Recognising $k$-connected hypergraphs in cubic time

Frank Drewes

Abstract


Hypergraph languages generated by hyperedge-replacement grammars of order $k$ are studied. It is shown that for $k$-connected hypergraphs having hyperedges of rank $k$ only the membership problem with respect to such a language is decidable in cubic time. This extends the corresponding result for the graph case recently proved by Vogler. The result is based on a theorem stating that these hypergraphs have unique split trees (so-called collapsed $k$-split trees). This is a generalisation of a 1937 result by MacLane about unique decompositions of cyclically connected graphs.

1. Introduction

Hyperedge replacement is a context-free way of generating (hyper)graphs which has been studied by many authors. In this paper we will be concerned with the problem of recognising hypergraph languages generated by hyperedge-replacement grammars. There are normal-form theorems (cf. [8]) which make it almost trivial to show that hyperedge-replacement languages can be recognised by a nondeterministic Turing machine in polynomial time. However, in contrast to context-free string languages it is rather unlikely that these languages admit a deterministic polynomial-time recognition algorithm, since there exists a very nice proof by Lange and Welzl [12] showing that there are NP-complete hyperedge-replacement languages. For this reason, people try to find special cases for which polynomial-time algorithms can be given.

Recently, a result by MacLane (see [16]) gained new interest as it was reformulated by Vogler [18] in order to show the following.

Correspondence to: F. Drewes, Universität Bremen, Fachbereich 3—Mathematik/Informatik, D-2800 Bremen, Germany. Email: drewes@informatik.uni-bremen.de.

0304-3975/93/$06.00 © 1993—Elsevier Science Publishers B.V. All rights reserved
Every edge-replacement graph language of 2-connected graphs is recognisable in cubic time.

MacLane's theorem states that each 2-connected graph decomposes uniquely into certain components which will be called (0,2)-stars, (1,2)-stars and 3-connected graphs throughout this paper. In order to be able to generalise Vogler's result to hyperedge-replacement languages satisfying related connectedness properties, we extend MacLane's uniqueness theorem to the hypergraph case. It turns out that \( k \)-connected \( k \)-hypergraphs (where a \( k \)-hypergraph is a hypergraph each hyperedge of which is incident with exactly \( k \) distinct vertices) have the desired property: They uniquely decompose into the so-called \((i,k)\)-stars \((0 \leq i \leq k/2)\) and \((k + 1)\)-connected \( k \)-hypergraphs.

Apart from its application in this paper, this result may be interesting in its own respect, because it shows that \( k \)-connected \( k \)-hypergraphs – which may look quite complicated – are in fact built upon a rather simple set of possible components, together with \((k + 1)\)-connected ones.

Using the uniqueness result it is possible to generalise Vogler's algorithm to the case where \( k \)-connected \( k \)-hypergraphs are considered, as we will show. A quite surprising thing is that the polynomial bound on the running time of this algorithm does not even depend on \( k \) (i.e. its degree does not). For every \( k \) we obtain a cubic bound:

Every hyperedge-replacement language of \( k \)-connected \( k \)-hypergraphs is recognisable in cubic time, if generated by a hyperedge-replacement grammar of order \( k \).

**Remark.** In fact, both Vogler's result and the one proved here are formulated a bit more generally. It is not necessary for the whole language to consist of \( k \)-connected \( k \)-hypergraphs. Instead, we have that the subset of all \( k \)-connected \( k \)-hypergraphs of a language generated by a hyperedge-replacement grammar of order \( k \) can be recognised in cubic time. However, this could also be obtained from the above statement using known results about hyperedge-replacement languages. This is because \( k \)-connectedness is certainly a compatible (or monadic second-order definable, or finite) property as investigated, e.g., in [8, 2, 15]. For every such property it is known (see the papers referred to) that the subset of all hypergraphs of a hyperedge-replacement language that satisfy this property is again a hyperedge-replacement language (of the same order).

The paper is organised as follows. In Section 2 the basic notions like hypergraphs, \( k \)-connectedness, splitting and merging are introduced. The last two are direct generalisations of the corresponding notions for graphs (taken from [18]). Further, some basic properties are mentioned. In Section 3 collapsed \( k \)-split decompositions and \( k \)-split trees are investigated and their uniqueness is proved. In Section 4 it is shown how \( k \)-split trees can be computed. The purpose of Section 5 is to define hyperedge

\[\text{Actually, even } O(n^{2.376}), \text{ as the running time is essentially determined by the time which is necessary to recognise context-free string languages (cf. [1]).}\]
recognising k-connected hypergraphs in cubic time

85

replacement by means of merging and to prove a basic lemma, which is then used in Section 6 to show the main result. Finally, in Section 7, we discuss the results and indicate directions for further research.

2. Basic notions

In this section we compile the basic notions concerning hypergraphs and splitting and merging of hypergraphs. These notions extend the ones used by Vogler (see [18]) and will be used later on to define what it means to decompose a hypergraph.

Remark. For the complexity investigations in this paper we will not refer to any particular machine model. However, we assume that pointer structures can be dealt with in an efficient way. So, accessing a node in a (hyper)graph from an incident (hyper)edge is assumed to take constant time. The reader who wants to have a particular model in mind might think of, e.g., random access machines.

General assumption. For the rest of the paper, let $k$ be an arbitrary but fixed natural number.

For a set $A$, $A^*$ denotes the set of all finite lists over elements from $A$, and for every such list $l$ of length $|l|$ the $i$th element of it is referred to as $l_i$. By $[l]$ we denote the set $\{l_1, \ldots, l_{|l|}\}$, and $l-S$ means $l$ without all elements occurring in $S$, for a set $S$. If we have any function $f: A \rightarrow B$, we denote its extensions to set and lists also by $f$, i.e., for $A' \subseteq A$, we define $f(A') = \{f(a) | a \in A'\}$ and for a list $l \in A^*$, $f(l)$ means $f(l_1) \ldots f(l_{|l|})$. A list $l$ with $[l] - S$ and no multiple occurrences of elements is called an ordering of $S$.

