Rytter-style Algorithm for Context-Fre Path Querying

Semyon Grigorev
Saint Petersburg State University
St. Petersburg, Russia
semen.grigorev@jetbrains.com

Ekaterina Shemetova
Saint Petersburg State University
St. Petersburg, Russia
katyacyfra@gmail.com

ACM Reference Format:

Semyon Grigorev and Ekaterina Shemetova. 2020. Rytter-style Algorithm for Context-Fre Path Querying. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 THE REDUCTION

Suppose we have Φ — an instance of 3-SAT problem contains m clauses over k variables.

First of all, we should to constract a graph. To do it we follow the next steps.

(1) Let $\gamma_i = \{v_1 \leftarrow b_1, v_2 \leftarrow b_2, \cdots, v_k \leftarrow b_k\}$ where $b_k \in \{0, 1\}$. For each substitution γ_i a vertex V_{γ_i} should be created.
(2) For each V_{γ_i} the following edges should be added: $\{(V_{\gamma_i}, [v_j \leftarrow b_l]^+, V_{\gamma_i}) \mid v_j \leftarrow b_l \in \gamma_i\}$.
(3) For each clause $(l_1 \vee l_2 \vee l_3)$ the following subgraph should be created. First, two new vertices are edded: c_1 and c_2. After that, the following edges for each l_p and for each γ_i should be added

$$\{(c_1, [v_j \leftarrow b_l]^-, c_2) \mid b_l = \begin{cases} 1 \text{ if } l_p = v_j \\ 0 \text{ if } l_p = \neg v_j \end{cases}\}.$$

(4) Subgraph for all clauses should be connected sequencially. Suppose we have seqence of subgraps with vertices

$$\{(c_1^1, c_2^1), (c_1^2, c_2^2), \cdots, (c_1^m, c_2^m)\}.$$

To connect them we should merge vertices c_2^i and c_1^{i+1} for all i except $i = m$. After that we fix c_1^1 as a start vertex of formula subgraph, and c_2^m as a final vertex of formula subgraph.
(5) Finally, for all V_{γ_i} we should add the following edge

$$(V_{\gamma_i}, q, c_1^1)$$

The second part is a query. Suppose, we have p different substitutions. The gramamr is following

$$\begin{aligned} &S \rightarrow q \\ &S \rightarrow [v_1 \leftarrow b_1]^+ \ S \ [v_1 \leftarrow b_1]^- \\ &\quad \mid \cdots \\ &\quad \mid [v_k \leftarrow b_k]^+ \ S \ [v_k \leftarrow b_k]^- \end{aligned}$$

After that we should applay transformation which is described in the section 4. As a result we get h-Dyck reachability problem (yes, we can reduce it to 2-Dyck reachability).

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.1 An Example of Reduction

Suppose we have the following instance of 3-SAT problem.

$$\Phi = (\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_1 \vee x_3) \wedge (x_1 \vee \neg x_3 \vee x_2)$$

Substitutions:

$$\begin{aligned} \gamma_1 &= \{x_1 \leftarrow 0, x_2 \leftarrow 0, x_3 \leftarrow 0\} \\ \gamma_2 &= \{x_1 \leftarrow 1, x_2 \leftarrow 0, x_3 \leftarrow 0\} \\ \gamma_3 &= \{x_1 \leftarrow 0, x_2 \leftarrow 1, x_3 \leftarrow 0\} \\ \gamma_4 &= \{x_1 \leftarrow 0, x_2 \leftarrow 0, x_3 \leftarrow 1\} \\ \gamma_5 &= \{x_1 \leftarrow 1, x_2 \leftarrow 1, x_3 \leftarrow 0\} \\ \gamma_6 &= \{x_1 \leftarrow 1, x_2 \leftarrow 0, x_3 \leftarrow 1\} \\ \gamma_7 &= \{x_1 \leftarrow 0, x_2 \leftarrow 1, x_3 \leftarrow 1\} \\ \gamma_8 &= \{x_1 \leftarrow 1, x_2 \leftarrow 1, x_3 \leftarrow 1\} \end{aligned}$$

Graph for Φ is presented in figure 2.

