# Contractions, Removals and Certifying 3-Connectivity in Linear Time 

Technical Report B 10-04

Jens M. Schmidt*<br>Institute of Computer Science<br>Freie Universität Berlin, Germany


#### Abstract

As existence result, it is well known that every 3 -connected graph $G=(V, E)$ on more than 4 vertices admits a sequence of contractions and a sequence of removal operations to $K_{4}$ such that every intermediate graph in the sequences is 3 -connected. We show that both sequences can be computed in linear time, improving the previous best known running time of $O\left(|V|^{2}\right)$ to $O(|V|+|E|)$. This settles also the open question of finding a certifying 3 -connectivity test in linear time and extents to certify 3 -edge-connectivity in linear time as well.


## 1 Construction Sequences

Let $G=(V, E)$ be a finite graph with $n:=|V|$ vertices and $m:=|E|$ edges. Let $G$ be connected if there is a path between every two vertices and disconnected otherwise. For $k \geq 1$, a graph $G$ is $k$-connected if $n>k$ and deleting every set of $k-1$ vertices leaves a connected graph. A vertex (a pair of vertices) that leaves a disconnected graph upon deletion is called a cut vertex (a separation pair). Note that $k$-connectivity does neither depend on parallel edges nor on self-loops. A graph has connectivity $k$ if it is $k$-connected but not $k+1$-connected. Let a path $P$ from vertex $v$ to vertex $w$ in $G$ be denoted by $v \rightarrow_{G} w$ and $s(P)=v$ and $t(P)=w$. For a vertex $v$ in $G$, let $N(v)=\{w \mid v w \in E\}$ denote its set of neighbors and $\operatorname{deg}(v)$ its degree. For a graph $G$, let $\delta(G)$ be the minimum degree of its vertices.

A subdivision of a graph $G$ replaces each edge of $G$ by a path of length at least one. Conversely, we want a notation to get back to the graph without subdivided edges. If $\operatorname{deg}(v)=2,|N(v)|=2$ and $v \notin N(v)$, let $\operatorname{smooth}_{v}(G)$ be the graph obtained from $G$ by deleting $v$ followed by adding an edge between its neighbors (we say $v$ is smoothed). If one of the conditions is violated, we set smooth $_{v}(G)=G$. Let smooth $(G)$ be the graph obtained by smoothing every

[^0]vertex. For an edge $e \in E$, let $G \backslash e$ denote the graph obtained from $G$ by deleting $e$.

Contracting an edge $e=x y$ in a graph deletes $e$, identifies vertices $x$ and $y$ and replaces iteratively two parallel edges by one single edge as long as possible. This way the simpleness of graphs is preserved by contractions. An edge $e$ is called contractible if contracting $e$ results in a 3-connected graph. Let $K_{n}$ be the complete graph on $n$ vertices and $K_{n}^{m}$ be the complete graph on $n$ vertices with $m$ edges between each pair of vertices. For a rooted tree $T$ and $x \in V(T)$, let $T(x)$ be the maximal subtree of $T$ rooted at $x$.

For convenience, we assume the input graph $G$ to be simple, although all results can be extended to multigraphs. The following operations are called Barnette and Grünbaum operations ( $B G$-operations).

- add an edge $x y$ (possibly a parallel edge)
- subdivide an edge $a b$ by a vertex $x$ and add the edge $x y$ for a vertex $y \notin\{a, b\}$
- subdivide two non-parallel edges by vertices $x$ and $y$, respectively, and add the edge $x y$

A sequence of BG-operations that starts with $K_{4}$ and ends with a graph $G$ is called a construction sequence of $G$. Let $G_{4}, G_{5}, \ldots, G_{z}$ with $G_{4}=K_{4}$ and $G_{z}=G$ be the graphs obtained in a construction sequence of $G$. As BGoperations preserve 3 -connectivity and $K_{4}$ is 3-connected, every $G_{j}, 4 \leq j \leq z$ is as well 3 -connected. We can represent the construction sequence in a different, but equivalent way $[1,4]$ : Every $G_{j}$ is identified with a $G_{j}$-subdivision $S_{j}$ in $G$, giving the sequence $S_{4}, \ldots, S_{z}$ of subgraphs in $G$. In particular, $S_{4}$ is a $K_{4^{-}}$ subdivision in $G$. As with contractions, we want the inverse of a BG-operation. Let removing the edge $e=x y$ of a graph be the operation of deleting $e$ followed by smoothing $x$ and $y$. An edge $e=x y$ in $G$ is called removable, if removing $e$ yields a 3 -connected graph.

The vertices $v$ in $S_{j}$ with $\operatorname{deg}(v) \geq 3$ are called real vertices, because they correspond to vertices in $G_{j}$. Let $V_{\text {real }}\left(S_{j}\right)$ be the set of real vertices in $S_{j}$. Note that the graphs $G_{j}$ may contain parallel edges, although every graph $S_{j}$ is simple, because $G$ is simple. We define the links of each $S_{j}$ to be the unique paths in $S_{j}$ with only their end vertices being real. The links of $S_{j}$ partition $E\left(S_{j}\right)$, as $S_{j}$ is simple, 2-connected and not a cycle and therefore two vertices of degree at least 3 must exist [6, Lemma 12.11]. Let two links be parallel if they share the same end vertices.

Definition 1. A $B G$-path for a subgraph $S_{j} \subset G$ is a path $P=x \rightarrow_{G} y$ with the following properties:

1. $S_{j} \cap P=\{x, y\}$
2. Every link of $S_{j}$ that contains $x$ and $y$, contains them as end vertices.
3. If $x$ and $y$ are inner vertices of links $L_{x}$ and $L_{y}$ of $S_{j}$, respectively, and $\left|V_{\text {real }}\left(S_{j}\right)\right| \geq 4$, then $L_{x}$ and $L_{y}$ are not parallel.

It is easy to see that every BG-path for a subdivision $S_{j}$ corresponds to a BG-operation on $G_{j}$ and vice versa. The choice of the $K_{4}$-subdivision $S_{4}$ is not
crucial [4]: At the expense of having additional parallel edges in intermediate graphs $G_{j} \subset G$, there exists a construction sequence to $G$ from each prescribed $K_{4}$-subdivision in $G$.

Theorem 2. The following statements are equivalent:

$$
\begin{align*}
& \text { A simple graph } G \text { is } 3 \text {-connected }  \tag{1}\\
& \Leftrightarrow\left.\exists \text { sequence of } B G \text {-operations from } K_{4} \text { to } G \text { (see }[1, ~ 6]\right)  \tag{2}\\
& \Leftrightarrow \exists \text { sequence of } B G \text {-paths from each } K_{4} \text {-subdivision in } G \text { to } G \text { and } \\
& \delta(G) \geq 3 \text { (see [4]) }  \tag{3}\\
& \Leftrightarrow \exists \text { sequence of removals from } G \text { to } K_{4} \text { on removable edges } e=x y \\
& \text { with }|N(x)| \geq 3,|N(y)| \geq 3 \text { and }|N(x) \cup N(y)| \geq 5 \text { (see [4]) }  \tag{4}\\
& \Leftrightarrow \exists \text { sequence of contractions from } G \text { to } K_{4} \text { on contractible edges } e=x y \\
& \text { with }|N(x)| \geq 3 \text { and }|N(y)| \geq 3 \text { (see [5]) } \tag{5}
\end{align*}
$$

Sequences 2.(2) and 2.(3) can be transformed into each other with a simple linear time algorithm [4]. Moreover, having sequence 2.(2) gives immediately sequence 2.(4) in linear time by removing the edges that were added as BGoperations in reverse order. The following result shows that sequences based on Barnette and Grünbaums' characterization are algorithmically at least as powerful as the well-known sequence of contractions and all other sequences of Theorem 2.

Lemma 3 ([4]). There is a simple algorithm that transforms a given sequence of type 2.(2) or 2.(3) to a sequence of type 2.(4) and to a sequence of type 2.(5) in linear time.

This allows us to focus only on computing a sequence of type 2.(3). To construct such a sequence, we will use the following Lemma for the special case that $H$ is a $K_{4}$-subdivision.

Lemma 4 ([4]). Let $G$ be a 3 -connected graph and $H \subset G$ with $H$ being a subdivision of a 3 -connected graph. Then there is a $B G$-path for $H$ in $G$. Moreover, every link of $H$ of length at least 2 contains an inner vertex on which a $B G$-path for $H$ starts.