For a tree $T$ we let $N_T$ denote the set of its nodes, and, for a node $n \in N_T$, $\text{neigh}_T(n)$ is the set of nodes $n$ is adjacent to in $T$. Two trees $T, T'$ are said to be isomorphic via an isomorphism $f: N_T \rightarrow N_{T'}$ if $f$ is a bijection and $\text{neigh}_{T'}(f(n)) = f(\text{neigh}_T(n))$ for every $n \in N_T$.

A rooted tree $T$ is a tree together with one distinguished node $R_T \in N_T$, called its root. The set of all successors of a node $n \in N_T$ is denoted by $\text{succ}_T(n)$ and for $n \neq R_T$ its predecessor is denoted by $\text{pred}_T(n)$. Two rooted trees $T, T'$ are isomorphic if their underlying trees are isomorphic via an isomorphism preserving the root, i.e., via an isomorphism $f$ with $f(R_T) = R_{T'}$.

Definition 2.1 (Hypergraph). A hypergraph $H$ is a pair $(V_H, E_H)$, where

- $V_H$ is a finite set of nodes (or vertices), and
- $E_H$ is a finite set of hyperedges $e$ each of which is associated with an ordering $\text{sources}(e) \in V_H^*$, called the source list, and a label $\text{lab}(e)$.

The size of $H$, denoted by $|H|$, is defined by $|H| = |V_H| + \sum_{e \in E_H} |\text{sources}(e)|$.

2 We generally assume that $k > 1$, since everything we treat in this paper is trivial for $k \leq 1$. 


Hyperedges may be labelled with a special symbol \(\tau\). In this case we also say that the hyperedge is *unlabelled*.

For a hyperedge \(e\) we define \(\text{num}_e : [\text{sources}(e)] \to \mathbb{N}\) by \(\text{num}_e(\text{sources}(e)_i) = i\). A hyperedge whose source list is of length \(k\) is also called a *\(k\)-hyperedge*. \(H\) is a *\(k\)-hypergraph* if all of its hyperedges are \(k\)-hyperedges. The set of all \(k\)-hypergraphs is denoted by \(\mathcal{H}_k\), and we let \(\mathcal{H}\) denote the set of all hypergraphs.

**Remarks.**

- Note that the source list \(\text{sources}(e)\) of a hyperedge \(e\) and its label \(\text{lab}(e)\) are part of the hyperedge itself, not of the hypergraph. This definition may be a bit unusual, but it turns out to be convenient, because we do not always have to take care of labelling and attachment functions. In particular, if we have two hyperedges \(e \in E_H\) and \(e' \in E_{H'}\) of two hypergraphs \(H\) and \(H'\), \(e = e'\) does automatically mean \(\text{sources}(e) = \text{sources}(e')\) and \(\text{lab}(e) = \text{lab}(e')\).
- The reader should be conscious of the fact that the sources of a hyperedge are pairwise distinct, by definition. This is important because, otherwise, the restriction to \(k\)-hypergraphs would be useless. Also, \(\text{num}_e\) would be ambiguous otherwise.
- A hyperedge \(e\) and a node \(v\) appearing in its source list are said to be *incident* with each other, as usual, and \(e\) is said to *connect* each two distinct ones of its sources with each other. The sources of a hyperedge are said to be adjacent.

Two hyperedges incident with the same set of vertices are called *parallel*.

**Example 2.2 (Hypergraph).** If we want to visualise hypergraphs, we draw nodes as filled circles and hyperedges as squares with lines pointing to their sources. If necessary, labels will appear inside the squares representing hyperedges. We usually do not indicate the order on the sources of a hyperedge. (However, in case this order is important, it will be indicated by numbers on the lines pointing to them.) A 2-hyperedge \(e\) may also be drawn as an ordinary edge, i.e., an arrow pointing from \(\text{sources}(e)_1\) to \(\text{sources}(e)_2\). As an example, a hypergraph with four nodes and three (hyper)edges, one of which is labelled \(A\), the other two \(B\), is shown in Fig. 1.

Isomorphisms and weak isomorphisms between hypergraphs are defined next.

**Definition 2.3 (Isomorphic hypergraphs).** Let \(H\) and \(H'\) be hypergraphs.
Recognising $k$-connected hypergraphs in cubic time

Two functions $b_V : V_H \to V_H$ and $b_E : E_H \to E_H$ are weakly consistent if for all $e \in E_H$ it holds that $b_V(\text{sources}(e)) = \text{sources}(b_E(e))$. If $b_V(\text{sources}(e)) = \text{sources}(b_E(e))$ then $b_V$ and $b_E$ are consistent.

$H$ and $H'$ are weakly isomorphic, denoted by $H \equiv_w H'$, if a weak isomorphism $b$ between $H$ and $H'$ exists, which is a pair of bijections $b_V : V_H \to V_{H'}$, $b_E : E_H \to E_{H'}$ that is weakly consistent and preserves hyperedge labels. If there is even a consistent pair of such bijections (called an isomorphism), then $H$ and $H'$ are isomorphic.

For $V \subseteq V_H$ and $e \in E_H$ we define $e - V$ to be $e$ without its sources from $V$, i.e., $e - V = e'$, with $\text{sources}(e') = \text{sources}(e) - V$ and $\text{lab}(e) = \text{lab}(e')$. This has a natural extension to hypergraphs; so, let

$$H - V = (V_H - V, \{e - V | e \in E_H\}).$$

(It is important to see that a hyperedge $e$ incident with a node from $V$ is not totally deleted; vertices not belonging to $V$ can still be connected by $e$.)

