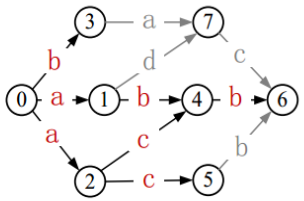
Answering Reachability Queries with Ordered Label Constraints over Labeled Graphs

Daoliang HE, Pingpeng YUAN, Hai JIN

Frontiers of Computer Science, DOI: 10.1007/s11704-022-2368-y

Problems & Ideas

- Ordered-Label-Constrained Reachability(OLCR) query:
  - The constraint is not just a label set but a label sequence in many scenarios.
  - Existing LCR solutions cannot answer OLCR queries directly.
  - It's not trivial to extend LCR solutions to OLCR as we need to build $\lfloor e|\Sigma|! - 1 \rfloor$ indices to cover all possible label sequences. (is the $|\Sigma|$bel set of graph)
- Ideas: combining constrained DFS with our proposed efficient partial indexing technique call DHL (dual hash labeling).



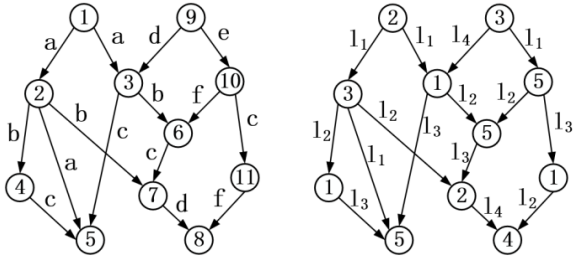
(a) A toy graph G_0.

DFS order

V	V_0	V_2	V_5	V_4	V_3	V_1	V_4	V_6
state	N	0	2	2	1	0	1	1

(b) *state(v)* with DFS travesal.

Fig. 1: A toy graph G_0 and the *state(v)* for querying $reach(v_0, v_6, a^*b^*c^*)$.



(a) An edge-labeled graph G. (b) A permutation of $h_1(v)$ and $h_2(l)$ for $v \in G$ and $l \in \Sigma$.

Fig. 2: A toy graph G and the corresponding permutation.

Figure 1 shows the procedures of constrained DFS. We search graph in DFS style and iteratively conduct following two steps: 1) update the state of visited vertex and prune the invalid branches which is not consistent with given pattern; 2) visit DHL index to prune the branches which is not reachable with corresponding label set in advance. Figure 2 shows an example of constructing dual hash labeling index: we hash both vertex IDs and edge labels with bloom filter. We can test the label constrained reachability with corresponding bloom filter bit arrays to quickly figure out many false queries.

Main Contributions

- Contributions:
  - A novel reachability model named OLCR, where the constraint consists of a sequence of labels instead of only label set.
  - A novel space-efficient indexing technique with bounded false positive rate called DHL (dual hash labeling) based on bloom filter.
  - A method combining constrained DFS with DHL index to answer OLCR queries.

Dataset	LI+		P2H+		**DHL** (K=5)	
	IT	IS	IT	IS	IT	IS
RT	0.03	1.1	0.01	0.72	0.01	0.15
ADG	3.4	49	0.74	14	0.16	2.4
AX	76	240	306	555	4.0	36.5
BG	214	560	65	321	2.5	24.8
SAM	99	107	37	56	4.1	6.8
YT	1.4k	435	168	150	12.3	13.2
SFC	836	426	1.5k	467	29.7	26.7
WS	2.1k	4.3k	1.3k	2.4k	22.8	348
WG	-	-	-	-	64	1.3k
CU	-	-	-	-	96	3.7k

* IT denotes the indexing time, IS denotes the index size

Table 1. indexing time and index size.

Name	|Σ|/4 or 2			|Σ|/2 or 4			|Σ| – 2 or 6		
	LI+	P2H+	DHL	LI+	P2H+	DHL	LI+	P2H+	DHL
RT	0.1	0.2	0.03	0.1	0.17	0.06			
ADG	1.05	0.68	0.62	0.45	0.53	0.7			
AX	86	2.6	7.4	105	1.8	3.9	191	1.1	1.8
BG	82	1.4	4.9	137	1.0	4.0	181	0.8	1.8
SAM	21.4	0.89	5.9	8.7	0.96	26	21.5	0.76	8.0
YT	106	0.86	90	69	0.91	68			
SFC	252	3.4	68	437	2.8	37	275	3.0	39
WS	14	1.3	0.03	123	1.3	0.27	798	1.64	1.28
WG	-	-	0.04	-	-	0.37	-	-	27.8
CU	-	-	0.01	-	-	0.01	-	-	0.02

* QT denotes the average query time

Table 2. query time of DHL.

Name	|Σ|/4 or 2					
	LI+		P2H+		DHL+	
	F	R	F	R	F	R
RT	0.47	0.45	0.89	0.41	0.14	0.09
ADG	3.2	6.9	2.1	15.3	1.9	5.6
AX	93	-	4.1	705ms	8.4	625ms
BG	90	131ms	2.4	25ms	5.7	32ms
SAM	27	1.2ms	1.9	210	6.9	1.4ms
YT	115	-	1.4	423ms	90	-
SFC	265	-	4.3	-	68	-
WS	98	9.9	11	8.3	0.21	0.23
WG	-	-	-	-	0.28	0.24

Table 3. query time of DHL+.

Table 1 shows the indexing time and index size of DHL versus two state-of-arts LI+ and P2H+. Table 2 shows the query time of the three methods with random LCR queries. Table 3 shows the query time of DHL+ versus the simple extension of LI+ and P2H+.