## 2 Chain Decomposition

Let $G$ be the simple, 3 -connected input graph. According to Lemma 4, it suffices to add iteratively BG-paths to a $K_{4}$-subdivision in $G$ to get sequence 2.(3). Note that we cannot make a wrong decision when choosing one BG-path in the construction sequence, since Lemma 4 can be applied on the new subdivision as well and therefore ensures a completion of the sequence. Instead of starting with a $K_{4}$-subdivision, we will w.l.o.g. start with a $K_{2}^{3}$-subdivision $S_{3}$ and demand that there is a BG-path for $S_{3}$ that results in a $K_{4}$-subdivision. We now construct $S_{3}$ in $G$ and then describe a decomposition of $G$ into special paths that allow us to find the BG-paths for each $S_{i}, 3 \leq j \leq z$, efficiently.

An arbitrary Depth First Search (DFS) is performed on $G$, assigning a Depth First Index (DFI) to every vertex. Let $T$ be the DFS-tree obtained, $r$ be the
root of $T$ and $u$ be the vertex that is visited second in the DFS. Both, $r$ and $u$, have exactly one child, as otherwise they would form a separation pair in $G$. For two vertices $v$ and $w$ in $T$, let $v$ be a (proper) ancestor of $w$ and $w$ be a (proper) descendant of $v$ if $v \in w \rightarrow_{T} r$ (and $\left.v \neq w\right)$. A backedge is an oriented edge $v w \in E(G) \backslash E(T)$ from $v$ to $w$ with $v$ being an ancestor of $w$ in $T$ (note that the orientation differs from standard notation). A backedge $v w$ is entering a subtree $T^{\prime}$ of a tree if $v \notin V\left(T^{\prime}\right)$ but $w \in V\left(T^{\prime}\right)$.

We choose two backedges $r a$ and $r b$ and denote the least common ancestor of $a$ and $b$ in $T$ with $x$. The paths $x \rightarrow_{T} r, r a \cup a \rightarrow_{T} x$ and $r b \cup b \rightarrow_{T} x$ are the three subdivided edges of $S_{3}$ in $G$ having real vertices $r$ and $x$.

Dependent on $T$, we describe the decomposition of $G$ into special paths $C:=\left\{C_{0}, C_{1}, \ldots, C_{m-n+1}\right\}$, called chains, whose edge sets partition $E(G)$. Additionally, the decomposition yields a total order $<$ on $C$ with $C_{0}<C_{1}<$ $\ldots<C_{m-n+1}$ that reflects the order in which the chains were computed. Let $\operatorname{low}(P)$ for a path $P$ in $G$ be the vertex in $P$ with maximal DFI value. We set $C_{0}:=x \rightarrow_{T} r, C_{1}:=r a \cup a \rightarrow_{T} x$ and $C_{2}:=r b \cup b \rightarrow_{T} x$. The remaining chains are obtained by applying the following procedure subsequently for $i=$ $0,1, \ldots, n-m+1$ : We iterate over all vertices $v \in V\left(C_{i}\right)$ from $t\left(C_{i}\right)$ to $\operatorname{low}\left(C_{i}\right)$. For every backedge $v w$ that is not already in a chain, we traverse the path $w \rightarrow_{T} r$ until a vertex $x$ is found that is contained in a chain. The traversed path including $v w$ forms the new chain $C_{j}=v \rightarrow_{G} x$, with $j$ being the next free index.

Note that every chain $C_{i} \neq C_{0}$ contains exactly one backedge and that $s\left(C_{i}\right)$ is always a proper ancestor of $t\left(C_{i}\right)$. For an edge $e \in E(G)$, let $C(e)$ be the chain that contains $e$. The next two lemmas show that $C$ partitions the edge set of $G$ and admits a tree structure.

Lemma 5. The edge sets of the chains in $C$ partition $E(G)$, hence $|C|=$ $m-n+2$. Let $e=e_{1} e_{2}$ be an edge in $T$ and $f=f_{1} f_{2}$ be an edge in $G$ with $e_{2}=\operatorname{low}(e), f_{2}=\operatorname{low}(f)$ and $e_{2}$ being an ancestor of $f_{2}$ in $T$. Then $C(e) \leq C(f)$ holds.

Proof. Whenever a vertex $v$ is processed in the chain decomposition, all its ancestors and their attached backedges have already been processed. At the same time, for every child $w$ of $v$, the tree edge $v w$ is already contained in a chain, since otherwise all backedges entering $T(w)$ have to start at $v$, implying that $v$ is a cut vertex separating $T(w)$. Therefore, $w$ will be processed later. This implies that every vertex and every edge in $G$ will be processed by the decomposition. Since all chains are pairwise edge-disjoint, the edge sets of chains in $C$ partition $E(G)$. Since every chain in $C \backslash\left\{C_{0}\right\}$ contains exactly one backedge, $C$ contains $m-n+2$ chains.

The same line of argument settles the second claim for the case when $e_{1}$ is an ancestor of $s(C(f))$, since then $e$ is already contained in a chain at the time $s(C(f))$ is processed and $C(e)<C(f)$ holds. The only remaining case is when $s(C(f))$ is an ancestor of $e_{1}$ with $s(C(f)) \neq e_{1}$. If $e$ is not part of a chain when $C(f)$ is found, $e$ will be contained in $C(f)$, because $e_{2}$ is an ancestor of $f_{2}$, and $C(e)=C(f)$. Otherwise, $e$ is part of a chain that was found before $C(f)$ and $C(e)<C(f)$ holds.

Definition 6. Let the parent of a chain $C_{i} \neq C_{0}$ be the chain that contains the edge from $t\left(C_{i}\right)$ to the parent of $t\left(C_{i}\right)$ in $T$.

Lemma 7. The parent relation defines a tree $U$ with $V(U)=C$ and root $C_{0}$.
Proof. Let $D_{0} \neq C_{0}$ be a chain in $C$ and let $D_{1}, \ldots, D_{k}$ be the sequence of chains containing the edges of $t\left(D_{0}\right) \rightarrow_{T} r$ in that order, omitting double occurrences. By definition of the parent relation, each $D_{i}, 0 \leq i<k$, has the parent $D_{i+1}$ and $D_{k}=C_{0}$ holds. It follows that $U$ is connected. Moreover, for each parent $C_{i}$ of a chain $C_{j}$ holds $C_{i}<C_{j}$ with Lemma 5 and, thus, no cycle can occur in $U$.

### 2.1 Classifying Chains

In order to find BG-paths efficiently, we extend the chain decomposition by assigning each chain in $C \backslash\left\{C_{0}\right\}$ immediately after it was found to one of five types: $1,2 a+b$ and $3 a+b$. These types form a partition of $C \backslash\left\{C_{0}\right\}$ and are defined by Algorithm 1. Note that the classification for $C_{i}$ is dependent on the type of the parent $C_{k}$ of $C_{i}$. At the beginning, all chains are unmarked.

```
Algorithm 1 classify \(\left(C_{i} \in C \backslash\left\{C_{0}\right\}\right.\), DFS-tree \(\left.T\right)\)
    \(C_{k}:=\operatorname{parent}\left(C_{i}\right) \quad \triangleright\) the parent of \(C_{i}\) in \(U: C_{k}<C_{i}\)
    if \(t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right)\) is contained in \(C_{k}\) then \(\quad \triangleright\) type 1
        assign \(C_{i}\) to type 1
    else if \(s\left(C_{i}\right)=s\left(C_{k}\right)\) then \(\triangleright\) type 2: \(C_{k} \neq C_{0}, t\left(C_{i}\right)\) is inner vertex of \(C_{k}\)
        if \(C_{i}\) is a backedge then
            assign \(C_{i}\) to type \(2 a \quad \triangleright\) type \(2 a\)
        else
            assign \(C_{i}\) to type \(2 b\); mark \(C_{i} \quad \triangleright\) type \(2 b\)
    else \(\triangleright\) type 3: \(s\left(C_{i}\right) \neq s\left(C_{k}\right), C_{k} \neq C_{0}, t\left(C_{i}\right)\) is inner vertex of \(C_{k}\)
        if \(C_{k}\) is not marked then
            assign \(C_{i}\) to type \(3 a \quad \triangleright\) type \(3 a\)
        else \(\quad \triangleright C_{k}\) is marked
            assign \(C_{i}\) to type \(3 b\); create a list \(L_{i}=\left\{C_{i}\right\} ; C_{j}:=C_{k} \quad \triangleright\) type \(3 b\)
            while \(C_{j}\) is marked do \(\quad \triangleright\) the chains in \(L_{i}\) form a caterpillar
                    unmark \(C_{j}\); append \(C_{j}\) to \(L_{i} ; C_{j}:=\operatorname{parent}\left(C_{j}\right)\)
```

Lemma 8. Computing a chain decomposition of a 3-connected graph and classifying each chain with Algorithm 1 takes running time $O(n+m)$.

