On graphs whose nodes are Black or White

In this note we consider finite directed graphs. Each edge is considered to be directed from some node, called its source, to some node, called its target. The in-degree of a node is defined as the number of edges that have that node as their target; the out-degree of a node is defined as the number of edges for which that node is the source. Finally we postulate that each node is either black or white. We now observe

(0)   the sum of the in-degrees of the black nodes
=   {when in doubt, consult the appendix}
        the number of edges with a black target
=   { a source is white or black}
        #WB + #BB

where #WB denotes the number of edges with a white source and a black target (and for #BB, etc. analogously). By colouring all nodes black, the first equality of (0) yields

(1)   the sum of the indegrees of all nodes equals the number of edges.

Interchanging source and target we get from (0)

(2)   the sum of the out-degrees of the
       black nodes equals #BW + #BB

and from (1)

(3)   the sum of the out-degrees of all
       nodes equals the number of edges.

Combining (0) and (2) we get

<u>Lemma</u> If the sum of the out-degrees of
the black nodes equals the sum of the in-
degrees of the black nodes, then
$$\# WB = \# BW \quad .$$

<u>Remark</u> From (1) and (3) we conclude

(4)   the sum of the in-degrees equals the
       sum of the out-degrees

and hence an alternative formulation for
the Lemma is

<u>Lemma'</u> If the sum of the out-degrees
of the white nodes equals the sum of the
in-degrees of the white nodes, then
$$\# WB = \# BW \quad .$$

(End of Remark.)

    The Lemma has the
<u>Corollary</u> If each node's in-degree equals
its out-degree, then $\# WB = \# BW$.

2

Appendix

With finite types node and edge we declare

n : node ;

e : edge ;

s, t : edge → node   (for "source" and
            "target" respectively) ;

in, out : node → nat   (for "in-degree" and
            "out-degree" respectively)

defined by $in.n = \langle \Sigma e: t.e = n: 1 \rangle$

$out.n = \langle \Sigma e: s.e = n: 1 \rangle$; .

b, w : node → bool (for "black" and "white"
            respectively)

satisfying $b.n \equiv \neg\, w.n$.

For the derivation of (0) we observe

$\langle \Sigma n: b.n: in.n \rangle$

=   {definition of in}

$\langle \Sigma n: b.n: \langle \Sigma e: t.e = n: 1 \rangle\rangle$

=   {unnesting}

$\langle \Sigma n, e: b.n \land t.e = n: 1 \rangle$

=   { Leibniz}

$\langle \Sigma n, e: b.(t.e) \land t.e = n: 1 \rangle$

=   { nesting}

$\langle \Sigma e: b.(t.e): \langle \Sigma n: t.e = n: 1 \rangle\rangle$

=   { 1-point rule}

$\langle \Sigma e: b.(t.e): 1 \rangle$

=   { notational liberty}

\# ? B

3

At first sight, the above derivation of

$$\langle \Sigma n: b.n: in.n \rangle = \langle \Sigma e: b.(t.e): 1 \rangle$$

seems an overkill. I have tried all sorts of alternatives — counting marks in a 2-dimensional table, mathematical induction over the number of edges or the number of black nodes. All these methods could, of course, be made to work, but if you wanted to eliminate the handwaving, the argument became longer than the above that uses no more than the standard quantification manipulation. I had the privilege of discussing the alternatives with Maarten Boasson and Fred B. Schneider, who happened to visit UT simultaneously.

Austin, 20 February 1999

prof. dr Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 - 1188
USA