A path between two nodes $v_0, v_1 \in V_H$—also called a $v_0v_1$-path—is an alternating sequence $w_0e_1w_1e_2 \ldots w_ne_n$ of nodes $w_0, \ldots, w_n \in V_H$ and hyperedges $e_1, \ldots, e_n \in E_H$, such that $w_0 = v_0$, $w_n = v_1$, and $e_i$ connects $w_{i-1}$ with $w_i$ for $i = 1, \ldots, n$. If there are two vertices, $v_0, v_1 \in V_H$, such that there is no path between them, we say that $H$ is disconnected. If $H - V$ is disconnected for some $V \subseteq V_H$, i.e., if there are two nodes $v_0, v_1 \in V_H - V$ with no $v_0v_1$-path in $H - V$, then $V$ is said to disconnect $H$, and to separate $v_0$ from $v_1$.

Note that in a disconnected hypergraph $H$, for every node $v_0$ there is at least one node $v_1$ separated from it. Hence, if $V$ separates $v_0$ from $v_1$ for every node $v_0 \in V_H - V$, there is a node $v'_1 \in V_H$ separated from $v'_0$ by $V$.

**Definition 2.4 (k-connectedness).** A hypergraph $H$ is said to be $k$-connected if $|V_H| \geq k$ and no $V \subseteq V_H$ with $|V| < k$ disconnects $H$.

**Remark.** Observe that every hypergraph all of whose nodes are adjacent is $|V_H|$-connected. In particular, if $H$ is a $k$-hypergraph and $|V_H| = k$ then it is $k$-connected if and only if $E_H \neq \emptyset$.

The following lemma will turn out to be useful from time to time.

**Lemma 2.5.** Let $H \in \mathcal{H}_k$ be $k$-connected. If $|V_H| > k$ then every node $v \in V_H$ is incident with at least two nonparallel hyperedges.

**Proof.** This follows directly from the more general observation that every vertex of a $k$-connected hypergraph $H$ with $|V_H| > k$ must be adjacent to at least $k$ nodes. This is because we can separate any node from all others by deleting the set of nodes it is adjacent to. $\square$
The interior of a path \( p = w_0 e_0 \ldots e_n w_{n+1} \) is defined by \( \text{int}(p) = \{ w_1, \ldots, w_n \} \). We call two paths \( p \) and \( p' \) openly disjoint if \( \text{int}(p) \cap \text{int}(p') = \emptyset \). Concerning \( k \)-connectedness of hypergraphs with more than \( k \) nodes we have the following characterisation similar to the graph case.

**Lemma 2.6** (Characterisation of \( k \)-connectedness). Let \( H \) be a hypergraph and let \( |V_H| > k \).

\( H \) is \( k \)-connected if and only if there are at least \( k \) pairwise openly disjoint \( vv' \)-paths in \( H \) between each two nodes \( v, v' \in V_H \), \( v \neq v' \).

**Proof.** Due to an old result by Whitney [19], a graph is \( k \)-connected if and only if there are at least \( k \) pairwise openly disjoint paths between each two distinct ones of its nodes. We want to use this result. Define an unlabelled graph \( \hat{H} \) by \( V_{\hat{H}} = V_H \) and \( E_{\hat{H}} = \bigcup_{e \in E_H} \{ e_{i,j} \mid 1 \leq i < j \leq |\text{sources}(e)| \} \), where \( \text{sources}(e_{i,j}) = \text{sources}(e) \) for each \( e \in E_H \), \( 1 \leq i < j \leq |\text{sources}(e)| \). This means that we replace each hyperedge by a complete graph on its sources. Every path \( p = w_0 e_1 \ldots e_n w_{n+1} \) in \( H \) gives rise to a path \( \hat{p} = w_0 e_{1,1} \ldots e_{n,n} \), where \( e_{l,l} = e_{l-1,l} = \text{sources}(e) \) for \( l = 1, \ldots, n \). Obviously, this defines a bijection between paths in \( H \) and paths in \( \hat{H} \). Since both paths go through the same nodes, it is also clear that
- two paths \( p_1 \) and \( p_2 \) are openly disjoint in \( H \) if and only if \( \hat{p}_1 \) and \( \hat{p}_2 \) are openly disjoint in \( \hat{H} \), and
- any set \( V \subseteq V_{\hat{H}} \) separates two nodes \( v \) and \( v' \) in \( H \) if and only if it does so in \( \hat{H} \).

Thus, the assertion holds for \( H \) if and only if it holds for \( \hat{H} \), which is true by the mentioned theorem by Whitney. \( \square \)

The two notions of *merging* and *splitting* are crucial for our investigations. These are straightforward extensions of the notions Vogler uses for the graph case (cf. [18]).

**Definition 2.7** (Merging and splitting). Let \( H_1, H_2 \in \mathcal{H} \) such that \( E_{H_1} \cap E_{H_2} = \{ e \} \) and \( V_{H_1} \cap V_{H_2} = [\text{sources}(e)] \).

1. **Merging** \( H_1 \) and \( H_2 \) (along \( e \)) yields the hypergraph
   \[
   H_1 \langle H_2 \rangle = (V_{H_1} \cup V_{H_2}, (E_{H_1} \cup E_{H_2}) \setminus \{ e \}).
   \]

2. A hypergraph \( H \) \( k \)-splits into \( H_1 \) and \( H_2 \) (with new hyperedge \( e \)) if
   - \( H = H_1 \langle H_2 \rangle \),
   - \( |\text{sources}(e)| = k \) and
   - \( |H_1|, |H_2| < |H| \).

   The splitting is called *general* if \( e \) is unlabelled; otherwise, it is said to be a *particular* one.