Proof. The DFS tree $T$ can be obtained in time $O(n+m)$. The subdivision $S_{3}$ can be found in time linearly dependent on $E\left(S_{3}\right)$ by taking two arbitrary backedges $r a$ and $r b$ with $r$ being the root of $T$ and finding the lowest common ancestor of $a$ and $b$ by traversing $T$ upwards. The computation of each remaining chain $C_{i}, i>2$, takes time linearly dependent on its length, too, which gives a running time of $O(n+m)$ for the chain decomposition.

In order to obtain a fast classification in Algorithm 1, we store the following information on each chain $C_{i}$ : A pointer to its parent $C_{k}$ (for $C_{i} \neq C_{0}$ ), pointers
to $s\left(C_{i}\right)$ and $t\left(C_{i}\right)$ and the information whether $C_{i}$ is a backedge. In addition, for each inner vertex of $C_{i}$ a pointer to $C_{i}$ is stored. That allows us to check vertices on being contained as inner vertices or end vertices in arbitrary chains in $O(1)$. If $C_{k}=C_{0}$, we can check the condition on $C_{i}$ being of type 1 in constant time by testing whether $s\left(C_{i}\right)$ and $t\left(C_{i}\right)$ are contained in $C_{0}$. If $C_{k} \neq C_{0}$, we check in constant time whether $s\left(C_{i}\right)$ and $t\left(C_{i}\right)$ are contained in $C_{k} \backslash\left\{s\left(C_{k}\right)\right\}$. The condition for type 2 needs constant time as well. Every chain is marked at most once, therefore unmarked as most once in line 15 of Algorithm 1, which gives a total running time of $O(n+m)$.

### 2.2 Restrictions

Definition 9. Let a subdivision $S_{j} \subseteq G$ be upwards-closed if for each vertex in $S_{j}$ the edge to its parent is in $E\left(S_{j}\right)$. Let $S_{j}$ be modular if $S_{j}$ is the union of chains.

In order to find BG-paths efficiently, we want to restrict every subdivision $S_{l}$ to be upwards-closed and modular.

Lemma 10. Let $S_{l}$ be upwards-closed and modular. Then a $B G$-path $P$ for $S_{l}$ is a chain if and only if $S_{l+1}$ is upwards-closed and modular.

Proof. If $P$ is a chain, $t(P)$ is contained in $S_{l}$ and $S_{l+1}$ must be upwards-closed and modular due to the DFS structure. If $P$ is not a chain, we assume to the contrary that $S_{l+1}$ is upwards-closed and modular. Then $P$ must be the union of $t>1$ chains; let $C_{i}$ be the first chain in $P$. Now $P$ cannot start with $t\left(C_{i}\right)$, since $s\left(C_{i}\right)$ is in $S_{l}$ and property 1.1 contradicts $t>1$. Thus, $P$ starts with $s\left(C_{i}\right)$, which contradicts $t>1$ as well, as $S_{l+1}$ is upwards-closed and a second chain in $P$ would include another backedge in $P$ at a vertex that is already incident to two DFS tree edges.

Lemma 10 shows that this restriction implies every BG-path to be a chain. Unfortunately, configurations exist where no BG-path for a subdivision $S_{l}$ is a chain (show an example here) and we have to weaken the restriction in order to ensure the existence of a construction sequence. Instead of forcing every subdivision to be upwards-closed and modular, we will only assume that the current subdivision $S_{l}$ has that properties and then find $t$ BG-paths that result in an upwards-closed subdivision when applied successively and whose union is the union of $t$ distinct chains. This ensures $S_{l+t}$ to be modular and implies $z=|C|=m-n+2$. Since $S_{3}$ is upwards-closed and modular, we now can assume that whenever we are searching for a new BG-path, the current subdivision $S_{l}$ is upwards-closed, modular and consists of exactly $l$ chains (and at least $l$ links).

We impose the additional restriction $\left(R_{2}\right)$ (see Figure 1), which will prevent some BG-paths in the sequence from violating property 1.3. In total, we obey the following restrictions in a construction sequence:
$\left(R_{1}\right)$ For each subdivision $S_{l}$, BG-paths are only added as

- single chains of type $1,2 a$ or $3 a$, respectively, with $S_{l+1}$ being upwardsclosed and modular or as

(a) allowed

(b) forbidden

Figure 1: The effect of restriction $\left(R_{2}\right)$ on the dashed BG-path.

- set of $t>1$ successive BG-paths that construct an upwards-closed modular subdivision $S_{l+t}$ differing from $S_{l}$ in exactly $t$ chains of types $2 b$ and $3 b$.
$\left(R_{2}\right)$ For each subdivision $S_{l}$, every link of $S_{l}$ that contains only tree edges of $T$ has no parallel link, except $C_{0}$ in $S_{3}$.

We will prove the existence of a construction sequence restricted by $\left(R_{1}\right)$ and $\left(R_{2}\right)$ in Section 3. Now it is shown that restriction $\left(R_{2}\right)$ implies property 1.3 on upwards-closed modular subdivisions.

Lemma 11. Let $S_{l}$ be a subdivision constructed under ( $R_{1}$ ) and ( $R_{2}$ ) (in particular, $S_{l}$ is upwards-closed and modular). Then each path $P$ for $S_{l}$ having properties 1.1 and 1.2 is a $B G$-path. If $P$ is additionally a chain of type $2 a$ or $3 a,\left(R_{1}\right)$ and $\left(R_{2}\right)$ are preserved.

Proof. For the first claim, assume to the contrary that $P$ violates property 1.3. Then $\left|V_{\text {real }}\left(S_{i}\right)\right| \geq 4$ must hold and $S_{l} \neq S_{3}$ follows. Let $R$ and $Q$ be the parallel links of $S_{l}$ that contain the end vertices of $P$ as inner vertices, respectively. At least one of them, say $R$, contains a backedge, since otherwise $T$ would contain a cycle. Let $C_{i} \neq C_{0}$ be the chain in $S_{l}$ that contains $R$. Since $C_{i}$ contains exactly one backedge, $s\left(C_{i}\right)$ is an end vertex of $R$. If $R \subset C_{i}, Q$ must contain a backedge, as $t\left(C_{i}\right)$ is an inner real vertex of $t(R) \rightarrow_{T} s(R)$. In that case, all inner vertices of $Q$ lie in a subtree of $T$ that cannot be reached by $P$ due to property 1.1 and $S_{l}$ being upwards-closed. Thus, $R=C_{i}$ and with the same argument $Q=t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right)$ holds. With $\left(R_{2}\right), S_{l}$ must be $S_{3}$ and $Q=C_{0}$, which contradicts our assumption.

For the second claim, each chain $C_{i}$ of type 2 or 3 has by definition an inner real vertex in $t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right)$ and therefore preserves $\left(R_{2}\right)$. If $C_{i}$ is of type $2 a$ or $3 a,\left(R_{1}\right)$ is preserved as well, as $S_{l+1}$ is upwards-closed and modular with Lemma 10.

We now show that the chains of type $3 a$ help to find BG-paths efficiently.
Lemma 12. Let $C_{k}$ be the parent of a chain $C_{i} \neq C_{0}$.

- If $C_{i}$ is not of type $1, C_{k} \neq C_{0}$ and $t\left(C_{i}\right)$ is an inner vertex of $C_{k}$.


Figure 2: A chain $C_{i} \nsubseteq S_{l}$ of type 3.

- Let $C_{k}$ but not $C_{i}$ be contained in a subdivision $S_{l}$ constructed under $\left(R_{1}\right)$ and $\left(R_{2}\right)$. If $C_{i}$ is either of type 3 a or of type 1 with an inner real vertex in $t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right), C_{i}$ is a BG-path for $S_{l}$ preserving $\left(R_{1}\right)$ and $\left(R_{2}\right)$.

Proof. Assume to the contrary that $C_{i}$ is not of type 1 and $C_{k}=C_{0}$. Because $t\left(C_{i}\right)$ is contained in $C_{0}, s\left(C_{i}\right)$ must be in $C_{0}$ as well. But then $C_{i}$ would be of type 1 , since $t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right) \subseteq C_{0}$. Therefore, if $C_{i}$ is not of type $1, C_{k} \neq C_{0}$ holds and $C_{k}$ must start with a backedge. Then the definition of the parent relation implies that $t\left(C_{i}\right)$ is an inner vertex of $C_{k}$.

