Yashashwin Karthikeyan, Roozbeh Ali
ECE 108 - Discrete Mathematics and Logic 1
University of Waterloo

March 25, 2024

ASSIGNMENT 7 — List Induction and Graph Theory

1 Proof of Deletion

1.1 Read

Yes.

1.2 Proof

$\vdash \forall xs\colon list(T_y).\ \forall x\colon T_y.\ \forall y \in del(x, xs).\ y \neq x.$

By list induction on goal,

Subproof 1 (Base case)

$\vdash \forall x\colon T_y.\ \forall y \in del(x, [\,]).\ y \neq x.$

By forall-elim on goal,

(1.1)$x\colon T_y$
$\vdash \forall y \in del(x, [\,]).\ y \neq x.$

By subst. definition of *del* into goal by logic,

$\vdash\ \forall y \in [\,].\ y \neq x.$

QED by empty-list membership axiom and logic.

Subproof 2 (Inductive step)

(2.1)$xs\colon list(T_y)$
(2.2)$\forall x\colon T_y.\ \forall y \in del(x, xs).\ y \neq x.$
(2.3)$x'\colon T_y.$
$\vdash \forall x\colon T_y.\ \forall y \in del(x, push(x', xs)).\ y \neq x.$

By forall-elim on goal,

(2.4)$x\colon T_y.$
$\vdash \forall y \in del(x, push(x', xs)).\ y \neq x.$

By forall-elim on (2.2) using (2.4),

(2.5)$\forall y \in del(x, xs).\ y \neq x.$

By cases $x = x'$, $x \neq x'$,

Subproof 2.1 (Completeness)

$\vdash (x = x') \vee (x \neq x')$

QED by logic.

Subproof 2.2 (Equality case)

$(3.1) x = x'$
$\vdash \forall y \in del(x, push(x', xs)).\ y \neq x.$

By rev subst. (3.1) into goal,

$\vdash \forall y \in del(x, push(x, xs)).\ y \neq x.$

By subst. definition of *del* into goal and the front axiom,

$\vdash \forall y \in del(x, pop(push(x, xs))).\ y \neq x.$

By the pop axiom,

$\vdash \forall y \in del(x, xs).\ y \neq x.$

QED by (2.5).

Subproof 2.2 (Inequality case)

$(4.1) x \neq x'$
$\vdash \forall y \in del(x, push(x', xs)).\ y \neq x.$

By susbt. definition of *del* into goal, (4.1) and the front axiom,

$\vdash \forall y \in push(x', del(x, pop(push(x', xs)))).\ y \neq x.$

By subst. pop axiom by logic,

$\vdash \forall y \in push(x', del(x, xs)).\ y \neq x.$

By subst. universal quantification over a list by logic,

$\vdash (front(push(x', del(x, xs))) \neq x)\ \wedge\ (\forall y \in pop(push(x', del(x, xs))).\ y \neq x.)$

By subst. pop axiom and front axiom into goal by logic,

$\vdash (x' \neq x)\ \wedge\ (\forall y \in del(x, xs).\ y \neq x.)$

QED by (4.1) and (2.5).

TA Notes

1. For the defn subst, we know the value in both cases by the equality assumptions, the front axiom, and the size being at least 1 for any return of the push function.

2. Universal quantification requires a set of non-zero size, and by definition any return of the push function has non-zero size.

2 Categories of graphs

2.1 List of nodes win

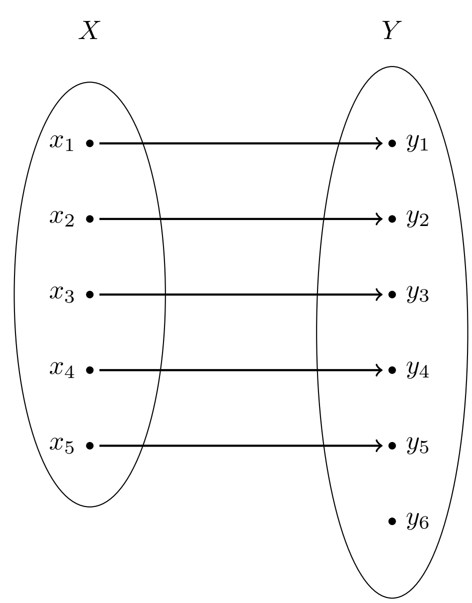
In undirected, directed and simple graphs, a list of nodes is advantageous for choosing a path. In undirected graphs, bi-directional edges play no importance, because a path cannot have repeat edges. Thus, a list of nodes effectively descibes a path for an undirected graph with minimal information. This similarly makes a list of nodes effective for simple graphs, because they are undirected. Directed graphs preserve direction by virtue of their edges being uni-directional. As such, a list of nodes would be preferrable in order to minimze the information required to make a graph.

2.2 List of edges win

A list of edges would be advantageous for a multigraph. Since nodes can be the times, a list of edges is useful to identify the start and end point amongst potential parallel edges, and thus form a path. Choosing a path to be a list of edges would be useful for multigraphs. A core axiom of multigraphs are that

3 Graphs and injective functions

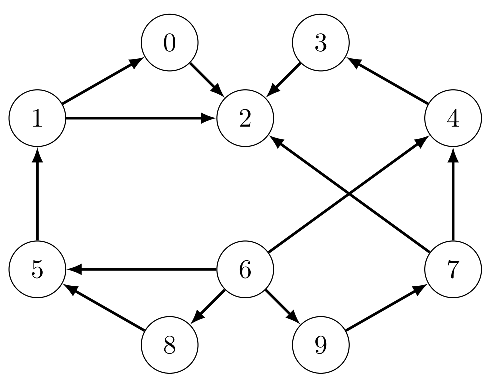
To represent an injective function, we move the endpoint of the edge (x_4, y_3) to y_4, and delete the edge (x_5, y_4). These operations have a total cost of 3. The following graph represents an injective function after the aforementioned operations are performed.



4 DAG

In order to convert the graph into a directed acyclic graph (DAG), we must remove at minimum 1 edge. If we represent edges in (tail, head) tuples, then by deleting $(2, 6)$, we create a DAG.

Submitted by Yashashwin Karthikeyan, Roozbeh Ali on March 25, 2024.


0
3
1
2
4
5
6
7
8
9