**Remarks.**
- The set \( V_{H_1} \cap V_{H_2} = [\text{sources}(e)] \) is called a \((k-)\)splitset for \( H \), and we say that \( H_1 \) is the result of *splitting off* \( H_2 \), denoted \( H!H_2 \). (Note that this is symmetric, i.e., \( H_2 = H!H_1 \), too.)
Recognising k-connected hypergraphs in cubic time

The set $inner_H(H_1)$ of inner vertices of $H_1$ with respect to $H$ is defined by

$$inner_H(H_1) = V_{H_1} - \lfloor \text{sources}(e) \rfloor.$$  

We denote $inner_H(H_1)$ also by $inner(H_1)$ if $H$ is understood from the context.

- The set of all hypergraphs $H$ k-splits into is denoted by $split_k(H)$, i.e.,

$$split_k(H) = \{H_1 \mid H k\text{-splits into } H_1 \text{ and } H_2 \text{ for some } H_2\}.$$  

- Note that we require $|H_1|, |H_2| < |H|$ in the definition of k-splitting. This constraint prevents trivial splittings. In particular, it implies that every repeated splitting process must eventually terminate.

- If we write $H_1 \langle H_2 \rangle$ or $H_1!H_2$, we always implicitly assume that the relevant conditions are satisfied.

Example 2.8. If we merge the two hypergraphs on the left-hand side of Fig. 2 along $e$, we get the one on the right-hand side. Conversely, $H_1 \langle H_2 \rangle$ can be split into $H_1$ and $H_2$ (provided that $H_1 \langle H_2 \rangle$ is larger than each of $H_1$ and $H_2$).

There is a close relationship between k-connectedness on the one hand and k-splitting (k-splitsets) on the other. The following two lemmas are concerned with this correspondence. The first one is of a more general nature, whereas the second one deals with the particular case of k-connected k-hypergraphs.

Lemma 2.9 (Splitting k'-connected hypergraphs). Let $H \in \mathcal{H}$ with $|V_H| \geq k'$.

$H$ is k'-connected if and only if for every $k'' < k'$ such that $H k''$-splits into hypergraphs $H_1$ and $H_2$ we have $|V_{H_1}| = k''$ or $|V_{H_2}| = k''$.

Remark. The lemma states more or less that k'-connectedness is equivalent to the nonexistence of k''-splitsets for all $k'' < k'$. The only possible exception is that there may be a set $V \subseteq V_H$ containing just $k''$ nodes and hyperedges $e_1, \ldots, e_n$ having their

![Fig. 2. Merging $H_1$ and $H_2$ along $e$.](image-url)
sources in \( V \) only. Provided that the resulting hypergraphs are large enough, we are then able to split off \((V, \{e, e_1, \ldots, e_n\})\), where \( e \) is a new hyperedge with \([\text{sources}(e)] = V\).

Observe that this means that a \( k \)-hypergraph (with \( V_H \geq k \)) is \( k \)-connected if and only if it has no splitset of size smaller than \( k \). This is because we cannot split off a hypergraph with fewer than \( k \) nodes since this hypergraph could not contain any hyperedge of the original hypergraph. Such a split would violate the size requirement: the remaining part contained all the nodes and hyperedges and, hence, would be at least as large as \( H \).

**Proof.** Let \( H_1 = (V_1, E_1), H_2 = (V_2, E_2) \) merge along \( e \) with \( |\text{sources}(e)| = k'' \) and let \( H = H_1 \langle H_2 \rangle \) be \( k' \)-connected. By definition of merging, \( H_1 \) and \( H_2 \) intersect in \( V = [\text{sources}(e)] \) only, i.e., every \( v_1 v_2 \)-path with \( v_1 \in V_1 \) and \( v_2 \in V_2 \) must contain a node of \( V \). Hence, \( V \) separates the nodes in \( V_1 - V \) from those in \( V_2 - V \), and, if both \( |V_1| > k'' \) and \( |V_2| > k'' \), these sets are nonempty. So, \( H \) is not \( k' \)-connected.

For the other direction, let \( H \) not be \( k' \)-connected. Since \( V_H \geq k' \), there is a minimum number \( k'' < k' \), for which there is a set \( V \subseteq V_H \) disconnecting \( H \). We show that there are \( H_1, H_2 \in \not\emptyset \) with \( |V_{H_1}|, |V_{H_2}| > k'' \) such that \( H \) \( k'' \)-splits into \( H_1 \) and \( H_2 \). Let \( v \in V_H - V \) and let \( H'_1 = (V'_1, E'_1) \) be the hypergraph containing exactly the nodes reachable from \( v \) on a path in \( H \) not using any node of \( V \) (except, perhaps, as an endpoint), i.e.,

\[
V'_1 = \{ v \in V_H \mid \exists v' \cdot \text{path } p: \text{int}(p) \cap V = \emptyset \}
\]

and

\[
E'_1 = \{ e \in E_H \mid \text{sources}(e) \cap (V'_1 - V) \neq \emptyset \}.
\]

The definition of \( H'_1 \) is sound because all sources of hyperedges in \( E'_1 \) are in \( V'_1 \). Now, let \( H'_2 \) be all the rest of \( H \), i.e., \( H'_2 = (V'_2, E'_2) = ((V_H - V'_1) \cup V, E_H - E'_1) \). For \( i \in \{1, 2\} \), we have that \( V'_i - V \neq \emptyset \) (since \( V \) disconnects \( H \) and \( v \in V_1 - V \)). Furthermore, \( V \subseteq V'_i \) because \( V \) is minimal and \( V'_i \cap V \) disconnects \( H \), by definition of \( V'_i \).

We set \( H_1 = (V_1, E_1) = (V'_1, E'_1 \cup \{\text{new}\}) \) and \( H_2 = (V_2, E_2) = (V'_2, E'_2 \cup \{\text{new}\}) \), where \( \text{new} \) is a new hyperedge with \( |\text{sources}(\text{new})| = V \). By definition of merging, now

\[
H_1 \langle H_2 \rangle = (V_1 \cup V_2, E_1 \cup E_2 - \{\text{new}\}) = H.
\]

So, it remains to show that \(|H_1|, |H_2| < |H|\).