For the second claim, let $C_{i}$ first be of type $3 a$. Since $S_{l}$ is upwards-closed, modular and contains $C_{k}, C_{i}$ satisfies the property 1.1 of BG-paths. In addition, $s\left(C_{i}\right) \neq s\left(C_{k}\right)$ holds by definition and with $C_{k}<C_{i}, s\left(C_{i}\right)$ must be an inner vertex of the path $t\left(C_{k}\right) \rightarrow_{T} s\left(C_{k}\right)$ (see Figure 2). Therefore, the only chains $C_{j}$ that contain $s\left(C_{i}\right)$ and $t\left(C_{i}\right)$ are different from $C_{0}$ and fulfill $C_{i} \cap C_{j}=$ $\left\{s\left(C_{i}\right), t\left(C_{i}\right)\right\}=\left\{s\left(C_{j}\right), t\left(C_{j}\right)\right\}$. This implies $C_{i}$ having property 1.2. Using Lemma 11, $C_{i}$ is a BG-path for $S_{l}$ that preserves $\left(R_{1}\right)$ and $\left(R_{2}\right)$.

If $C_{i}$ is of type 1 , property 1.1 follows from the same argument as before. Additionally, the inner real vertex in $t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right)$ prevents any link containing $s\left(C_{i}\right)$ and $t\left(C_{i}\right)$ from having $s\left(C_{i}\right)$ or $t\left(C_{i}\right)$ as an inner vertex and therefore ensures property 1.2. Lemma 11 implies that $C_{i}$ is a BG-path for $S_{l}$ and $C_{i}$ must preserve $\left(R_{1}\right)$ and ( $R_{2}$ ), the latter due to the inner real vertex in $t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right)$.

### 2.3 Caterpillars

While chains of type $3 a$ form BG-operations under the conditions of Lemma 12, chains of types 1 and 2 in general do not. For every chain $C_{i}$ of type $3 b$, Algorithm 1 collects a list $L_{i}$ that contains only $C_{i}$ and chains of type $2 b$ (see line 15). We call each list $L_{i}$ a caterpillar. Caterpillars bundle the single chains of type $2 b$, which cannot immediately be added as BG-paths, and will offer a simple decomposition into successive BG-paths later.

Proposition 13. Every caterpillar $L_{i}$ contains exactly one chain of type $3 b$, namely the chain $C_{i}$. All other chains in $L_{i}$ are of type $2 b$.

Lemma 14. $C \backslash\left\{C_{0}\right\}$ is partitioned into the chains of types $1,2 a, 3 a$ and the chains being contained in caterpillars. Moreover, no chain is contained in two caterpillars.

Proof. With Proposition 13, it remains to show that every chain $C_{i}$ of type $2 b$ or $3 b$ is contained in exactly one caterpillar. If $C_{i}$ is of type $3 b, C_{i}$ is part of the caterpillar $L_{i}$ (see Algorithm 1, line 13) and will not be assigned to a second caterpillar afterwards, as it is not marked. Otherwise, $C_{i}$ is of type $2 b$ and was therefore marked. We show that, after all chains in $C$ have been classified, $C_{i}$ is not marked anymore. This forces $C_{i}$ to be contained in exactly one caterpillar, as the only way to unmark chains is to assign them to a caterpillar (see Algorithm 1, line 15) and no chain is marked twice.

Let $C_{k}$ be the parent of $C_{i}$. Because $C_{i}$ is of type $2 b, s\left(C_{i}\right)=s\left(C_{k}\right)$ holds and $C_{i}$ is not a backedge, implying that the last edge $e$ of $C_{i}$ is in $T$. Let $x$ be the end vertex of $e$ different from $t\left(C_{i}\right)$. Using Lemma $12, C_{k} \neq C_{0}$ holds and $t\left(C_{i}\right)$ is an inner vertex of $C_{k}$. Then at least one backedge $v w$ with $v \notin\left\{s\left(C_{i}\right), t\left(C_{i}\right)\right\}$ must enter $T(x)$, since otherwise $s\left(C_{i}\right)$ and $t\left(C_{i}\right)$ would be a separation pair of $G$. Let $C_{j}$ be the minimal chain with respect to $<$ that contains such a backedge.

As $C_{j}>C_{i}$ holds due to Lemma 5 , the vertex $v$ is an inner vertex of $t\left(C_{i}\right) \rightarrow_{T} s\left(C_{i}\right)$, implying that $C_{j}$ is not of type 2 . In addition, $C_{j}$ is not of type 1, since $t\left(C_{j}\right) \rightarrow_{T} v$ contains edges from $C_{i}$ and $C_{k}$. At the time $C_{j}$ is found in the chain decomposition, every chain that already ends at a vertex in $T(x)$ starts at $s\left(C_{i}\right)$ and is therefore of type $2 a$ or $2 b$. Since chains that are backedges cannot have children, the parent of $C_{j}$ is marked and $C_{j}$ is of type $3 b$. Moreover, every chain corresponding to an inner vertex of the path $C_{j} \rightarrow_{U} C_{i}$ is marked. This concludes $C_{i}$ to become unmarked due to line 15 of Algorithm 1.

Definition 15. Let the parent of a caterpillar $L_{j}$ be the parent of the chain in $L_{j}$ that is minimal with respect to $<$. Let a caterpillar $L_{j}$ with parent $C_{k}$ be bad if $s\left(C_{j}\right)$ is a descendant of $t\left(C_{k}\right)$ and $s\left(C_{k}\right) \rightarrow_{C_{k}} s\left(C_{j}\right)$ contains no inner real vertex (see Figure 3(a)). Otherwise, $L_{j}$ is called a good caterpillar (see Figure 3(b)).

We now show that, under some minor conditions, caterpillars can be decomposed into multiple BG-paths nicely.

Lemma 16. Let $L_{j}$ be a caterpillar that contains $t$ chains and has parent $C_{k}$. Let $C_{k}$ but no chain in $L_{j}$ be contained in a subdivision $S_{l}$ that was constructed under ( $R_{1}$ ) and ( $R_{2}$ ). Then the chains in $L_{j}$ can be efficiently decomposed into $t$ successive $B G$-paths preserving $\left(R_{1}\right)$ and creating subdivisions $S_{l+1}, S_{l+2}, \ldots, S_{l+t}$, each of which satisfies $\left(R_{2}\right)$, if and only if $L_{j}$ is good.

Proof. Let $L_{j}$ be good and let $y$ be the last vertex of the minimal chain in $L_{j}$, thus $y \in V\left(C_{k}\right)$. We assume at first that $s\left(C_{j}\right)$ is a proper ancestor of $t\left(C_{k}\right)$ (see Figure 3(b)). Then the path $P=C_{j} \cup\left(t\left(C_{j}\right) \rightarrow_{T} y\right)$ fulfills properties 1.1 and 1.2 and is a BG-path for $S_{l}$ with Lemma 11. Adding $P$ preserves $S_{l}$ to be upwards-closed but not modular. The restriction $\left(R_{2}\right)$ is also preserved, as $t\left(C_{k}\right)$ is real and, for $S_{l}=S_{3}, C_{k}$ must be either $C_{1}$ or $C_{2}$, implying that $s(P)$ becomes


Figure 3: Two kinds of caterpillars.
an inner real vertex of $C_{0}$. Successively, for each chain $C_{i}$ of the $t-1$ chains in $L_{i} \backslash\left\{C_{j}\right\}$, we now add $C_{i} \backslash P$, which is a BG-path yielding an upwards-closed subdivision for analogue reasons.

Now assume that $s\left(C_{j}\right)$ is a descendant of $t\left(C_{k}\right)$ (see Figure $3(\mathrm{a})$ ). Then $s\left(C_{j}\right) \in V\left(C_{k}\right)$ and since $L_{j}$ is good, there is a real vertex a strictly between $s\left(C_{j}\right)$ and $s\left(C_{k}\right)$ in $C_{k}$. We first show that $t\left(C_{k}\right) \rightarrow_{T} s\left(C_{k}\right)$ contains an inner real vertex as well. Assume the contrary. Then $C_{k}$ must be of type 1 and has been added before, contradicting restriction $\left(R_{2}\right)$ unless $S_{l}=S_{3}$. But $S_{l}$ must be different from $S_{3}$, since $a$ exists, and it follows that $t\left(C_{k}\right) \rightarrow_{T} s\left(C_{k}\right)$ contains an inner real vertex $b$. Let $C_{h}$ be the parent of $C_{j}$. Then $\left(C_{j} \cup C_{h}\right) \backslash\left(\left(t\left(C_{j}\right) \rightarrow_{T}\right.\right.$ $\left.y) \backslash\left\{t\left(C_{j}\right)\right\}\right)$ is a BG-path due to the real vertices $a$ and $b$ and we add it, although it neither preserves $S_{l+1}$ to be upwards-closed nor modular. We next add $t\left(C_{j}\right) \rightarrow_{T} y$, which restores upwards-closedness. The resulting subdivisions $S_{l+1}$ and $S_{l+2}$ both satisfy $\left(R_{2}\right)$, as $b$ is real in $S_{l+1}$ and $S_{l+2}$ and $y$ is real in $S_{l+2}$. We proceed with adding successively paths, namely for each chain $C_{i}$ of the $t-2$ remaining chains in $L_{i} \backslash\left\{C_{j}, C_{h}\right\}$ the path $C_{i} \backslash\left(t\left(C_{j}\right) \rightarrow_{T} y\right)$. With the same line of argument, these paths obtain upwards-closed subdivisions $S_{l+3}, \ldots, S_{l+t}$, each of which satisfies $\left(R_{2}\right)$.

