

1 Notes

- $V(X)$ – vertices of structure X . Will be written as X when obvious.
- $a - b$, when a and b are nodes – a and b are neighbors.
- $a \cdots b$, when a and b are nodes – a and b are not neighbors.
- $a - X$, when a is a node and X is a set of nodes – a has a neighbor in X .
- $a \cdots X$, when a is a node and X is a set of nodes – a has a nonneighbor in X .
- $a \blacktriangleleft X$, when a is a node and X is a set of nodes – a is complete to X .
- $a \ntriangleleft X$, when a is a node and X is a set of nodes – a is anticomplete to X .
- $X \blacksquare Y$, when X and Y are set of nodes – X is complete to Y .
- $X \not\sqsupset Y$, when X and Y are set of nodes – X is anticomplete to Y .
- $[n] - \{1, \dots, n\}$.
- $L(BS(K_4))$ – a line-graph of a bipartite subdivision of K_4 .
- $a \leftarrow b$ – let a be equal b .
- $a : \in X$ – let a be equal to any element of X
- $a \underline{\vee} b$ – a xor b

2 Algorithms

COLOR-GOOD-PARTITION($G, K_1, K_2, K_3, L, R, c_1, c_2$)

Input: G – square-free, Berge graph
 K_1, K_2, K_3, L, R – good partition
 c_1, c_2 – colorings of $G \setminus R$ and $G \setminus L$ (possibly NULL)

Output: $\omega(G)$ -coloring of G

```

1  $G_1 \leftarrow G \setminus R$ 
2  $G_2 \leftarrow G \setminus L$ 
3 if  $c_1, c_2 = \text{NULL}$  then
4    $c_1 \leftarrow \text{COLOR-GRAPH}(G_1)$ 
5    $c_2 \leftarrow \text{COLOR-GRAPH}(G_2)$ 
6 foreach  $u \in K_1 \cup K_2$  do
7   relabel  $c_2$ , so that  $c_1(u) = c_2(u)$ 
8  $B \leftarrow \{u \in K_3 : c_1(u) \neq c_2(u)\}$ 
9 if  $B = \emptyset$  then return  $c_1 \cup c_2$ 
10 foreach  $h \in [2]$ , distinct colors  $i, j$  do
11    $G_h^{i,j} \leftarrow$  subgraph induced on  $G_h$  by  $\{v \in G_h : c_h(v) \in \{1, 2\}\}$ 
12 foreach  $u \in K_3$  do
13    $C_h^{i,j}(u) \leftarrow$  component of  $G_h^{i,j}$  containing  $u$ 
    ASSERT:  $C_h^{c_1(u), c_2(u)}(u) \cap K_2 = \emptyset$ 
14 if  $\exists u \in B, h \in [2] : C_h^{c_1(u), c_2(u)}(u) \cap K_1 = \emptyset$  then
15    $c'_1 \leftarrow c_1$  with colors  $i$  and  $j$  swapped in  $C_1^{i,j}(u)$ 
    ASSERT:  $c'_1$  and  $c_2$  agree on  $K_1 \cup K_2$ 
    ASSERT:  $\forall u \in K_3 \setminus B : c'_1(u) = c_1(u)$ 
    ASSERT:  $c'_1(u) = j = c_2(u)$ 
16   return COLOR-GOOD-PARTITION( $G, K_1, K_2, K_3, L, R, c'_1, c_2$ )
17 else
18    $w \leftarrow$  vertex of  $B$  with most neighbors in  $K_1$ 
    ASSERT:  $\forall u \in B : N(u) \cap K_1 \subset N(w) \cap K_1$ 
19   relabel  $c_1, c_2$ , so that  $c_1(w) = 1, c_2(w) = 2$ 
20    $P \leftarrow$  chordless path  $w - p_1 - \dots - p_k - a$  in  $C_1^{1,2}(w)$  so that
      $k \geq 1, p_1 \in K_3 \cup L, p_2 \dots p_k \in L, a \in K, c_1(a) \in [2]$ 
21    $Q \leftarrow$  chordless path  $w - q_1 - \dots - q_l - a$  in  $C_2^{1,2}(w)$  so that
      $l \geq 1, q_1 \in K_3 \cup R, q_2 \dots q_l \in R, a \in K, c_2(a) \in [2]$ 
22    $i \leftarrow c_1(a)$ 
23    $j \leftarrow 3 - i$ 
    ASSERT: exactly one of the colors 1 and 2 appears in  $K_1$  (as in
    Lemma 2.2.(3))
    ASSERT:  $|P|$  and  $|Q|$  have different parities
    ASSERT:  $p_1 \in K_3 \vee p_2 \in K_3$  (as in Lemma 2.2.(4))
    ASSERT:  $\nexists y \in K_3 : c_1(y) = 2 \wedge c_2(y) = 1$  (as in Lemma 2.2.(5))
24   if  $p_1 \in K_3$  then
     | ASSERT:  $c_2(p_1) \notin [2]$ 
25   | relabel  $c_2$ , so that  $c_2(p_1) = 3$ 