Let, for \( i \in \{1, 2\} \), \( \#E_i = \sum e \in E_i |\text{sources}(e)| \). Due to the minimality of \( V \), we have that, for every \( v_1 \in V'_1 - V \) and \( v_2 \in V \), there is a \( v_1 v_2 \)-path in \( H_i \). Thus, every vertex of \( H_i \) is incident with at least one hyperedge and, hence, \( \sum e \in E_i |\text{sources}(e)| > k'' \), implying that \( \#E_i > 2k'' \) since \( E_i = E_i \cup \{\text{new}\} \). So, for \( \{i, j\} = \{1, 2\} \) we get

\[
|H| = |V_i| + |V_j| - k'' + \#E_i + \#E_j - 2k'' \quad \text{(for } |V_i \cap V_j| = |V| = k'' \text{ and new} \not\in E_H \)
\]

\[
\geq |V_i| + 1 + \#E_i + 1 \quad \text{(see above)}
\]

\[
> |H_i|,
\]

which is what we had to show. \( \square \)
Lemma 2.10 (Splitting \( k \)-connected \( k \)-hypergraphs). Let \( H = H_1 \langle H_2 \rangle \) be a \( k \)-hypergraph.

1. If \( H \) \( k \)-splits into \( H_1 \) and \( H_2 \), then \( H \) is \( k \)-connected if and only if both \( H_1 \) and \( H_2 \) are \( k \)-connected.

2. If \( H \) is \( k \)-connected then \( |H_1|, |H_2| < |H| \) (i.e., \( H \) \( k \)-splits into \( H_1 \) and \( H_2 \)) if and only if \( |E_{H_1}|, |E_{H_2}| \geq 3 \).

3. \( H \) is \( k \)-connected and has no \( k \)-splitset (i.e., \( H \) is minimum with respect to \( k \)-splitting), but \( |E_{H}| \geq 3 \), if and only if either
   
   (i) \( |E_{H}| = 3 \) and \( |V_{H}|-k+i \), where \( 0 \leq i \leq k/2 \) and all nodes of \( H \) are adjacent, or
   
   (ii) \( |E_{H}| \geq 4 \) and \( H \) is \((k+1)\)-connected, with no parallel hyperedges.

**Proof.** (1) Let \( E_{H_1} \cap E_{H_2} = \{ \text{new} \} \). We first show that both \( H_1 \) and \( H_2 \) are \( k \)-connected if \( H \) is. For this consider, without loss of generality, \( H_1 \) with \( v, v' \in V_{H_1} \). We can modify each \( vv' \)-path in \( H \) to become a \( vv' \)-path in \( H_1 \) by simply replacing each maximum \( ww' \)-subpath going entirely through \( H_2 \) \((w, w' \in [\text{sources(new)}])\), then) by \( w \text{ new } w' \). Therefore, there is a \( vv' \)-path in \( H_1 - S \) for \( S \subseteq V_{H_1} \), if there is one in \( H - S \), implying that \( H_1 \) is \( k \)-connected, since \( H \) is.

For the other direction consider some \( S \subseteq V_{H}, \) with \( |S| < k \), \( S_1 = V_{H_1} \cap S \) and \( I = [\text{sources(new)}] \). We have to show that \( S \) does not disconnect \( H \). Let \( H' = (V_{H_1}, E_{H_1} - \{ \text{new} \}) \) be the part of \( H \) belonging to \( H_1 \). For every node \( v \in V_{H_1} - S_1 \), let

\[
C_1(v) = \{ v' \in I - S_1 | \exists vv' \text {-path in } H_1 - S_1 \}
\]

be the set of nodes from \( I - S_1 \) \( v \) is connected with in \( H_1' - S_1 \). We show the following:

For all \( v \in V_{H_1} - S_1 \) : \( |C_1(v)| \geq k - |S_1| \). \( \quad (1) \)

By symmetry we then get a similar statement for \( H_2 \), and are thus done: for every two nodes \( v_1 \in V_{H_1}, v_2 \in V_{H_2} \) it follows that \( |C_1(v_1)| \geq k - |S_1|, |C_2(v_2)| \geq k - |S_2| \) (with the obvious definitions for \( S_2 \) and \( C_2 \)). Therefore,

\[
|C_1(v_1) + C_2(v_2)| \geq 2k - |S| - |S \cap I|
\]

\[
> k - |S \cap I| \quad \text{(since } |S| < k) \]

\[
= |I - S| \quad \text{(since } |I| = k),
\]

implying that \( C_1(v_1) \cap C_2(v_2) \neq \emptyset \), which in turn means that there is at least one \( v_1v_2 \)-path in \( H - S \).

To prove (1) consider the case \( v \notin I \) first. The assertion is certainly true if \( C_1(v) = I - S_1 \). If this is not the case, let \( v' \in I - S_1 \) be a node which is separated from \( v \) by \( S_1 \). By Lemma 2.6, there are \( k \) openly disjoint paths between \( v \) and \( v' \) in \( H_1 \). Therefore, there are at least \( k - |S_1| \) in \( H_1 - S_1 \). Since none of these paths exists in \( H_1' - S_1 \), they must all use \( \text{new} \), hence a node of \( I - S_1 \). Since the paths are openly disjoint we are done.
In case \( v \in I \), we have some \( e \neq \text{new} \) being incident with \( v \) since each node of \( H_1 \) must be incident with at least two hyperedges. (For \( |V_{H_1}| > k \) this is due to Lemma 2.5. If \( |V_{H_1}| = k \), it is also true because of the size requirement.) If there is some \( v' \in [\text{sources}(e)] - (S_1 \cup I) \) we get \( |C_1(v)| \geq |C_1(v')| \geq k - |S_1| \) by the above. Otherwise, \([\text{sources}(e)] - S_1 \subseteq I\) and, hence, \(|C_1(v)| \geq |\text{sources}(e)| - |S_1| \geq k - |S_1|\).