The grammar:

$$\begin{aligned} S \rightarrow\ & [x_1 \leftarrow 0]^+ \ S \ [x_1 \leftarrow 0]^- \\ \mid\ & [x_2 \leftarrow 0]^+ \ S \ [x_2 \leftarrow 0]^- \\ \mid\ & [x_3 \leftarrow 0]^+ \ S \ [x_3 \leftarrow 0]^- \\ \mid\ & [x_1 \leftarrow 1]^+ \ S \ [x_1 \leftarrow 1]^- \\ \mid\ & [x_2 \leftarrow 1]^+ \ S \ [x_2 \leftarrow 1]^- \\ \mid\ & [x_3 \leftarrow 1]^+ \ S \ [x_3 \leftarrow 1]^- \\ \mid\ & q \end{aligned}$$

The intuition of such path finding is that substitution vertex (V_{γ_i}) should provide appropriate values for respective variable in appropriate order to satisfy the given formula. It can be done by appropriate traversing of loops. After that, each edge from c_i^j to c_l^k "uses" provided values to satisfie respective closure, and it can be done if and only if the respective vertex provides value required. This fact is expressed by usung balanced-bracket grammar. So, if there exists a path from V_{γ_i} to c_2^3, such that the corresponded word is derivable from S, then V_{γ_i} satisfy the given formula.

2 IMPROVED REDUCTION

Firs step is to split variables into three group of the same size. Suppose this splitting preserves the order. So, we have a set of partial substitution.

By the same way we create vertices for partial subctitutions.

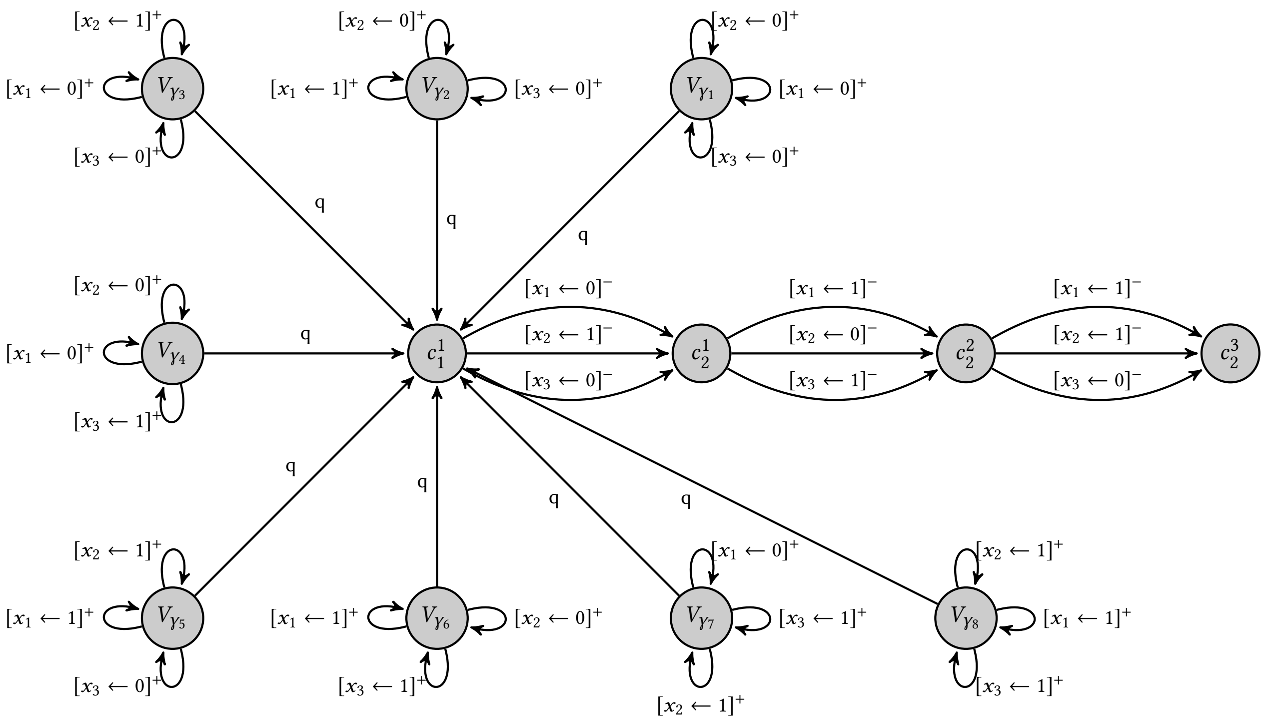

Figure 1: Example of graph for Φ

3 AN EXAMPLE OF IMPROVED REDUCTION

For our example:

$$
\begin{aligned}
\gamma_1^1 &= \{x_1 \leftarrow 1\}\\
\gamma_1^2 &= \{x_1 \leftarrow 0\}\\
\gamma_2^1 &= \{x_2 \leftarrow 1\}\\
\gamma_2^2 &= \{x_2 \leftarrow 0\}\\
\gamma_3^1 &= \{x_3 \leftarrow 1\}\\
\gamma_3^2 &= \{x_3 \leftarrow 0\}
\end{aligned}
$$