In both cases, $S_{l+t}$ is modular, since $L_{j}$ is a list of chains. Moreover, the $t$ chosen BG-paths preserve $\left(R_{1}\right)$, as the chains in $L_{j}$ are of types $2 b$ and $3 b$ only, $t>1$ holds and $S_{l+t}$ is upwards-closed. All paths can be computed in time linearly dependent on the total number of edges in $L_{j}$.

For the only if part, let $P_{1}$ and $P_{2}$ be the first two BG-paths in a decomposition of the chains in $L_{j}$; these exist, since $t>1$ holds in every caterpillar. Let $L_{j}$ be bad, as otherwise the claim follows. Then $s\left(C_{j}\right) \in V\left(C_{k}\right)$. We show that $L_{j}$ cannot be bad, as $S_{l}$ contains a real vertex in $C_{k}$ strictly between $s\left(C_{j}\right)$ and $s\left(C_{k}\right)$. Because of properties 1.1 and 1.2, $P_{1} \cap S_{l}=\left\{s\left(C_{k}\right), s\left(C_{j}\right)\right\}$ must hold and $P_{1}$ is a link of $S_{l+1}$ being parallel to $s\left(C_{j}\right) \rightarrow_{C_{k}} s\left(C_{k}\right)$. Since only the chain of type $3 b$ in $L_{j}$ starts at $s\left(C_{j}\right)$, both end vertices of $P_{2}$ must be different from $s\left(C_{j}\right)$. Then, due to properties 1.1 and $1.2, P_{2}$ joins inner vertices of the parallel links $P_{1}$ and $s\left(C_{j}\right) \rightarrow_{C_{k}} s\left(C_{k}\right)$ in $S_{l+1}$, contradicting property 1.3, as
$\left|V_{\text {real }}\left(S_{l+1}\right)\right| \geq 4$.

## 3 Existence of the Restricted Construction Sequence

We show that there still exists a construction sequence under restrictions $\left(R_{1}\right)$ and $\left(R_{2}\right)$.

Definition 17. Let $S_{l}$ be a subdivision of $G$ constructed under $\left(R_{1}\right)$ and $\left(R_{2}\right)$. We define the equivalence relation $\sim$ on $E(G) \backslash E\left(S_{l}\right)$ with

- $\forall e \in E(G) \backslash E\left(S_{l}\right): e \sim e$ and
- $\forall e, f \in E(G) \backslash E\left(S_{l}\right): e \sim f$ if there is a path $e \rightarrow_{G} f$ without an inner vertex in $S_{l}$.

Let the equivalence classes of $\sim$ be the segments of $S_{l}$. Note that every segment consists of a disjoint union of chains. For a chain $C_{i}$ that is not contained in $S_{l}$, let the segment of $C_{i}$ be the segment of $S_{l}$ that contains $C_{i}$.

Lemma 18. Let $S_{l}$ be a subdivision constructed under $\left(R_{1}\right)$ and $\left(R_{2}\right)$ and let $C_{i}$ be a chain of type 3 such that $s\left(C_{i}\right) \in V\left(S_{l}\right), C_{i} \nsubseteq S_{l}$ and $C_{i}$ is minimal among the chains of type 3 in its segment $H$. Let $D_{1}>\ldots>D_{k}$ be all ancestors of $C_{i}$ in $H$ with $D_{1}=C_{i}$ and $D_{k}$ being the minimal chain in $H$. Then all chains $D_{1}, \ldots, D_{k}$ can be successively added as $B G$-paths preserving ( $R_{1}$ ) and ( $R_{2}$ ) (possibly being part of caterpillars), unless one of the following exceptions holds:

1. $C_{i}$ is of type $3 a, k=2, D_{k}$ is of type $1, s\left(C_{i}\right)$ is an inner vertex of $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$ and there is no inner real vertex in $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$ (Figure $4(a)$ ),
2. $C_{i}$ is of type $3 b, D_{k}$ is of type $2 b$ and $L_{i}=\left\{D_{1}, \ldots, D_{k}\right\}$ with $L_{i}$ being bad (Figure 4(b)),
3. $C_{i}$ is of type $3 b, L_{i}=\left\{D_{1}, \ldots, D_{k-1}\right\}, D_{k}$ is of type $1, s\left(C_{i}\right)$ is an inner vertex of $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$ and there is no inner real vertex in $t\left(D_{k}\right) \rightarrow_{T}$ $s\left(D_{k}\right)$ (Figure $4(c)$ ).

Proof. Let $D \in\left\{D_{2}, \ldots, D_{k}\right\}$. Then $D$ is not of type 3 by assumption and not of type $2 a$, as chains of that type cannot have children. Assume that $D$ is of type $2 b$ and let $L_{j}$ be the caterpillar containing $D$ due to Lemma 14. If $C_{j} \neq C_{i}, C_{j}<C_{i}$ holds, as otherwise $C_{j}$ would not be the chain of type $3 b$ in $L_{j}$. But then $C_{j}$ contradicts the minimality of $C_{i}$, since $C_{j}$ is not contained in $S_{l}$ and of type $3 b$. We conclude that every chain in $\left\{D_{2}, \ldots, D_{k}\right\}$ of type $2 b$ is contained in $L_{i}$ and forces $C_{i}$ to be of type $3 b$. This is used in the following case distinction.

Let $C_{i}$ be of type $3 a$. If $k=1, C_{i}$ is a BG-path for $S_{l}$ with Lemma 12 and the claim follows. Otherwise, $k>1$ and all chains in $\left\{D_{2}, \ldots, D_{k}\right\}$ are of type 1. Then $s\left(D_{2}\right)$ is a proper ancestor of $s\left(C_{i}\right)$, since $D_{2}<C_{i}$ and $C_{i}$ is not of type 2. Moreover, $s\left(C_{i}\right)$ is a proper ancestor of $t\left(D_{2}\right)$, because otherwise


Figure 4: The three exceptions of Lemma 18. The black vertices in 18.1 and 18.3 can also be non-real.
$H \cap S_{l}=\left\{s\left(D_{2}\right), t\left(D_{2}\right)\right\}$ is a separation pair of $G$ due to the minimality of $C_{i}$. It follows that $s\left(C_{i}\right)$ is an inner vertex of $t\left(D_{2}\right) \rightarrow_{T} s\left(D_{2}\right)$. If $k>2, D_{3}$ must contain $t\left(D_{2}\right) \rightarrow_{T} s\left(D_{2}\right)$, because $D_{2}$ is of type 1 and a child of $D_{3}$. Therefore, the edge $e$ joining $s\left(C_{i}\right)$ with the parent of $s\left(C_{i}\right)$ in $T$ is contained in $D_{3}$. But since $S_{l}$ is upwards-closed, $e$ is also contained in $S_{l}$, contradicting that $D_{3} \nsubseteq S_{l}$. Thus, $k=2$. If $t\left(D_{2}\right) \rightarrow_{T} s\left(D_{2}\right)$ contains an inner real vertex, $D_{2}$ and $C_{i}$ can be subsequently added as BG-paths with Lemma 12, otherwise 18.1 is satisfied.

Let $C_{i}$ be of type $3 b$. Then all chains in $\left\{D_{2}, \ldots, D_{k}\right\}$ that are of type $2 b$ must be contained in $L_{i}$. Since every caterpillar $L_{j}$ contains the parent of the chain $C_{j}$ and since $S_{l}$ contains no chain in $L_{i}$ due to $\left(R_{1}\right), k>1$ holds and $D_{2}$ is of type $2 b$ with $D_{2} \in L_{i}$. Let $D_{t}$ with $1<t \leq k$ be the minimal chain in $L_{i}$. If $t=k$ and $L_{i}$ is good, all chains in $L_{i}$ can be decomposed to BG-paths according to Lemma 16. If $t=k$ and $L_{i}$ is bad, 18.2 is satisfied. Only the case $k>t$ remains. Then $D_{t+1}$ is of type 1 and, using the same arguments as in the case for type $3 a, s\left(C_{i}\right)$ is an inner vertex of $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$ and $k=t+1$. If $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$ contains an inner real vertex, Lemmas 12 and 16 imply that $D_{k}$ and $L_{i}$ can be iteratively added as set of successive BG-paths, preserving $\left(R_{1}\right)$ and ( $R_{2}$ ). Otherwise, 18.3 is satisfied.