2. Let \([i,j] = \{1,2\}\). Using the relevant definitions it is easily verified that \(|H| - |H_j| = |V_{H_j}| + k|E_{H_j}| - 3k| \); so, we have \(0 < |V_{H_j}| + k|E_{H_j}| - 3k| \), by the size requirement. Now \(|V_{H_j}| = k| \) does immediately imply \(|E_{H_j}| > 2 \). Otherwise, by the first part of the lemma, \( H_1 \) is \( k \)-connected. Thus, we get that every node of \( H_1 \) is incident with at least two hyperedges, i.e., \(2|V_{H_1}| \leq k|E_{H_1}| \) and, hence, \(0 < 3/2k|E_{H_1}| - 3k < 2|E_{H_1}|\).

3. Let \( H \) be \( k \)-connected with \(|E_H| > 3\) and let there be no \( k \)-splitset for \( H \). If there are two parallel hyperedges \( e_1, e_2 \in E_H \), we define \( H_1 = (V_H, E_H - \{e_1, e_2\} \cup \{\text{new}\}) \) and \( H_2 = ([\text{sources}(e_1)], \{e_1, e_2, \text{new}\}) \), where \( \text{new} \) is new with \([\text{sources}(\text{new})] = [\text{sources}(e_1)] \). Then \( H = H_1 \cup H_2 \), so \(|E_{H_1}| < 3| \), by the second part of the lemma (since \( H \) has no splitset) and, hence, \(|E_{H_1}| = 3\). This implies that \(|V_{H_1}| = k| \) because every node is incident with at least 2 hyperedges; so, \( H \) is of the first type.

If there are no parallel hyperedges in \( H \) then \(|V_{H}| > k| \), i.e., \( H \) is \((k + 1)\)-connected, by Lemma 2.9. Hence, either \( H \) is of the second type or \(|E_{H}| = 3| \). In the latter case, because every node is incident with at least two hyperedges, we have \(|V_{H}| \leq 3/2k| \), i.e., \(|V_{H}| = k + i \) for some \( i \) with \(0 < i < k/2 \) and all nodes are necessarily adjacent.

For the other direction, if \( H \) is of the first type, there can be no \( k \)-splitset because, by the second part of the lemma, this requires \(|E_{H}| \geq 4\). Also, \( H \) is \(|V_{H}|\)-connected because all nodes are adjacent. If \( H \) is of the second type, we have \(|V_{H}| > k| \). Hence, Lemma 2.9 applies implying that \( H \) has not \( k \)-splitset. \( \square \)

The first part of the above lemma comprises the reason why \( k \)-splitting is convenient for dealing with \( k \)-connected \( k \)-hypergraphs. The new hyperedge splitting causes the components of a \( k \)-connected \( k \)-hypergraph to be \( k \)-connected again. In some sense the new hyperedge substitutes for the part split off, so that one can think of it as a placeholder for a \( k \)-connected \( k \)-hypergraph. Unfortunately, the "if" direction is not true any more when we consider graphs instead of \( k \) hypergraphs.

The two special types of \( k \)-connected \( k \)-hypergraphs defined below play an important role in the following.

**Definition 2.11 ((i, k)-triples and k-boxes).** Let \( H \in H_k \) be \( k \)-connected and let \( i \), \( 0 \leq i \leq k/2 \), be a natural number.

\( H \) is an \((i, k)\)-triple if \(|V_H| = k + i \) and \(|E_H| = 3 \) and a \( k \)-box if it is \((k + 1)\)-connected and has more than three hyperedges but no parallel ones.\(^3\) The set of all \( k \)-boxes is denoted by \( \text{box}_k \).

\(^3\) As we will see later on, \( k \)-boxes split off a \( k \)-connected \( k \)-hypergraph cannot overlap with other components; hence, the name \( k \)-boxes.
According to Definition 2.11, \((i, k)\)-triples are \(k\)-hypergraphs of the kind described in Lemma 2.10(3i) and \(k\)-boxes are those of Lemma 2.10(3ii). Hence, we always have \(0 \leq i \leq k/2\), and every \(k\)-connected \(k\)-hypergraph \(H\) with \(E_H \geq 3\), which is minimum with respect to \(k\)-splitting, is either an \((i, k)\)-triple for some \(i, 0 \leq i \leq k/2\), or a \(k\)-box. We have the following lemma.

**Lemma 2.12** (Weak isomorphism of \((i, k)\)-triples). Let \(0 \leq i \leq k/2\) and let \(H, H'\) be unlabelled \((i, k)\)-triples. Then \(H \equiv_w H'\).

**Proof.** Let \(E_H = \{e_1, e_2, e_3\}\), let \(V = \bigcap_{i=1, \ldots, 3} \text{sources}(e_i)\) and let, for \(1 \leq i < j \leq 3\), \(V_{\{i,j\}} = [\text{sources}(e_i)] \cap [\text{sources}(e_j)] - V\). By definition, these four node sets are mutually disjoint and \(V_H = V \cup V_{\{1,2\}} \cup V_{\{1,3\}} \cup V_{\{2,3\}}\) since every node is incident with at least two hyperedges. For the same reason we also have \([\text{sources}(e_i)] \cup [\text{sources}(e_j)] = V_H\); thus, we get \(|V_{\{i,j\}}| = 1\) for \(1 \leq i < j \leq 3\).