Grammar:

$$
\begin{aligned}
& S \to S_1 \mid S_2 \mid S_3 \mid S_4 \mid S_5 \mid S_6 \mid S_7 \mid S_8\\
& S_1 \to q \mid p^+\ S_1\ p^- \mid [x_1 \leftarrow 0]^+\ S_1\ [x_1 \leftarrow 0]^-\\
& \quad \mid [x_2 \leftarrow 0]^+\ S_1\ [x_2 \leftarrow 0]^- \mid [x_3 \leftarrow 0]^+\ S_1\ [x_3 \leftarrow 0]^-\\
& S_2 \to q\\
& \quad \mid p^+\ S_2\ p^- \mid [x_1 \leftarrow 1]^+\ S_2\ [x_1 \leftarrow 1]^-\\
& \quad \mid [x_2 \leftarrow 0]^+\ S_2\ [x_2 \leftarrow 0]^- \mid [x_3 \leftarrow 0]^+\ S_2\ [x_3 \leftarrow 0]^-\\
& S_3 \to q \mid p^+\ S_3\ p^- \mid [x_1 \leftarrow 0]^+\ S_3\ [x_1 \leftarrow 0]^-\\
& \quad \mid [x_2 \leftarrow 1]^+\ S_3\ [x_2 \leftarrow 1]^- \mid [x_3 \leftarrow 0]^+\ S_3\ [x_3 \leftarrow 0]^-\\
& S_4 \to q \mid p^+\ S_4\ p^- \mid [x_1 \leftarrow 0]^+\ S_4\ [x_1 \leftarrow 0]^-\\
& \quad \mid [x_2 \leftarrow 0]^+\ S_4\ [x_2 \leftarrow 0]^- \mid [x_3 \leftarrow 1]^+\ S_4\ [x_3 \leftarrow 1]^-\\
& S_5 \to q \mid p^+\ S_5\ p^- \mid [x_1 \leftarrow 1]^+\ S_5\ [x_1 \leftarrow 1]^-\\
& \quad \mid [x_2 \leftarrow 1]^+\ S_5\ [x_2 \leftarrow 1]^- \mid [x_3 \leftarrow 0]^+\ S_5\ [x_3 \leftarrow 0]^-\\
& S_6 \to q \mid p^+\ S_6\ p^- \mid [x_1 \leftarrow 1]^+\ S_6\ [x_1 \leftarrow 1]^-\\
& \quad \mid [x_2 \leftarrow 0]^+\ S_6\ [x_2 \leftarrow 0]^- \mid [x_3 \leftarrow 1]^+\ S_6\ [x_3 \leftarrow 1]^-\\
& S_3 \to q \mid p^+\ S_7\ p^- \mid [x_1 \leftarrow 0]^+\ S_7\ [x_1 \leftarrow 0]^-\\
& \quad \mid [x_2 \leftarrow 1]^+\ S_7\ [x_2 \leftarrow 1]^- \mid [x_3 \leftarrow 1]^+\ S_7\ [x_3 \leftarrow 1]^-\\
& S_8 \to q \mid p^+\ S_8\ p^- \mid [x_1 \leftarrow 1]^+\ S_8\ [x_1 \leftarrow 1]^-\\
& \quad \mid [x_2 \leftarrow 1]^+\ S_8\ [x_2 \leftarrow 1]^- \mid [x_3 \leftarrow 1]^+\ S_8\ [x_3 \leftarrow 1]^-
\end{aligned}
$$

4 FROM ARBITRARY CFPQ TO DYCK QUERY

This reduction is inspired by the construction described in [1].

Consider a context-free grammar $\mathcal{G} = (\Sigma, N, P, S)$ in BNF where Σ is a terminal alphabet, N is a nonterminal alphabet, P is a set of productions, $S \in N$ is a start nonterminal. Also we denote a directed labeled graph by $G = (V, E, L)$ where $E \subseteq V \times L \times V$ and $L \subseteq \Sigma$.

