A Method to Evaluate CFG Comparison Algorithms

2014 14th International Conference on Quality Software
Patrick P.F. Chan, Christian Collberg

김영철

2016. 6. 29.

EVALUATION METHOLOGY

- Graph Edit Distance 개념을 기초로 함
  - Add a zero-degree node
  - Delete a zero-degree node
  - Add an edge between two existing nodes
  - Delete an existing edge

- scores 대신에 rankings 을 사용

EVALUATION METHOLOGY

- Evaluation Algorithm
  - Algorithm 1은 evaluation process 정보를 줌
  - generate test case
  - compare seed CFG with test case
  - rank the results
  - compute a goodness score

$$
\begin{array}{l}
\textbf{procedure } \textsc{Evaluate}(\langle \mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k \rangle, G_0, d, n) \\
\quad \textbf{for } a \leftarrow 1, k \textbf{ do} \\
\quad\quad \textbf{for } t \leftarrow 1, n \textbf{ do} \\
\quad\quad\quad \textbf{for } g \leftarrow 1, d \textbf{ do} \\
\quad\quad\quad\quad \mathrm{scores}_g \leftarrow \Delta_{\mathcal{A}_a}(G_0, \mathrm{TestCases}_{t,g}) \\
\quad\quad\quad \textbf{end for} \\
\quad\quad\quad \mathrm{rank} \leftarrow \mathbf{rank}(\mathrm{scores}) \\
\quad\quad\quad \mathrm{corr}_{\mathcal{A}_a,t} \leftarrow \mathbf{pearson}((G_{t,1}, G_{t,2}, \ldots, G_{t,d}), \mathrm{rank}) \\
\quad\quad \textbf{end for} \\
\quad\quad \mathrm{avg}_{\mathrm{corr}_{\mathcal{A}_a}} = \mathbf{average}(\mathrm{corr}_{\mathcal{A}_a,1}, \mathrm{corr}_{\mathcal{A}_a,2}, \ldots, \mathrm{corr}_{\mathcal{A}_a,t}) \\
\quad \textbf{end for} \\
\quad \textbf{return } \mathbf{rank}(\mathrm{avg}_{\mathrm{corr}}) \\
\textbf{end procedure}
\end{array}
$$

Algorithm 1

EVALUATION METHOLOGY

- Benchmark Graph Generation
  - 하나의 EDG (Edit Distance Graph)는 seed CFG로부터 생성될 수 있는 가능한 모든 CFG를 나타냄
    - EDG는 무한하기 때문에 depth d를 설정
  - EDG nodes의 level이 생성됨
    - ex) i edit operations로 만들어진 EDG node는 level i에 위치한다
    - $ED(G_0, G_i) = i$

CFG SIMILARITY ALGORITHMS

A. Based on k-subgraphs Mining

B. Based on Edit Distance

C. Based on Neighbor Matching

D. Based on Simulation

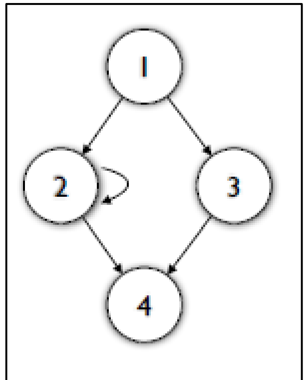
Based on k-subgraphs Mining

- Step1) Generating k-subgraphs
  - maximum outdegree는 2, maximum indegree가 1인 graphs의 set 생성
- Step2) Graph Fingerprinting
  - subgraph들은 숫자로 맵핑
- Step3) Computing Similarity
  - Jaccard index를 사용하여 비교

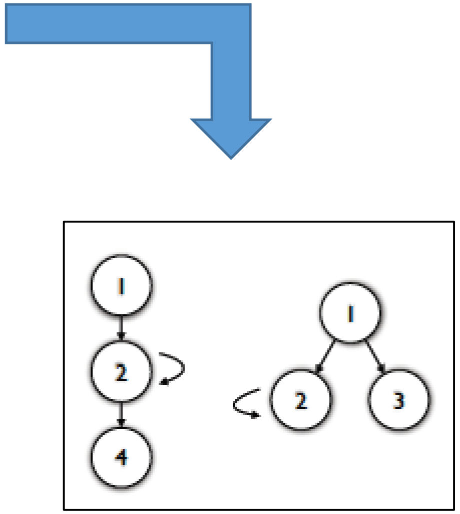
$$\frac{|\text{fingerprint}(A) \cap \text{fingerprint}(B)|}{|\text{fingerprint}(A) \cup \text{fingerprint}(B)|}$$