The same applies to \(H'\); so, with \(E_H = \{e'_1, e'_2, e'_3\}\), we can define \(V', V'_{\{1,2\}}, V'_{\{1,3\}}\) and \(V'_{\{2,3\}}\) similarly. By the above, there is a bijection \(b_V : V_H \rightarrow V_{H'}\) satisfying \(b_V(V) = V'\) and \(b_V(V_{\{i,j\}}) = V'_{\{i,j\}}\) for \(1 \leq i < j \leq 3\). Defining \(b_E : E_H \rightarrow E_{H'}\) by \(b_E(e_i) = e'_i\) for \(i = 1, \ldots, 3\), we get that \((b_V, b_E)\) is a weak isomorphism since \(b_V([\text{sources}(e_i)]) = b_V(V \cup V_{\{i,j\}} \cup V_{\{i,j'\}}) = V' \cup V'_{\{i,j\}} \cup V'_{\{i,j'\}} = [\text{sources}(e'_i)] = [\text{sources}(b_E(e_i))],\) where \(j\) and \(j'\) are such that \(\{i, j, j'\} = \{1, 2, 3\}\). \(\square\)

Observe that in the proof above it is irrelevant which one of the hyperedges of \(H\) is mapped to which one of \(H'\) by the bijection \(b_E\). Every such bijection defines a set of weak isomorphisms since it determines \(b_V\) as above. In particular, every bijection from \(E_H\) onto itself can be extended to a (weak) automorphism on \(H\). Intuitively, this means that \((i, k)\)-triples are symmetric with respect to their hyperedges.

As an example, for \(k = 5\) we get the unlabelled \((i, 5)\)-triples shown in Fig. 3 (up to weak isomorphism). As one can see here, \((i, k)\)-triples can be drawn like stars with three peaks. A peak consists of those nodes incident with the same two hyperedges (i.e., the peaks are the sets \(V_{\{i,j\}}\) in the above proof) and the centre of the star is given by the set of nodes each of which is incident with all three hyperedges. So, the centre consists of \(k - 2i\) nodes and the peaks are of size \(i\) each. A straightforward generalisation of this are hypergraphs having an arbitrary number of peaks, each of size \(i\), and a centre of size \(k - 2i\) all hyperedges are incident with. These are the so-called \((i, k)\)-stars defined in the following.

**Fig. 3.** All unlabelled \((i, 5)\)-triples up to weak isomorphism.
**Definition 2.13 ((i, k)-star).** Let \( H \in \mathcal{W}_k \) and let \( i \) be a natural number, \( 0 \leq i \leq k/2 \).

\( H \) is an \((i, k)\)-star if there are mutually disjoint sets \( \text{centre}, \text{peak}_0, \ldots, \text{peak}_{n-1} \) and there are hyperedges \( e_0, \ldots, e_{n-1} \) for some \( n > 2 \), such that

1. \( |\text{centre}| = k - 2i \) and \( |\text{peak}_0| = \cdots = |\text{peak}_{n-1}| = i \),
2. \( V_H = \text{centre} \cup \bigcup_{j=0}^{n-1} \text{peak}_j \) and \( E_H = \{ e_1, \ldots, e_{n-1} \} \), and
3. \( \{ \text{sources}(e_j) \} = \text{peak}_j \cup \text{peak}_{j+1 \text{(mod } n)} \cup \text{centre} \) for \( 0 \leq j < n \).

We define, for \( j = 0, \ldots, n-1 \),

\[
\text{peaks}_H(e_j) = \{ \text{peak}_j, \text{peak}_{j+1 \text{(mod } n)} \} \quad \text{and} \quad \text{peaks}_H = \{ \text{peak}_0, \ldots, \text{peak}_{n-1} \}.
\]

Two peaks \( \text{peak}_j, \text{peak}_{j'} \in \text{peaks}_H \) are said to be **adjacent** if there is a hyperedge \( e \in E_H \) such that \( \text{peaks}_H(e) = \{ \text{peak}_j, \text{peak}_{j'} \} \), i.e., if \( j = j' + 1 \text{(mod } n) \) or \( j' = j + 1 \text{(mod } n) \).

The set of all \((i, k)\)-stars is denoted by \( \text{star}_i^k \).

**Remark.** Note that \( \text{peaks}_H(e_j) \) as well as \( \text{peaks}_H \) are sets of sets of nodes.

By definition, for fixed \( i \) and \( k \), fixed-size \((i, k)\)-stars are always weakly isomorphic (as long as the labelling coincides). The peaks consist of \( i \) nodes each and the centre is of size \( k - 2i \). Each hyperedge connects the nodes of two adjacent peaks and those of the centre with each other.

The set of all \((i, k)\)-stars can also be obtained by repeatedly merging \((i, k)\)-triples, as expressed by the following characterisation.

**Lemma 2.14 (Constructing \((i, k)\)-stars by means of merging).** Let \( i \) be a natural number, \( 0 \leq i \leq k/2 \).

The set \( \text{star}_i^k \) is the set \( S \) inductively defined as follows:

(i) If \( H \) is an \((i, k)\)-triple then \( H \in S \).

(ii) Let \( H_1, H_2 \in S \) merge along \( e \). Then \( H = H_1 \cup H_2 \in S \), provided that

\[
\text{peaks}_{H_1}(e) = \text{peaks}_{H_2}(e) \quad \text{(this will be called the peak condition in the sequel)}.
\]

In the latter case we have that \( \text{centre}_H = \text{centre}_{H_1} = \text{centre}_{H_2} \) and for all \( e' \in E_H \)

\[
\text{peaks}_H(e') = \begin{cases} 
\text{peaks}_{H_1}(e') & \text{if } e' \in E_{H_1}, \\
\text{peaks}_{H_2}(e') & \text{if } e' \in E_{H_2}.
\end{cases}
\]

**Remarks.**

- The peak condition ensures that the division of \( e \) into two peaks is the same in both hypergraphs, so that merging these hypergraphs along \( e \) does not destroy the regular structure. Observe that this condition is not automatically satisfied, because how the sources of a hyperedge divide into peaks does not only depend on the hyperedge itself but also on its two “neighbours” in the \((i, k)\)-star.