```

```

25 // else //  $\nexists u \in B, h \in [2] : C_h^{c_1(u), c_2(u)}(u) \cap K_1 = \emptyset$ 
26 // if  $p_1 \in K_3$  then
27   ASSERT: color 3 does not appear in  $K_2$ 
28   ASSERT: color 3 does not appear in  $K_1$ 
29   ASSERT:  $C_2^{j,3}(p_1) \cap K_1 = \emptyset$ 
30    $c'_2 \leftarrow c_2$  with colors  $j$  and 3 swapped in  $C_2^{j,3}(p_1)$ 
31   ASSERT:  $j = 2$ 
32   return COLOR-GOOD-PARTITION( $G, K_1, K_2, K_3, L, R, c_1, c'_2$ )
33 else
34   relabel  $c_1$ , so that  $c_1(q_1) = 3$ 
35   if 3 does not appear in  $K_1$  then
36     ASSERT:  $C_1^{j,3}(q_1) \cap K_1 = \emptyset$ 
37     ASSERT:  $j = 1$ 
38      $c'_1 \leftarrow c_1$  with colors  $j$  and 3 swapped in  $C_1^{j,3}(q_1)$ 
39     return COLOR-GOOD-PARTITION( $G, K_1, K_2, K_3, L, R, c'_1, c_2$ )
40   else
41     ASSERT:  $q_1 \nrightarrow \{a, a_3\}$ 
42     ASSERT:  $C_1^{i,3}(q_1) \cap K_1 = \emptyset$ 
43     ASSERT:  $i = 1$ 
44      $c'_1 \leftarrow c_1$  with colors  $i$  and 3 swapped in  $C_1^{i,3}(q_1)$ 
45     return COLOR-GOOD-PARTITION( $G, K_1, K_2, K_3, L, R, c'_1, c_2$ )

```

GROW-HYPERPRISM(G, G, M, F)

// Lemma 3.3

Input: G – square-free, Berge graph

$H = (A_1, \dots, B_3)$ – a hyperprism in G

M – the set of major neighbors of H in G

F – a minimal component of $G \setminus (H \cup M)$ with a set of attachments in H not local.

