Algorithmische Graphentheorie
SS 09

Planarity testing

Procedure Search(v)

mark v "old";
$DFN(v) := COUNT$;
$COUNT := COUNT + 1$;
$LOW(v) := DFN(v)$;
for $w \in Adj(v)$ **do**
  if w is marked "new" **then**
    Add $\{v, w\}$ doe DFS-Tree T;
    $FATH(w) := v$;
    SEARCH(w);
    $LOW(v) :=$ min$\{LOW(v), LOW(w)\}$;
  else if $w \neq FATH(v)$ **then**
    $LOW(v) :=$ min$\{LOW(v), DFN(w)\}$;
  end if
end for

Procedure DFS(G)

$T := \emptyset$; { T is a DFS-Tree}
$COUNT := 1$;
mark each vertex of G as "new";
select an arbitrary vertex v of G;
SEARCH(v);

Procedure ST-Number(G)

mark s, t, and $\{s,t\}$ as "old" and all other vertices and edges as "new";
push down t and s into a stack S in this order;
$\{s$ is over $t\}$
$COUNT := 1$;
pop up the top entry v from S;
while $v \neq t$ **do**
  if $PATH(v) = \emptyset$ **then**
    $STN(v) := COUNT$;$COUNT := COUNT + 1$;
  else
    let $PATH(v) := vu_1u_2 \ldots u_kw$;
    push down the vertices $u_k, u_{k-1}, \ldots, u_1, v$ into S in this order; $\{$ v is a top entry of S $\}$
  end if
  pop up the top entry v from S;
end while
STN(t):=COUNT;

Procedure PLANAR(G)

G is a given graph;
assign st-numbers to all vertices of G and name the vertices by these numbers;
construct a PQ-tree corresponding to G_1
{ a single P-node with virtual edges incident on source $s = 1$ }
for $v = 2$ to n **do**
    { reduction step}
    try to gather all the pertinent leaves by repeatedly applying the template matchings from the leaves to the root of the pertinent subtree;
    if the reduction fails **then**
        print out "G is not planar";
        return;
    end if
    { vertex addition step}
    replace all the full nodes of the PQ-tree by a new P-node (which corresponds to a cut-vertex v in G'_v);
    add to the PQ-tree all the neighbors of v larger than v as the sons of the P-node;
end for
print out "G is planar";