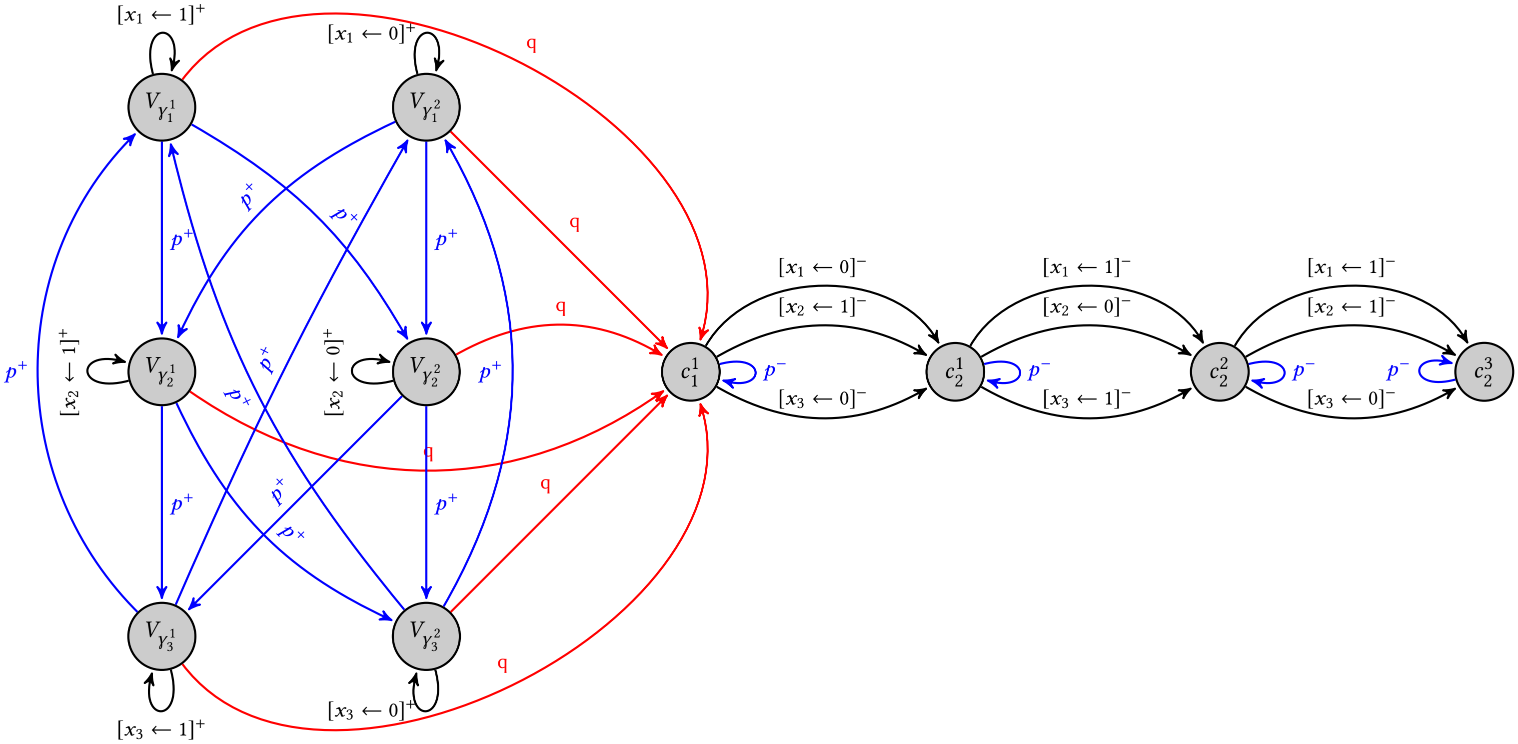

Figure 2: Example of graph for Φ

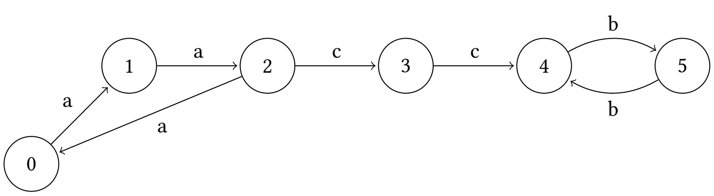

Figure 3: The input graph

We should construct new input graph G' and new grammar $\mathcal{G}'$ such that $\mathcal{G}'$ specifies a Dyck language and there is a simple mapping from CFPQ($\mathcal{G}'$, G') to CFPQ($\mathcal{G}$, G). Step-by-step example with description is provided below.