Output: H' – a larger hyperprism, or

L – a $L(BS(K_4))$

$X \leftarrow$ set of attachments of F in H

```

1  if  $\exists i : X \cap C_i \neq \emptyset$  then
2      relabel strips of  $H$ , so that  $X \cap C_1 \neq \emptyset$ 
3       $x_1 \in X \cap C_1$ 
4      ASSERT:  $X \cap S_2 \neq \emptyset$ 
5       $x_2 \in X \cap S_2$ 
6       $R_1 \leftarrow$  1-rung of  $H$ , so that  $x_1 \in V(R_1)$ 
7       $R_2 \leftarrow$  2-rung of  $H$ , so that  $x_2 \in V(R_2)$ 
8       $R_3 \leftarrow$  a 3-rung of  $H$ 
9       $\forall i \in [3] : a_i, b_i \leftarrow$  ends of  $R_i$ , so that  $a_i \in A_i, b_i \in B_i$ 
10      $K \leftarrow$  a prism  $(R_1, R_2, R_3)$ 
11     ASSERT: no vertex in  $F$  is major w.r.t.  $K$  (as in SPGT 10.5)
12      $f_1 - \dots - f_n \leftarrow$  a minimal path in  $F$ , so that
13          $f_1 \blacktriangleleft \{a_2, a_3\}$ ,
14          $f_n - R \setminus \{a_1\}$ 
15         there are no other edges between  $\{f_1, \dots, f_n\}$  and  $V(K) \setminus \{a_1\}$ 
16     ASSERT:  $F = \{f_1, \dots, f_n\}$ 
17     ASSERT:  $f_1 \blacktriangleleft A_3$ 
18      $A'_1 \leftarrow A_1 \cup \{f_1\}$ 
19      $C'_1 \leftarrow C_1 \cup \{f_2, \dots, f_n\}$ 
20     return  $H' \leftarrow (A'_1, A_2, \dots, B_3, C'_1, C_2, C_3)$ 
21 else
22     relabel strips of  $H$ , so that there is  $\{x_1 \in A_1, x_2 \in A_2\} \subset X$  that is
23     not local
24     find a path  $x - f_1 - \dots - f_n - x_2$ 
25     ASSERT:  $F = \{f_1, \dots, f_n\}$ 
26     if  $n$  is even and  $H$  is even, or  $n$  is odd and  $H$  is odd then
27         ASSERT:  $f_1 - a_3 \vee f_n - b_3$ 
28         if  $f_1 - a_3$  then
29              $H' \leftarrow$  mirrored  $H$  – every  $A_i$  and  $B_i$  are swapped
30             # TODO: check if  $M$  and  $F$  are OK
31             return GROW-HYPERPRISM( $G, H', M, F$ )
32         else
33             if  $f_n \blacktriangleleft B_2 \cup B_3$  then
34                  $B'_1 \leftarrow B_1 \cup \{f_n\}$ 
35                  $C'_1 \leftarrow C_1 \cup \{f_1, \dots, f_{n-1}\}$ 