We extend Lemma 18 to non-minimal chains of type 3.
Lemma 19. Let the preconditions of Lemma 18 hold. If $C_{i}$ is not contained in one of the exceptions 18.1-18.3 (as $C_{i}$ ), the chains of type 3 in $H$ that start in $S_{l}$ and their ancestors in $H$ can be successively added as $B G$-paths, preserving ( $R_{1}$ ) and $\left(R_{2}\right)$.

Proof. Using Lemma 18, we add the chains $C_{i}, D_{2}, \ldots, D_{k}$ in $H$ as BG-paths. This partitions $H$ into new segments; let $H^{\prime} \subseteq H \backslash\left\{C_{i}, D_{2}, \ldots, D_{k}\right\}$ be such a


Figure 5: After $C_{i}$ was added, the next minimal chain $C_{j}$ is in no exception.
new segment. If $H^{\prime}$ does not contain chains of type 3 that start in $S_{l}$, the claim follows for such chains in $H^{\prime}$. Otherwise, let $C_{j}$ be the minimal chain of type 3 in $H^{\prime}$ that starts in $S_{l}$ and let $C_{j}>D_{2}^{\prime}>\ldots>D_{k}^{\prime}$ be its ancestors in $H^{\prime}$. We show that $C_{j}$ is not contained in one of the exceptions 18.1-18.3 and can therefore be added as BG-path with Lemma 18, along with its proper ancestors in $H^{\prime}$. First, assume to the contrary that $C_{j}$ is contained in exception 18.1 or 18.3 (see Figure $5(\mathrm{a})$ ). Because $D_{k}^{\prime}$ is a proper descendant of $D_{k}$ and $D_{k}^{\prime}$ is of type 1, $s\left(C_{j}\right) \in V\left(S_{l}\right)$ cannot be an inner vertex of $t\left(D_{k}^{\prime}\right) \rightarrow_{T} s\left(D_{k}^{\prime}\right)$, contradicting the assumption. Now assume to the contrary that $C_{j}$ is contained in exception 18.2 (see Figure $5(\mathrm{~b})$ ). Then $C_{j}$ is of type $3 b$ and part of a bad caterpillar $L_{j}$, whose parent $D$ is not contained in $H^{\prime}$. Because $L_{j}$ contains only chains in $H^{\prime}, D$ must be a descendant of $D_{k}$ and is therefore contained in $H \backslash H^{\prime}$. Since $L_{j}$ is bad, $s\left(C_{j}\right)$ is contained in $S_{l} \cap D$ and it follows with $s\left(C_{j}\right) \neq s(D)$ that $D$ must end in $S_{l}$ at the vertex $s\left(C_{j}\right)$. As $D_{k}$ is the only chain in $H$ that ends in $S_{l}, D=D_{k}$ must hold. But this contradicts $L_{j}$ being bad, as $D$ contains the inner real vertex $s\left(D_{k-1}\right)$. Thus, $C_{j}$ and its ancestors in $H^{\prime}$ can be added, partitioning $H^{\prime}$ into smaller segments. Iterating the same argument for these segments establishes the claim for all chains of type 3 in $H$ that start in $S_{l}$.

Each of the exceptions 18.1-18.3 in Lemma 18 contains a certain path without inner real vertices. We refer to this path in the following way.

Definition 20. Let a chain $C_{i}$ that is of type 1 or $2 a$ and has parent $C_{k}$ be dependent on the path $s\left(C_{i}\right) \rightarrow_{C_{k}} t\left(C_{i}\right)$. Let every chain $C_{i}$ of type $2 b$ or $3 b$ that is contained in a caterpillar $L_{i}$ that parent $C_{k}$ be dependent on the path $s\left(C_{i}\right) \rightarrow_{T} t\left(C_{k}\right) \cup t\left(C_{k}\right) \rightarrow_{C_{k}} s\left(C_{k}\right)$ (chains of type $3 a$ are not dependent on anything).

The following lemma shows that the only chains of type 1 that cannot be added are either backedges or are contained as $D_{k}$ in exceptions 18.1-18.3.

Lemma 21. Let $S_{l}$ be a subdivision constructed under $\left(R_{1}\right)$ and $\left(R_{2}\right)$, let $C_{j}$ be a chain in $S_{l}$ and let $D_{k}$ be a child of $C_{j}$ that is of type 1 and not in $S_{l}$. If $D_{k}$ is not a backedge, there is a chain of type 3 in the segment containing $D_{k}$ that starts in $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right) \subset C_{j}$. If $D_{k}$ is neither a backedge nor contained (as $D_{k}$ ) in the exceptions 18.1 and 18.3, $D_{k}$ can be added as $B G$-path.

Proof. Let $H$ be the segment of $D_{k}$ and assume that $D_{k}$ is not a backedge. We first show that $H$ contains a chain of type 3 that starts in $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$. We can assume that $D_{k}$ is not contained in the exceptions 18.1 and 18.3, as then $H$ would contain such a chain by definition. Let $x$ be the last but one vertex in $D_{k}$. Since $G$ is 3-connected, there is a minimal chain $C_{i}$ entering $T(x)$ such that $s\left(C_{i}\right)$ is an inner vertex of $t\left(D_{k}\right) \rightarrow_{T} s\left(D_{k}\right)$, as otherwise the inner vertices of $D_{k}$ would be separated by $\left\{s\left(D_{k}\right), t\left(D_{k}\right)\right\}$. By definition of the chain decomposition, $C_{i}$ must be of type $3 a$ or $3 b$. Because $D_{k}$ is not contained in exceptions 18.1 and 18.3 and $H$ cannot contain exception 18.2, Lemma 18 can be applied on $C_{i}$, obtaining the last claim.

The next lemma ensures for every subdivision $S_{l}$ the existence of a chain or caterpillar that obtains a BG-path for $S_{l}$, preserving $\left(R_{1}\right)$ and $\left(R_{2}\right)$.

Theorem 22. Let $S_{l}$ be constructed under $\left(R_{1}\right)$ and $\left(R_{2}\right)$ and let $C_{i}$ be a chain such that, for every proper ancestor $C_{j}$ of $C_{i}$, every child of $C_{j}$ and every chain of type 3 starting in $V\left(C_{j}\right)$ is contained in $S_{l}$. Let $X^{\prime}$ be the set of children of $C_{i}$ that are not contained in $S_{l}$ and let $Y^{\prime}$ be the set of chains of type 3 that start in $C_{i}$ and are not contained in $S_{l}$. Then the chains in $X^{\prime} \dot{\cup} Y^{\prime}$ can be successively added as $B G$-paths (possibly being part of caterpillars) such that $\left(R_{1}\right)$ and $\left(R_{2}\right)$ is preserved and every chain in $Y^{\prime}$ is followed by its proper ancestors in its segment.

Proof. By assumption, $C_{i}$ is contained in $S_{l}$. Let $D \nsubseteq S_{l}$ be a child of $C_{i}$. If $C_{i}=C_{0}, D$ must be of type 1. Let $C_{i} \neq C_{0}$. Then $D$ cannot be of type $3 b$, as otherwise it would be contained in $S_{l}$ due to ( $R_{1}$ ) and $C_{i} \subset S_{l}$. It is neither of type $3 a$, since in that case $s(D)$ is contained in a proper ancestor of $C_{i}$, implying $D \subset S_{l}$ by assumption. We conclude that $D$ is of type 1 or 2 and focus on the cases where $D$ can not be added. Let $P$ be the path on which $D$ depends on. If $D$ is of type $1, P$ does not contain an inner real vertex, as otherwise $D$ can be added as BG-path due to Lemma 12. With Lemma 21, $D$ must be either a backedge or be contained as the minimal chain in exception 18.1 or 18.3. If $D$ is of type $2 a, s\left(C_{i}\right)$ is real and neither $t(D)$ nor an inner vertex in $P$ can be real, since otherwise $D$ can be added as BG-path, preserving $\left(R_{1}\right)$ and $\left(R_{2}\right)$. If $D$ is of type $2 b, D$ is the minimal chain of a caterpillar $L_{a}$ with parent $C_{i}$. According to Lemma 16, $L_{a}$ is bad and, thus, corresponds to exception 18.2. The following is a list of the possible cases for which a child $D$ of $C_{i}$ is not added.