Let the input grammar is

$$S \rightarrow a\ S\ b \mid a\ C\ b$$
$$C \rightarrow c \mid C\ c$$

The input graph is presented in fig. 3.

(1) Let $\Sigma_{()} = \{t_(, t_) | t \in \Sigma\}$.
(2) Let $N_{()} = \{N_(, N_) | N \in N\}$.
(3) Let $M_{\mathcal{G}} = (V_{\mathcal{G}}, E_{\mathcal{G}}, L_{\mathcal{G}})$ is a directed labeled graph, where $L_{\mathcal{G}} \subseteq (\Sigma_{()} \cup N_{()})$. This graph is created the same manner as described in [1] but we do not require the grammar be in CNF. Let $x \in V_{\mathcal{G}}$ and $y \in V_{\mathcal{G}}$ is "start" and "final" vertices respectively. This graph may be treated as a finite automaton, so it can be minimized and we can compute an ε-closure if the input grammar contains ε productions. The graph $M_{\mathcal{G}}$ for our example is presented in fig. 4. The minimized graph is presented in fig. 5.
(4) For each $v \in V$ create $M_{\mathcal{G}}^v$: unique instance of $M_{\mathcal{G}}$.
(5) New graph $G^{'}$ is a graph G where each label t is replaced with $t_)^i$ and some additional edges are created:
   - Add an edge $(v', S_(, v)$ for each $v \in V$.
   - And the respective $M_{\mathcal{G}}^v$ for each $v \in V$:

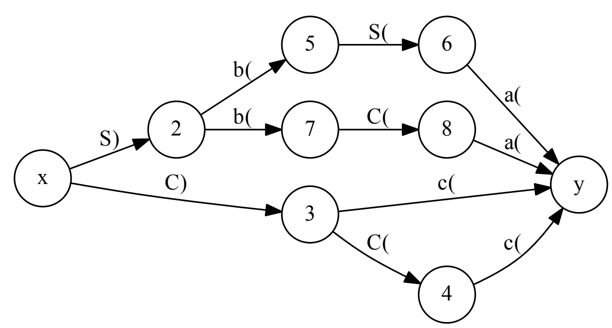

Figure 4: The $M_{\mathcal{G}}$ graph

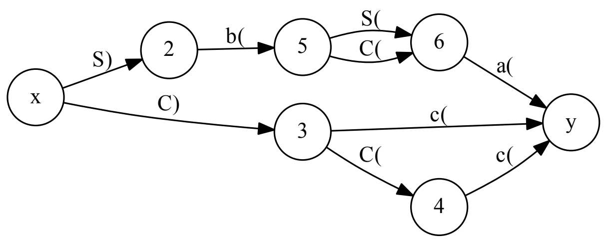

Figure 5: The minimized $M_{\mathcal{G}}$

- reattach all edges outgoing from x^v ("start" vertex of $M^v_{\mathcal{G}}$) to v;
- reattach all edges incoming to y^v ("final" vertex of $M^v_{\mathcal{G}}$) to v.

New input graph is ready. It is presented in fig. 6.

(6) New grammar $\mathcal{G}' = (\Sigma^{'}, N', P', S')$ where $\Sigma^{'} = \Sigma_{()} \cup N_{()}$, $N' = \{S'\}$, $P' = \{S' \to b_(\; S' \; b_); S' \to b_(\; b_) \mid b_(, b_) \in \Sigma^{'}\} \cup \{S' \to S' \; S'\}$ is a set of productions, $S' \in N'$ is a start nonterminal.

Now, if CFPQ$(\mathcal{G}', G')$ contains a pair (u'_0, v') such that $e = (u'_0, S_(, u'_1) \in E'$ is an extension edge (step 5, first subitem), then $(u'_1, v') \in$ CFPQ$(\mathcal{G}, G)$. In our example, we can find the following path: $7 \xrightarrow{S_(} 1 \xrightarrow{S_)} 22 \xrightarrow{b_(} 25 \xrightarrow{C_(} 26 \xrightarrow{a_(} 1 \xrightarrow{a_)} 2 \xrightarrow{C_)} 33 \xrightarrow{C_(} 34 \xrightarrow{c_(} 2 \xrightarrow{c_)} 3 \xrightarrow{C_)} 43 \xrightarrow{c_(} 3 \xrightarrow{c_)} 4 \xrightarrow{b_)} 5$. Edge $7 \xrightarrow{S_(} 1$ is the extension, so (1,5) should be in CFPQ$(\mathcal{G}, G)$ and it is true.