```

```

21 // else //  $\forall_{i \in [3]} X \cap C_i = \emptyset$ 
22 // if  $n$  is even and  $H$  is even, or  $n$  is odd and  $H$  is odd then
23 // else //  $f_n - b_3$ 
24 // if  $f_n \triangleleft B_2 \cup B_3$  then
25     return  $H' \leftarrow \begin{pmatrix} A_1 & C'_1 & B'_1 \\ A_2 & C'_2 & B'_2 \\ A_3 & C'_3 & B'_3 \end{pmatrix}$ 
26 else
27      $\forall_{i \in [3]} : A'_i \leftarrow \text{neighbors of } f_1 \text{ in } A_i$ 
28      $\forall_{i \in [3]} : A''_i \leftarrow A_i \setminus A'_i$ 
29      $\forall_{i \in [3]} : B'_i \leftarrow \text{neighbors of } f_n \text{ in } B_i$ 
30      $\forall_{i \in [3]} : B''_i \leftarrow B_i \setminus B'_i$ 
31     ASSERT: Every  $i$ -ring is between  $A'_i$  and  $B'_i$  or  $A''_i$  and  $B''_i$ 
32      $\forall_{i \in [3]} : C'_i \leftarrow \text{union of interiors of } i\text{-rings between } A'_i \text{ and } B'_i$ 
33      $\forall_{i \in [3]} : C''_i \leftarrow \text{union of interiors of } i\text{-rings between } A''_i \text{ and } B''_i$ 
34     ASSERT:  $C_i = C'_i \cup C''_i, C'_i \cap C''_i = \emptyset$ 
35     ASSERT:  $A'_i \cup C'_i \sqsupset C''_i \cup B''_i, A''_i \cup C''_i \sqsupset C_i \cup B_i$ 
36     ASSERT:  $A'_i \blacksquare A''_i, B'_i \blacksquare B''_i$ 
37     ASSERT:  $A'_1, A'_2, A'_3, A'_3 \neq \emptyset$ 
38      $H' \leftarrow \begin{pmatrix} A'_1 & C'_1 & B'_1 \\ A'_2 \cup A'_3 & C'_2 \cup C'_3 & B'_2 \cup B'_3 \\ \bigcup_i A''_i \cup \{f_1\} & \bigcup_i C''_i \cup \{f_2, \dots, f_n\} & \bigcup_i B''_i \end{pmatrix}$ 
39     return  $H'$ 
40 else
41      $a_1 \leftarrow \text{neighbor of } f_1 \text{ in } A_1$ 
42      $R_1 \leftarrow \text{1-rung with end } a_1$ 
43      $b_1 \leftarrow \text{the other end of } R_1$ 
44      $b_2 \leftarrow \text{neighbor of } f_2 \text{ in } B_2$ 
45      $R_2 \leftarrow \text{2-rung with end } b_2$ 
46      $a_2 \leftarrow \text{the other end of } R_2$ 
47     ASSERT:  $b_1 \in X, a_2 \in X$ 
48     ASSERT:  $(b_1 - f_1 \wedge a_2 - f_n) \vee (b_1 - f_n \wedge a_2 - f_1)$ 
49     if  $f_1 - b_1$  then
50         ASSERT:  $H$  is odd
51          $R_3 \leftarrow \text{any 3-rung with ends } a_3, b_3, \text{ such that}$ 
52          $\{a_3, b_3\} \sqsupset \{f_1, f_n\}$ 
53         return  $V(R_1) \cup V(R_2) \cup V(R_3) \cup \{f_1, \dots, f_n\}$  - a  $L(BS(K_4))$ 
54         # TODO: Is it valid input for part of ALG I?

```

```

45 // else //  $\forall_{i \in [3]} X \cap C_i = \emptyset$ 
46 // else //  $n$  is odd and  $H$  is even, or  $n$  is even and  $H$  is odd
47 // else //  $f_1 - a_2$ 
48    $\forall_{i \in [3]} : A'_i \leftarrow A_i \cap X, A''_i \leftarrow A_i \setminus X$ 
49    $\forall_{i \in [3]} : B'_i \leftarrow B_i \cap X, B''_i \leftarrow B_i \setminus X$ 
50    $\forall_{i \in [3]} : C'_i \leftarrow$  union of  $i$ -rungs between  $A'_i$  and  $B'_i$ 
51    $\forall_{i \in [3]} : C''_i \leftarrow$  union of  $i$ -rungs between  $A''_i$  and  $B''_i$ 
52   ASSERT:  $C_i = C'_i \cup C''_i, C'_i \cap C''_i = \emptyset$ 
53   if  $f_1$  is complete to at least two of  $A_i$  then
54     relabel strips of  $H$ , so that  $f_1$  is complete to  $A_1$  and  $A_2$ 
55     ASSERT:  $f_n$  is complete to  $B_1$  and  $B_2$ 
56     ASSERT:  $n > 1$  (as in SPGT 10.5 OK for odd  $H$ ?)
57     return  $\begin{pmatrix} A_1 & C_1 & B_1 \\ A_2 & C_2 & B_2 \\ A_3 \cup \{f_1\} & C_3 \cup \{f_2, \dots, f_{n-1}\} & B_3 \cup \{f_n\} \end{pmatrix}$ 
58   else
59     ASSERT:  $A'_i \blacksquare A''_i$  # TODO: OK odd  $H$ ?
60     ASSERT:  $B'_i \blacksquare B''_i$  # TODO: OK odd  $H$ ?
61     return  $\begin{pmatrix} A'_1 & C'_1 & B'_1 \\ A'_2 \cup A'_3 & C'_2 \cup C'_3 & B'_2 \cup B'_3 \\ \bigcup_i A''_i \cup \{f_1\} & \bigcup_i C''_i \cup \{f_2, \dots, f_{n-1}\} & \bigcup_i B''_i \cup \{f_n\} \end{pmatrix}$ 