1. $D$ is of type 1 without an inner real vertex in $P$ and either a backedge or the minimal chain in exception 18.1 or 18.3
2. $C_{i} \neq C_{0}$ and $D$ is of type $2 a$ without a real vertex in $P \backslash\{s(D)\}$
3. $C_{i} \neq C_{0}$ and $D$ is of type $2 b$ without an inner real vertex in $P$ ( $D$ is the minimal chain in exception 18.2)

We iteratively add all chains $D$ in $X^{\prime} \dot{\cup} Y^{\prime}$ that do not satisfy one of the above three cases 22.1-22.3 for $D \in X^{\prime}$ and whose segments do not contain one of the exceptions 18.1-18.3 for $D \in Y^{\prime}$ (the latter followed by adding the proper ancestors in the segment of $D$ according to Lemma 19). Let $X$ be the set of remaining chains in $X^{\prime}$ and let $Y$ be the set of remaining chains in $Y^{\prime}$. If $X=\emptyset, Y=\emptyset$ holds as well, as otherwise the minimal chain in the segment containing one of the exceptions 18.1-18.3 is a child of $C_{i}$, contradicting $X=\emptyset$. This implies the claim for $X=\emptyset$.

We prove the theorem by showing that $X=\emptyset$ must hold. Assume to the contrary that $X \neq \emptyset$ and let $S_{t}$ be the current subdivision (all segments will be dependent on $S_{t}$ ). Then $C_{i}$ must contain a link $L$ of length at least two, because the dependent path $P$ in each of the cases 22.1-22.3 is in $C_{i}$ and contains a non-real vertex due to the 3 -connectivity and simpleness of $G$. According to Lemma $4, L$ contains an inner vertex $v$ on which a BG-path $B$ starts (not necessarily being a chain and not necessarily preserving $\left(R_{1}\right)$ or $\left.\left(R_{2}\right)\right)$. Let $e$ be the first edge of $B$. Then $e$ is not contained in the segment of any $x \in X$, as otherwise $B$ would not have property 1.2, because $v$ is non-real and all start vertices of the chains in the segment of $x$ that are in $S_{t}$ are contained in $L$. Thus, $C(e)$ cannot be a child of $C_{i}$ and it follows that $s(C(e))=v$. In particular, $C(e)$ is not of type 1 .

The segment of $e$ cannot contain a chain of type 3 that starts in $C_{i}$, as it otherwise contains a chain $x \in X$ of type 1 or $2 b$ due to exceptions 18.1-18.3, contradicting the previous argument. In particular, $C(e)$ is not of type 3 and the only remaining case is that $C(e)$ is of type 2 .

Let $C_{k}$ be the maximal ancestor of $C(e)$ that is not of type 2 . Then $s\left(C_{k}\right)=v$ holds by construction of the chain decomposition and $C_{k}$ must be contained in the segment of $C(e)$ due to $\left(R_{1}\right),\left(R_{2}\right)$ and $v$ being non-real. Since the segment of $e$ cannot contain a chain of type 3 that starts in $C_{i}, C_{k}$ must be of type 1. But then, as $v$ is an inner vertex of $L, C_{k}$ must be a child of $C_{i}$, contradicting that $e$ is not contained in the segment of any $x \in X$. This is a contradiction to the existence of $B$ and it follows that $X=\emptyset$, which implies the claim.

Corollary 23. Let $G$ be a 3 -connected graph with a chain decomposition $C=$ $\left\{C_{0}, \ldots, C_{m-n+1}\right\}$. Then $C_{3}, \ldots, C_{m-n+1}$ can be added to $S_{3}=C_{0} \cup C_{1} \cup C_{2}$ such that every chain satisfies $\left(R_{1}\right)$ and $\left(R_{2}\right)$.

Remark. From Theorem 22, one can possibly derive the characterization of 3connected graphs of Vo in $[7,8]$.

## 4 A Linear-Time Algorithm

With Lemma 8, a chain decomposition $C_{0}, \ldots, C_{m-n+1}$ can be computed by essentially one DFS-traversal in $O(n+m)$. According to Theorem 22, we can find a construction sequence that satisfies $\left(R_{1}\right)$ and $\left(R_{2}\right)$ by the following method: Iteratively for each chain $C_{i}, 3 \leq i \leq m-n$, we add the children of $C_{i}$ and the chains of type 3 that start in $C_{i}$ as BG-paths or, if of type $3 b$, as part of
caterpillars, which can be decomposed into BG-paths. This obtains the desired construction sequence of type 2.(3), which can be transformed easily to the other sequences 2.(2), 2.(4) and 2.(5). However, Theorem 22 does not specify in which order the children of $C_{i}$ and the chains of type 3 that start in $C_{i}$ have to be added. Moreover, this order depends strongly on the input graph. We show how the computation of this order can be integrated in the chain decomposition while preserving the overall running time of $O(n+m)$.

We maintain the following information during the chain decomposition: On each vertex $v \notin\left\{s\left(C_{0}\right), t\left(C_{0}\right)\right\}$, we store a pointer to the unique chain that contains $v$ as an inner vertex. On each chain $C_{j}$, we store pointers to $s\left(C_{j}\right)$ and $t\left(C_{j}\right)$, a list Children ${ }_{12}\left(C_{j}\right)$ of the children of $C_{j}$ that are of type 1 or 2 and the information whether $C_{j}$ was already added. For each chain $C_{i} \neq C_{0}$, we save a list Type $3_{3}\left(C_{i}\right)$ of pointers to the chains of type 3 that start in $C_{i} \backslash\left\{s\left(C_{i}\right), t\left(C_{i}\right)\right\}$. For $C_{0}$, we save the list $\operatorname{Type}_{3}\left(C_{0}\right)$ of pointers to the chains of type 3 that start in $C_{0}$. For each caterpillar $L_{k}$, we maintain the list of chains that are contained in $L_{k}$ and store a pointer to this list on $C_{k}$ and on the minimal chain in $L_{k}$.

Let $C_{i}$ be the current chain, i.e., all chains $C_{j}$ with $j<i$ satisfy the preconditions of Theorem 22. Let $S_{l}$ be the current subdivision. Then Type $3_{3}\left(C_{i}\right)$ contains exactly the chains of type 3 that start in $C_{i}$ and are not contained in $S_{l}$, because the chains $C_{i} \neq C_{0}$ that start with $s\left(C_{i}\right)$ or $t\left(C_{i}\right)$ cannot be contained in $S_{l}$. Let $X$ be the children of $C_{i}$ that are not contained in $S_{l}$. It follows from the first argument in the proof of Theorem 22 that $X \subseteq$ Children $_{12}\left(C_{i}\right)$. In order to ensure $X=$ Children $_{12}\left(C_{i}\right)$, all chains in $\operatorname{Children}_{12}\left(C_{i}\right)$ that are contained in $S_{l}$ are deleted in time $O\left(\left|E\left(C_{i}\right)\right|\right)$.

We partition Type $_{3}\left(C_{i}\right)$ into the classes of chains that are contained in the same segment of $S_{l}$. This is done by storing a pointer for each $C_{j} \in \operatorname{Type}_{3}\left(C_{i}\right)$ to the minimal chain $D_{k}$ of the segment $H$ that contains $C_{j}$. If $D_{k}$ is a backedge, $H$ consists only of one chain $D_{k}=C_{j}$, which can be added immediately due to Lemma 18. For every chain $C_{j} \in \operatorname{Type}_{3}\left(C_{i}\right)$ in a segment $H$ containing at least two chains, we traverse and mark the path $\left(t\left(C_{j}\right) \rightarrow_{T} u\right) \backslash\{u\}$, where $u$ is the first ancestor of $t\left(C_{j}\right)$ in $T$ that is contained in $S_{l}$ or marked. Let $P$ be the first traversed path in $H$. Then the last but one vertex of $P$ must be an inner vertex of the minimal chain $D_{k}$ in $H$ and we use $D_{k}$ as marker for $P$, for all paths that end at $P$ and for all further paths that are contained in $H$. This way $C_{k}$ can be computed by traversing only the chains that we can add by Theorem 22; the total running time amortizes therefore to $O(n+m)$.

Let now $H$ be the segment of a chain $C_{j} \in \operatorname{Type}_{3}\left(C_{i}\right)$. If the minimal chain $D$ in $H$ is not of type 1 or $2 b$, we can immediately add $C_{j}$ along with its ancestors in $H$ due to Lemma 19. Otherwise, $D$ is contained in Children $_{12}\left(C_{i}\right)$ and the previous computation provides the set $H \cap$ Type $_{3}\left(C_{i}\right)$. If $D$ is of type $2 a$ and $t(D)$ is real, we add $D$. Now every remaining chain $D$ in Children $_{12}\left(C_{i}\right)$ is the chain $D$ of one of the cases 22.1-22.3 with the only exception that there may be inner real vertices on its dependent path $P \subseteq C_{i}$. If $P$ contains an inner real vertex, each chain in $H \cap T y p e_{3}\left(C_{i}\right)$ can be added along with its ancestors in $H$. If $P$ does not contain an inner real vertex, no chain in $H$ can be added under $\left(R_{1}\right)$ and $\left(R_{2}\right)$. It is therefore possible to restrict ourselves to find an ordering on the chains Children $_{12}\left(C_{i}\right)$, because the chains in the segments of every such chain can be added immediately afterwards. However, we lack an efficient way of choosing subsequently chains in $\operatorname{Children}_{12}\left(C_{i}\right)$ that contain an inner real vertex in their dependent path, although Theorem 22 guarantees the

(a) The chain $C_{i}$ with the chains in Type $_{3}\left(C_{i}\right) \cup$ Children $_{12}\left(C_{i}\right)$.