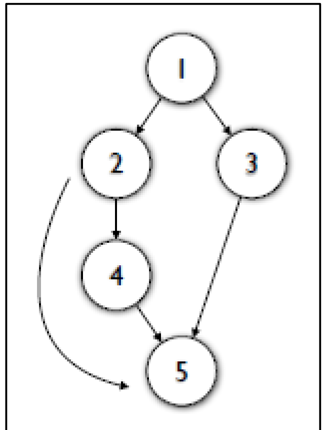
Jaccard index

Based on k-subgraphs Mining

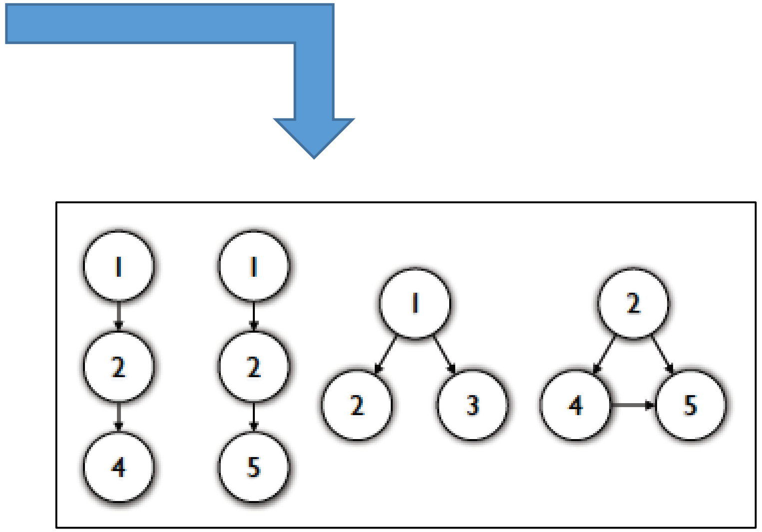


(a) CFG A.





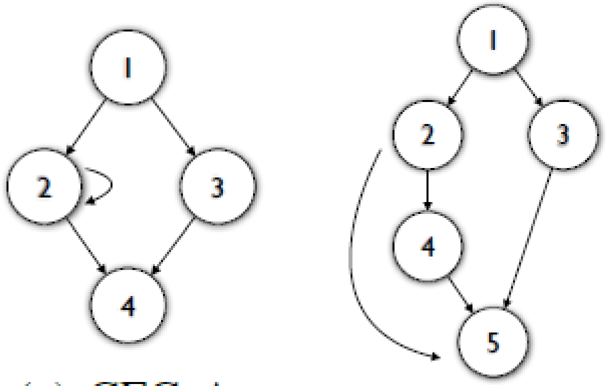
(b) CFG B.



두 그래프의 유사도는 0

Based on Edit Distance

- Step1) Building The Cost Matrix
  - G_1, G_2의 Vertex 수 V_1, V_2를 가지고 (|V1|+|V2|)*(|V1|+|V2|) matrix 생성



(a) CFG A.

(b) CFG B.

Cost of matching nodes

Cost of deleting nodes in CFG A

Cost of matching node 1 of CFG A to node 1 of CFG B

Cost of deleting node 4 of CFG B

Cost of deleting node 1 of CFG B

$$
\begin{pmatrix}
0 & 1 & 2 & 2 & 5 & 3 & \infty & \infty & \infty \\
2 & 1 & 2 & 2 & 3 & \infty & 5 & \infty & \infty \\
2 & 1 & 0 & 0 & 3 & \infty & \infty & 3 & \infty \\
4 & 3 & 2 & 2 & 1 & \infty & \infty & \infty & 3 \\
3 & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 \\
\infty & 4 & \infty & \infty & \infty & 0 & 0 & 0 & 0 \\
\infty & \infty & 3 & \infty & \infty & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & 3 & \infty & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & 4 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Cost of deleting nodes in CFG B