```

GOOD-PARTITION-FROM-EVEN-HYPERPRISM(G, H, M)
Input: G – square-free, Berge graph containing no $L(BS(K_4))$
 $H = (A_1, \dots, B_3)$ – maximal even hyperprism in G
 M – set of major neighbors of H
Output: A good partition of G

- 1 $Z \leftarrow \bigcup \{V(C) : C \text{ is a component of } G \setminus \{V(H) \cup M\} \text{ with no attachments in } H\}$
- 2 relabel strips of H , so that $M \cup A_1$ and $M \cup B_1$ are cliques
- 3 $F_1 \leftarrow \bigcup \{V(C) : C \text{ is a component of } G \setminus \{H \cup M \cup Z\} \text{ that attaches to } A_1 \cup B_1 \cup C_1\}$
- ASSERT: M is a clique
- ASSERT: $M \cup A_i$ is a clique for at least two values of i
- ASSERT: $M \cup B_j$ is a clique for at least two values of j
- 4 $K_1 \leftarrow A_1, K_2 \leftarrow M, K_3 \leftarrow B_1$
- 5 $R \leftarrow C_1 \cup F_1 \cup Z$
- 6 $L \leftarrow G \setminus \{K_1 \cup K_2 \cup K_3 \cup R\}$
- 7 **return** (K_1, K_2, K_3, L, R)

GOOD-PARTITION-FROM-EVEN-HYPERPRISM(G, H, M)

Input: G – square-free, Berge graph containing no $L(BS(K_4))$
 $H = (A_1, \dots, B_3)$ – maximal odd hyperprism in G
 M – set of major neighbors of H

Output: A good partition of G

- 1 $Z \leftarrow \bigcup \{V(C) : C \text{ is a component of } G \setminus \{V(H) \cup M\} \text{ with no attachments in } H\}$
- 2 relabel strips of H , so that $A_1 \sqsupset B_1$ and $A_2 \sqsupset B_2$
 ASSERT: $C_1 \neq \emptyset, C_2 \neq \emptyset$
- 3 $\forall_{i \in [3]} F_i \leftarrow \bigcup \{V(C) : C \text{ is a component of } G \setminus \{H \cup M \cup Z\} \text{ that attaches to } A_i \cup B_i \cup C_i\}$
- 4 $F_B \leftarrow \bigcup \{V(C) : C \text{ is a component of } G \setminus \{H \cup M \cup Z \cup F_1 \cup F_2 \cup F_3\} \text{ that attaches to } B_1 \cup B_2 \cup B_3\}$
 # **TODO:** F_i, F_A, F_B are from algIV, make sure it is correct
 ASSERT: At least two of A_i and at least two of B_i are cliques
 ASSERT: M is complete to at least two of A_i and at least two of B_i
 ASSERT: M is a clique
 ASSERT: For at least two i : $A_i \cup M$ is a clique
 ASSERT: For at least two j : $A_j \cup M$ is a clique
- 5 choose h , so that $M \cup A_h$ and $M \cup B_h$ are cliques
- 6 **if** $h = 1 \vee h = 2$ **then** // # **TODO:** make sure $h = 2$ is ok
 - 7 $K_1 \leftarrow A_1, K_2 \leftarrow M, K_3 \leftarrow B_1$
 - 8 $R \leftarrow C_1 \cup F_1 \cup Z$
 - 9 $L \leftarrow G \setminus \{K_1 \cup K_2 \cup K_3 \cup R\}$
 - 10 **return** (K_1, K_2, K_3, L, R)
- 11 **else**
 - 12 relabel H so that $M \cup A_1$ and $M \cup B_2$ are cliques
 - 13 $K_1 \leftarrow B_2 \cup B_3, K_2 \leftarrow M, K_3 \leftarrow A_1 \cup A_3$
 - 14 $L \leftarrow B_1 \cup C_1 \cup F_1 \cup F_B$
 - 15 $R \leftarrow G \setminus \{K_1 \cup K_2 \cup K_3 \cup L\}$
 - 16 **return** (K_1, K_2, K_3, L, R)