REFERENCES

[1] Krishnendu Chatterjee, Bhavya Choudhary, and Andreas Pavlogiannis. 2017. Optimal Dyck Reachability for Data-dependence and Alias Analysis. *Proc. ACM Program. Lang.* 2, POPL, Article 30 (Dec. 2017), 30 pages. https://doi.org/10.1145/3158118

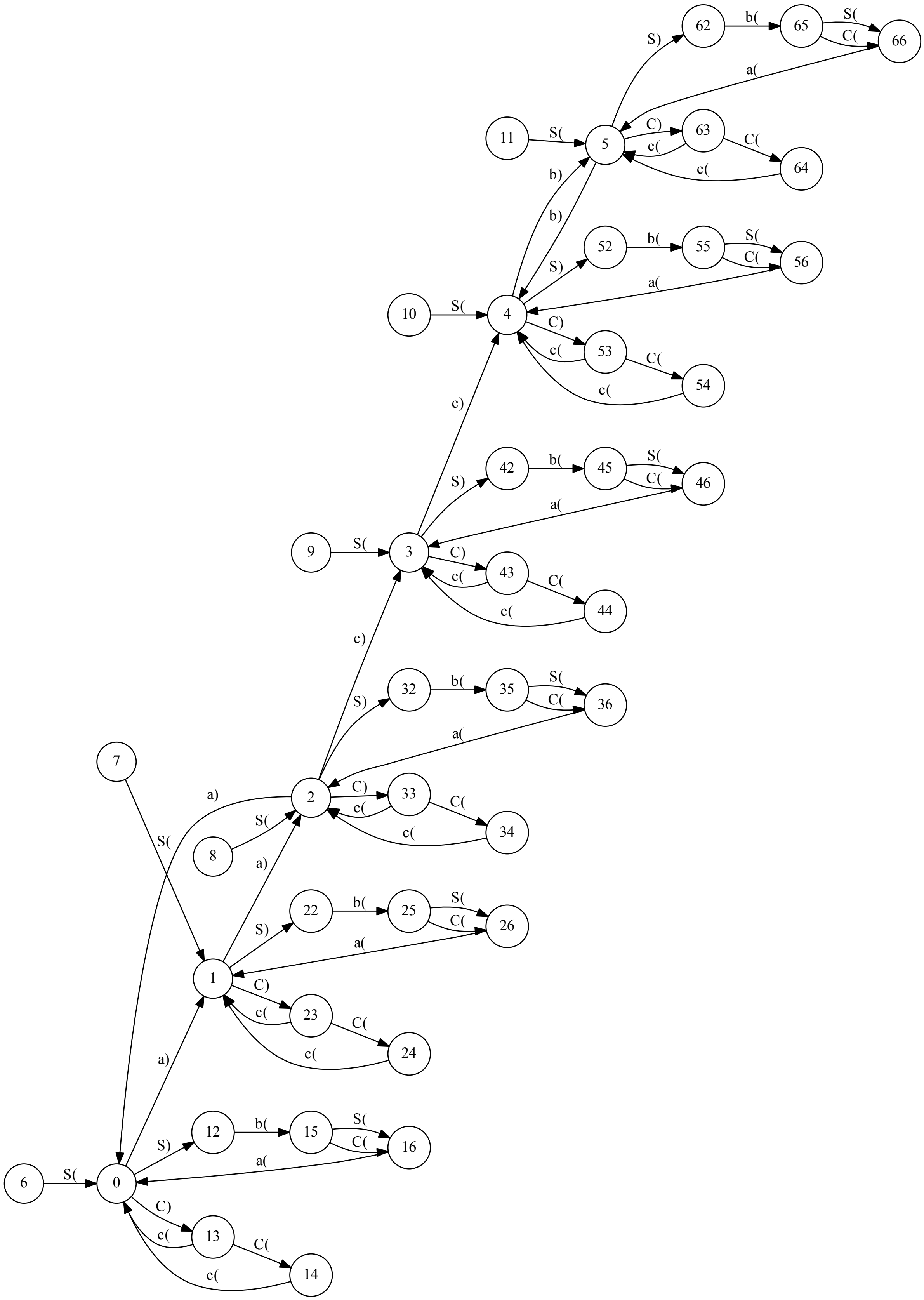

Figure 6: New input graph