Cost of matching dummy nodes

Based on Edit Distance

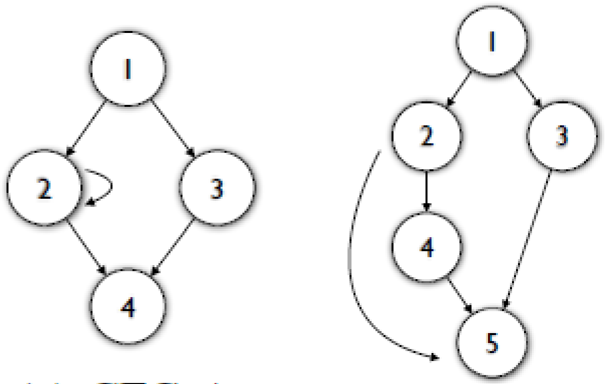
- Step2) Finding Best Match of the Nodes
  - 가장 낮은 코스트를 갖는 매칭 비용을 찾는 것이 목적
  - Hungarian algorithm이 assignment problem의 optimal solution을 찾음

- Step3) Computing Similarity

$$\mathrm{sim}(G_1, G_2) = 1 - \frac{\text{edit distance}}{|V_1| + |E_1| + |V_2| + |E_2|}$$

Based on Edit Distance

- Example
  - Hungarian algorithm을 이용하여 best matching 비용을 구함



(a) CFG A.

(b) CFG B.

$$
\begin{pmatrix}
0 & 1 & 2 & 2 & 5 & 3 & \infty & \infty & \infty \\
2 & 1 & 2 & 2 & 3 & \infty & 5 & \infty & \infty \\
2 & 1 & 0 & 0 & 3 & \infty & \infty & 3 & \infty \\
4 & 3 & 2 & 2 & 1 & \infty & \infty & \infty & 3 \\
3 & \infty & \infty & \infty & \infty & 0 & 0 & 0 & 0 \\
\infty & 4 & \infty & \infty & \infty & 0 & 0 & 0 & 0 \\
\infty & \infty & 3 & \infty & \infty & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & 3 & \infty & 0 & 0 & 0 & 0 \\
\infty & \infty & \infty & \infty & 4 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

Based on Neighbor Matching

- neighbor의 similarity에 근거하여 similarity matrix를 빌드함

- Step1) Topological Similarity
  - k=0, $a^0_{ij}=1$

$$a_{ij}^{k+1} = \frac{s_{\text{in}}^{k+1}(i,j) + s_{\text{out}}^{k+1}(i,j)}{2}$$

where

$$s_{\text{in}}^{k+1}(i,j) = \frac{1}{m_{\text{in}}} \sum_{l=1}^{n_{\text{in}}} a^k_{f_{ij}^{\text{in}}(l) g_{ij}^{\text{in}}(l)}$$

$$s_{\text{out}}^{k+1}(i,j) = \frac{1}{m_{\text{out}}} \sum_{l=1}^{n_{\text{out}}} a^k_{f_{ij}^{\text{out}}(l) g_{ij}^{\text{out}}(l)}$$

$$\begin{aligned} m_{\text{in}} &= \max(|\mathrm{IE}_i|, |\mathrm{IE}_j|) \\ n_{\text{in}} &= \min(|\mathrm{IE}_i|, |\mathrm{IE}_j|) \\ m_{\text{out}} &= \max(|\mathrm{OE}_i|, |\mathrm{OE}_j|) \\ n_{\text{out}} &= \min(|\mathrm{OE}_i|, |\mathrm{OE}_j|) \end{aligned}$$

- Step2) Node Similarity

$$a_{ij}^{k+1} = \sqrt{y_{ij} \cdot \frac{s_{\text{in}}^{k+1}(i,j) + s_{\text{out}}^{k+1}(i,j)}{2}}$$

Based on Simulation

- Labeled Transition Systems (LTS)를 사용하여 CFG를 만듦
- Recursively match the most similar outgoing nodes starting from the entry nodes
- Sum up the similarity of the matched nodes and edges

Based on Simulation

- entry n1, n2를 갖는 두 CFG의 유사도

$$S(n_1, n_2) = \begin{cases} N(n_1, n_2) & \text{, if } \mathrm{OD}_{n_1} = 0 \\ (1-p) \cdot N(n_1, n_2) + & \\ p \cdot max(W_1, W_2) & \text{, otherwise,} \end{cases}$$

where

$$p \in (0, 1)$$

$$W_1 = max_{n_2 \xrightarrow{b} n_2'} L(b, \epsilon) \cdot S(n_1, n_2')$$

$$W_2 = \frac{1}{\mathrm{OD}_{n_1}} \cdot \sum_{n_1 \xrightarrow{a} n_1'} max(max_{n_2 \xrightarrow{b} n_2'} (L(a, b) \cdot S(n_1', n_2')), L(a, \epsilon) \cdot S(n_1', n_2))$$