GOOD-PARTITION-FROM-J-STRIP-SYSTEM($G, J, (S, N), M$)

Input: G – square-free, Berge graph

J – a maximal 3-connected graph with appearance in G

(S, N) – a maximal J -strip system

M – a set of major vertices w.r.t. (S, N)

Output: A good partition of G

2.1 ALG VI

GOOD-PARTITION-FROM-J-STRIP-SYSTEM($G, J, (S, N), M$)

Input: G – square-free, Berge graph

(S, N) – a special K_4 strip system

M – a set of major vertices w.r.t. (S, N)

Output: A good partition of G

```

1  $\forall_{i,j \in [4]} O_{ij} \leftarrow$  set of vertices in  $V(G) \setminus V(S, N)$  that are complete to
    $(N_i \cup N_j) \setminus S_{ij}$  and anticomplete to  $V(S, N) \setminus (N_i \cup N_j \cup S_{ij})$ 
2 if  $(N_1 \setminus N_{12}) \cup M \cup O_{12}$  and  $(N_2 \setminus N_{12}) \cup M \cup O_{12}$  are cliques then
3    $K_1 \leftarrow N_1 \setminus N_{12}, K_2 \leftarrow O_{12} \cup M, K_3 \leftarrow N_2 \setminus N_{12}$ 
4    $L \leftarrow$  union of those components of  $G \setminus (K_1 \cup K_2 \cup K_3)$  that contain
   vertices of  $S_{12}$ 
5    $R \leftarrow G \setminus (K_1 \cup K_2 \cup K_3 \cup L)$ 
6   return  $(K_1, K_2, K_3, L, R)$ 
7 else
8   relabel  $G$  so that  $(N_1 \setminus N_{12}) \cup M \cup O_{12}$  is not a clique
   ASSERT:  $N_{12} \cup M \cup O_{12}$  is a clique
9   if  $N_{21} \cup M \cup O_{12}$  is a clique then
10     $X \leftarrow N_{21}$ 
11  else
12     $X \leftarrow N_2 \setminus N_{21}$ 
13     $K_1 \leftarrow N_{12}, K_2 \leftarrow M \cup O_{12}, K_3 \leftarrow X$ 
14     $L \leftarrow$  component of  $G \setminus (K_1 \cup K_2 \cup K_3)$  that contains  $N_1 \setminus N_{12}$ 
15     $R \leftarrow G \setminus (K_1 \cup K_2 \cup K_3 \cup L)$ 
16    return  $(K_1, K_2, K_3, L, R)$ 
```

FIND-SPECIAL-K4-STRIP-SYSTEM($G, J, (S, N), m$)

Input: G – square-free, Berge graph

J – a 3-connected graph with appearance in G

(S, N) – a J -strip system

$m \in G \setminus (S, N)$ that is major w.r.t. some choice of rungs of (S, N)
but not w.r.t. (S, N)

Output: (S', N') – A special K_4 -strip system, or
 (S'', N'') – a bigger J -strip system

GROWING-J-STRIP($G, J, (S, N)$)

Input: G – square-free, Berge graph

J – a 3-connected graph with appearance in G

(S, N) – a J -strip system

TODO: make sure def of J is correct

Output: J' and a maximal J' -strip system, or a special strip system

1 $M \leftarrow$ vertices of $G \setminus V(S, N)$ that are major on some choice of Rungs of (S, N)

