

If there is a packet p that matches both r_i and r_j, r_i and r_j are said to be overlapped.

e.g. Because, there is packet 01101 that matches r_1^P and r_6^P are overlapped

Classifier R
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_4^D = 0 1 0 1 *$
$r_5^D = 0 1 1 1 *$
$r_6^P = 0 1 * * *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$

DEPENDENCY RELATION

If r_i^e and r_j^f are overlapped and e is different from f, r_i^e and r_j^f are said to be dependent.

e.g.

Because, r_2^P and r_3^D are overlapped and those actions are different, r_2^P and r_3^D are dependent.

Interchanging r_2^P and r_3^D cause policy violation.

Classifier R
$r_1^P = 0 * 1 0 1$
$r_2^P = 0 0 0 0 *$
$r_3^D = 0 * * 0 1$
$r_4^D = 0 1 0 1 *$
$r_5^D = 0 1 1 1 *$
$r_6^P = 0 1 * * *$
$r_7^P = 0 0 * * *$
$r_8^P = 1 0 * 1 *$
$r_9^D = * * * * *$

DEPENDENCY RELATION

e.g.

r_2^P and r_3^D are overlapped because there are packets 00001,00101,01001,01101 that match both rules and those actions are different, so r_2^P and r_3^D are dependent.

→ Interchanging r_2^P and r_3^D causes policy violation.

Classifier R
$r_1^P = 0 * 1\ 0\ 1$
$r_2^P = 0\ 0\ 0\ 0 *$
$r_3^D = 0 * * 0\ 1$
$r_4^D = 0\ 1\ 0\ 1 *$
$r_5^D = 0\ 1\ 1\ 1 *$
$r_6^P = 0\ 1 * * *$
$r_7^P = 0\ 0 * * *$
$r_8^P = 1\ 0 * 1 *$
$r_9^D = * * * * *$

REORDERING RULES

$$L(R_\sigma, F) \equiv \sum_{i=1}^{n-1} i|E(R_\sigma, \sigma^{-1}(i))|_F + (n-1)|E(R_\sigma, \sigma^{-1}(n))|_F$$

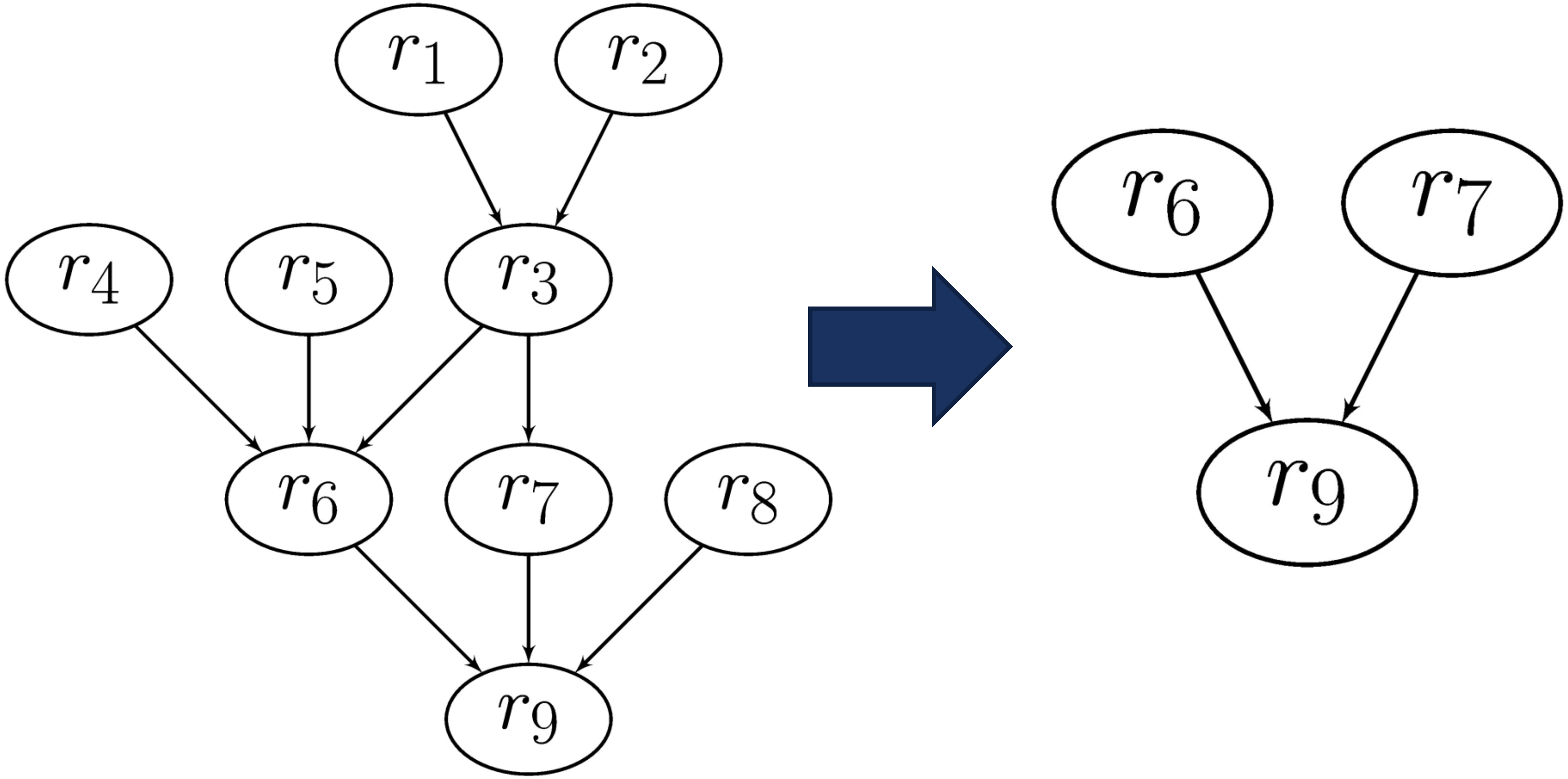
By placing the rules with large weights to higher position, the number of comparison of a packets can reduce.

→ Descending order of weight is desirable

↓

Because of dependency relation, most of rule list can not become descending order of weight

FIX OF SGM IN [2]



$$C[6] = 3$$
$$C[7] = 3$$
$$C[9] = 8$$

CONSIDERATION THE VARIATION OF WEIGHT

Classifier R
$r_1^P = * 0 * 1$
$r_2^P = 0 0 0 0$
$r_3^P = 0 * 0 0$
$r_4^D = 0 * 1 *$
$r_5^P = * 1 * 1$
$r_6^P = * * * 1$
$r_7^D = * * * *$

Classifier R'
$r_1^P = * 0 * 1$
$r_3^P = 0 * 0 0$
$r_2^P = 0 0 0 0$
$r_4^D = 0 * 1 *$
$r_5^P = * 1 * 1$
$r_6^P = * * * 1$
$r_7^D = * * * *$

r_2^P matches the packet $\{0000\}$
r_3^P matches the packet $\{0100\}$

r_2^P matches the packet $\{\}$
r_3^P matches the packet $\{0000, 0100\}$