(b) The dashed intervals in $I_{0}$ are constructed from the real vertices in $C_{i} \backslash s\left(C_{i}\right)$.

Figure 6: Mapping the dependent paths in the chain $C_{i}$ to intervals. Different shades depict different segments.
existence of such chains. We deal with this problem by the following approach.
Let $H$ be the segment that contains a chain $D \in \operatorname{Children}_{12}\left(C_{i}\right)$. We map the dependent path $P$ of $D$ to a set of integer intervals $I(D)$ on the path $t\left(C_{i}\right) \rightarrow_{C_{i}} s\left(C_{i}\right)$, where vertices depict integers. If $D$ is a backedge, $H=\{D\}$ holds and we set $I(D):=\{[s(P), t(P)]\}$ (see Figure 6). Otherwise, $D$ must be in case 22.1 or 22.3 and $H$ contains a chain of type 3 that starts at an inner vertex of $P$ by Lemma 21. Then the precomputed set $H \cap \operatorname{Type}_{3}\left(C_{i}\right)$ is nonempty and we can efficiently extract the set $\left\{v_{1}, \ldots, v_{k}\right\}$ of vertices in $H \cap P$, $k \geq 3$, ordered by increasing distance to $r \in V(T)$; note that every vertex in $H \cap P$ occurs only once and that $t(D)$ is also contained in $H \cap P$ if $D$ is of type $2 b$. We set $I(D):=\left\{\left[v_{1}, v_{k}\right],\left[v_{1}, v_{2}\right],\left[v_{2}, v_{3}\right], \ldots,\left[v_{k-1}, v_{k}\right]\right\}$. Additionally, we add for the ordered set $\left\{t\left(C_{i}\right)=w_{1}, w_{2}, \ldots, w_{u}=s\left(C_{i}\right)\right\}$ of real vertices in $C_{i}, u \geq 3$, the set of intervals $I_{0}:=\left\{\left[w_{1}, w_{2}\right],\left[w_{2}, w_{3}\right], \ldots,\left[w_{u-2}, w_{u-1}\right]\right\}$. Let $I$ be the set of all generated intervals. By construction, $|I|$ is at most $\left.\frac{3}{2} * \right\rvert\,$ Children $_{12}\left(C_{i}\right)|+|$ Type $_{3}\left(C_{i}\right)|+|\left\{x \mid x\right.$ is inner real node in $\left.C_{i}\right\} \mid$, where the first two terms are linearly dependent on the length of chains that will be added and the last term sums up to at most $n$ in total.

Let two intervals $[a, b]$ and $[c, d]$ overlap if $a<c<b<d$ or $c<a<d<b$. We find the next chain having an inner real vertex on its dependent path by finding a next overlapping interval when starting with an interval in $I_{0}$. We show that these two methods are equivalent. Clearly, an overlap of two intervals, of which exactly one is contained in the preimage of a chain in $S_{l}$, induces an inner real vertex in the other interval. Conversely, let $D_{2}$ be a chain in Children $_{12}\left(C_{i}\right)$ with the dependent path $P_{2}$, let $H$ be the segment containing $D_{2}$ and let $D_{1}$
be the chain that provides the first inner real vertex for $P_{2}$ (real vertices can only be provided by the remaining chains in Children $_{12}\left(C_{i}\right) \cup$ Type $\left._{3}\left(C_{i}\right)\right)$. Then $\left[s\left(P_{1}\right), t\left(P_{1}\right)\right] \in I\left(D_{1}\right)$ overlaps $\left[s\left(P_{2}\right), t\left(P_{2}\right)\right] \in I\left(D_{2}\right)$ or the start vertex $v$ of a chain in $H \cap$ Type $_{3} C_{i}$ is an inner vertex of $\left[s\left(P_{2}\right), t\left(P_{2}\right)\right]$. In the latter case, $v$ is by construction the end point of a chain in $I\left(D_{1}\right)$. It therefore suffices to find an ordering $\pi$ of $I$ that starts with an interval of $I_{0}$ such that for every two subsequent intervals $I_{1}$ and $I_{2}$ in $\pi$ holds

- $I_{1}$ overlaps $I_{2}$,
- $I_{2} \subseteq I_{0}$ or
- $I_{2} \subset I(D)$ for a chain $D$ such that $I(D)$ contains a smaller interval in the ordering.

For the set $I$ of intervals, let the overlap graph $O$ of $I$ be the graph $\left(I, E^{\prime}\right)$ with two intervals being in $E^{\prime}$ if and only if they overlap. If we drop the three latter conditions, there exists an sequential algorithm finding $\pi$ in time $O(|I|)$ by computing the connected components of $O$ [3]. Note that the connected components give us also the information whether there exists $\pi$ in the case of non-3-connected graphs. The Lemmas 4.1 and 4.2 in [3] allow for a simplified variant of this algorithm, which first constructs by two simple sweep lines the forests $F_{1}$ and $F_{2}$ such that the connected components of $F_{1} \cup F_{2}$ (having at most $2|I|$ edges) partitions $I$ into the same vertices as the connected components of $O$. We extend this algorithm to deal with our additional three conditions by merging the intervals in $I_{0}$ and $I(D)$ for each $D \in$ Children $_{12}\left(C_{i}\right)$ to the same connected component; the number of additional edges in $F_{1} \cup F_{2}$ is at most $|I|$ and the asymptotic linear running time is preserved. Using Corollary 23, this gives the following result.

Corollary 24. The construction sequences 2.(2), 2.(3), 2.(4) and 2.(5) of a 3 -connected graph can be computed in time $O(n+m)$.
Remark. The algorithm should easily extend to a certifying 3 -connectivity test. The main argument is that, for a graph $G$ that is not 3 -connected, a subdivision must occur for which no BG-path exists by Theorem 2. But every path that contradicts property 1.2 or 1.3 leads to a cut vertex or separation pair. It would be interesting to show how to get implicitly all the separation pairs (and the unique SPQR-tree), as done in [2].
Remark. Note that $C_{i}$ can already be processed when all chains with a starting vertex in $\operatorname{low}\left(C_{i}\right) \rightarrow_{T} r$ for $r$ being the root of $T$ have been found. Therefore, the chain decomposition does not have to be finished for testing whether a proper ordering on the chains Children $_{12}\left(C_{i}\right) \cup \operatorname{Type}_{3}\left(C_{i}\right)$ exists and these chains can immediately be added in the affirmative case. In that sense, the 3-connectivity is checked locally for each chain $C_{i}$; it might be interesting whether it is faster than the algorithm of Hopcroft and Tarjan [2] in practice.

## 5 Verifying the Construction Sequence

The certificate of the 3 -connectivity of $G$, e.g., the construction sequences 2.2 and 2.3, can be stored in $O(n+m)$ space and easily verified in time $O(n+m)$ as described in [4].

## References

[1] D. W. Barnette and B. Grünbaum. On Steinitz's theorem concerning convex 3polytopes and on some properties of 3 -connected graphs. Many Facets of Graph Theory, Lecture Notes in Mathematics, 110:27-40, 1969.
[2] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. SIAM J. Comput., 2(3):135-158, 1973.
[3] S. Olariu and A. Y. Zomaya. A time- and cost-optimal algorithm for interlocking sets - With applications. IEEE Trans. Parallel Distrib. Syst., 7(10):1009-1025, 1996.
[4] J. M. Schmidt. Construction sequences and certifying 3-connectedness. In 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10), Nancy, France, pages 633-644, 2010.
[5] W. T. Tutte. A theory of 3-connected graphs. Indag. Math., 23:441-455, 1961.
[6] W. T. Tutte. Connectivity in graphs. In Mathematical Expositions, volume 15. University of Toronto Press, 1966.
[7] K.-P. Vo. Finding triconnected components of graphs. Linear and Multilinear Algebra, 13:143-165, 1983.
[8] K.-P. Vo. Segment graphs, depth-first cycle bases, 3-connectivity, and planarity of graphs. Linear and Multilinear Algebra, 13:119-141, 1983.


[^0]:    *This research was supported by the Deutsche Forschungsgemeinschaft within the research training group "Methods for Discrete Structures" (GRK 1408). Email: jens.schmidt@inf.fuberlin.de).