TODO: M like ALGI

2 **if** $\exists m : m$ is not major on some choice of rungs of (S, N) **then**

3 $OUT \leftarrow$ **FIND-SPECIAL-K4-STRIP-SYSTEM**($G, J, (S, N), m$)

4 **if** OUT is a special strip system **then**

5 **return** OUT

6 **else**

7 **return** **GROWING-J-STRIP**(G, OUT)

8 **else if** $\exists F : F$ is a component of $G \setminus (V(S, N) \cup M)$, such that no member of F is major w.r.t. (S, N) and set of attachments of F on H is not local **then**

ASSERT: (as in 6.2, or actually SPGT 8.5)

9 $F \leftarrow$ minimal (component?) with this property

10 **if** $\exists v \in V(J) : X \subset \bigcup (S_{uv} : uv \in E(J))$ **then**

11 $x := X \cap S_{uv} \setminus N_v$, for some $uv \in E(J)$

TODO: brackets?

12 $x' := X \cap S_{u'v}$, for some $u'v \in E(J), u' \neq u$

ASSERT: $\{x, x'\}$ is not local w.r.t. (S, N)

13 $L(H) \leftarrow \forall_{i,j \in E(J)}$ choose ij -rung R_{ij} , so that

$x \in V(R_{uv}), x' \in V(R_{u'v})$

ASSERT: $\{x, x'\}$ is not local w.r.t. $L(H)$

14 $D \leftarrow$ a branch of H with ends d, u :

$\delta_H(d) \setminus E(D) = (X \cap E(H)) \setminus E(D)$

15 $P \leftarrow$ a path with ends p_1, p_2 , so that:

$p_1 \blacktriangleleft N_v \setminus N_{vu}$ and no other vertex of P has neighbors in $N_v \setminus N_{uv}$

$p_2 = x$ and no other vertex of P has neighbors in $S_{uv} \setminus N_v$

16 $(S', N') \leftarrow$ add p_1 to N_v and F to S_{uv}

17 **return** **GROWING-J-STRIP** ($G, J, (S', N')$)

18 **else**

19 $K \leftarrow \{uv \in E(J) : X \cap S_{uv} \neq \emptyset\}$

ASSERT: There are two disjoint edges in K (as in SPGT 8.5.(3))

20 F is a vertex set of a path $\leftarrow f_1 - \dots - f_n$

ASSERT: Every choice of rungs is broad

```

20   // if  $\exists F \dots$  then
    // else //  $\nexists v \in V(J) : X \subset \bigcup (S_{uv} : uv \in E(J))$ 
    ASSERT: every choice of rungs has the same traversal. (Hard to
    assert)
21    $ij \leftarrow$  the traversal edge
22    $A_1 \leftarrow N_i \setminus S_{ij}, A_2 \leftarrow N_j \setminus S_{ij}$ 
    ASSERT:  $X \cap (V(S, N) \setminus S_{ij}) = A_1 \cup A_2$ 
23   if  $n = 1$  then
24      $(S', N') \leftarrow$  add  $f_1$  to  $N_i, N_j, S_{ij}$ 
25     return GROWING-J-STRIP ( $G, J, (S', N')$ )
26   else
27      $x_1 : \in A_1, x_2 : \in A_2$ , so that  $x_1$  and  $x_2$  are in disjoint strips
    ASSERT:  $x_1 - f_1 \vee x_1 - f_n$ 
28     if  $x_1 - f_n$  then
29       relabel  $f_1 - \dots - f_n$  front to back
30      $(S', N') \leftarrow$  add  $f_1$  to  $N_i$ ,  $f_n$  to  $N_j$  and  $F$  to  $S_{ij}$ 
31     return GROWING-J-STRIP ( $G, J, (S', N')$ )
32 else
33   return  $J, (S, N)$  – a maximal  $J$ -strip

```