- If we look at the case \( k = 2 \), our \((0, 2)\)-triples compare to Vogler's **triple-bonds**, and the \((1, 2)\)-triples are his **triangles** (see [18]). Since the peak condition is automatically satisfied in the graph case (i.e., for \( k = 2 \)), \((0, 2)\)-stars and \((1, 2)\)-stars are just **bonds** (see [18]) and cycles, respectively.
**Proof.** By a straightforward induction using the definition of merging and the peak condition. □

Using the first part of Lemma 2.10, Lemma 2.14 does, in particular, imply that all \((i,k)\)-stars are \(k\)-connected.

A slightly more general formulation of the peak condition can be given which is isomorphism-independent and, hence, more convenient to use with hyperedge replacement. For \(H, H' \in \text{star}_k^i\) we say that \(e \in E_H\) and \(e' \in E_{H'}\) satisfy the peak condition if \(\text{num}_e(\text{peaks}_H(e)) = \text{num}_e(\text{peaks}_{H'}(e'))\). This means, rather than comparing the peaks directly – the result of which depends on the identity of nodes – we do now compare how the sources of the involved hyperedges divide between the peaks. It should be clear that this new formulation is isomorphism-independent. However, it is *not* independent of weak isomorphism. (Observe that, in the situation above, both formulations mean the same because there we had \(e = e'\), and \(\text{num}_e\) is an injection.)

**Example 2.15 (Peak condition).** As an example, consider the situation depicted in Fig. 4. The hyperedges \(e\) and \(e'\) satisfy the peak condition, but \(e\) and \(e''\) do not. If \(e' = e\), we can merge both hypergraphs and get the \((2,4)\) star shown in Fig. 5. On the other hand, if \(e'' = e\), merging \(H\) and \(H'\) yields the result in Fig. 6, which is no \((2,4)\)-star.

A nice property of \((i,k)\)-stars is that \(k\)-splitting some \(H \in \text{star}_k^i\) yields two \((i,k)\)-stars back again, which in addition satisfy the peak condition. So, every \(k\)-splitting of an
(i, k)-star reverses one step of the process which can be used to construct (i, k)-stars, as in Lemma 2.14. This is made precise in the following lemma.

**Lemma 2.16** (Closedness of $\text{star}_k^i$ under $k$-splitting). For every natural number $i$, $0 \leq i \leq k/2$, the set $\text{star}_k^i$ is closed under $k$-splitting, i.e., if $H \in \text{star}_k^i$ and $H_1 = H \cap H_2$ by a splitset of size $k$, then $H_1, H_2 \in \text{star}_k^i$. Furthermore, $H_1$ and $H_2$ satisfy the peak condition then.

If $i \geq 1$ then $\{H_1, H_2\}$ is uniquely determined by the splitset (up to the order on the sources of the new hyperedge).

**Remark.** Observe that the last is not true for $k$-splitting in general, since the splitset may divide $H$ into more than two connected components. Furthermore, there can be hyperedges whose set of sources is just the splitset, so that one is free to put them into either $H_1$ or $H_2$.

**Proof.** From Definition 2.13 it follows quite directly that, every $k$-splitset $S$ for $H$ must be of the form $\text{centre} \cup q \cup q'$ for some nonadjacent peaks $q, q' \in \text{peaks}_H$, implying that $H_1$ and $H_2$ are again of this form. Since this preserves the peaks, the two components satisfy the peak condition as claimed.

If $i \geq 1$, $H - S$ consists of two connected components, by definition of $(i, k)$-stars. By definition of splitting, two nodes which are not separated by $S$ belong either both to $H_1$ or both to $H_2$; so, we have that $V_{H_1}$ and $V_{H_2}$ are uniquely determined by $S$. It remains to be shown that a similar statement holds for hyperedges, too. Since $q$ and $q'$ are nonadjacent, there is no hyperedge $e \in E_H$ with $[\text{sources}(e)] = S$. Thus, every hyperedge of $H$ is incident with at least one vertex of either $\text{inner}(H_1)$ or $\text{inner}(H_2)$ and, hence, $E_H \cap E_H = \{e \in E_H \mid [\text{sources}(e)] \cap \text{inner}(H_i) \neq \emptyset\}$ for $i = 1, 2$. □

As an example, we may again consider the $(2, 4)$-star given in Fig. 5. It has two splitsets (the two diagonals), and both consist of the union of two peaks and the (empty) centre. Deleting four nodes out of three rather than two peaks does not disconnect this hypergraph, because then at most one of the peaks gets totally deleted. Together with the fact that $(i, k)$-triples are $(k+i)$-connected (they cannot be disconnected since all their nodes are directly connected with each other), the observation that the above $(2,4)$-star has two different splitsets gives rise to the following interesting corollary.

**Corollary 2.17** (Ambiguously splitting $k$-connected $k$-hypergraphs). For every $i \in \mathbb{N}$, $1 \leq i \leq k/2$, there is a $k$-connected $k$-hypergraph $H$ which has two different $k$-splitsets splitting it into $H_1$ and $H_2$ and into $H'_1$ and $H'_2$, but all of $H_1, H_2, H'_1$ and $H'_2$ are $(k+i)$-connected and do not split any further.

**Proof.** Obviously, our above observation can be generalised in the sense that every $(i, k)$-star ($1 \leq i \leq k/2$), obtained by merging two $(i, k)$-triples has two different splitsets:
the sets $\text{centre}_H \cup q \cup q'$, where $q$ and $q'$ are nonadjacent peaks. Both of these split it into two $(i, k)$-triples again. As we noted above, the resulting $(i, k)$-triples are indeed $(k+i)$-connected, which proves the claim. □

3. Collapsed $k$-split decompositions are unique

The purpose of this section is to prove that collapsed $k$-split decompositions of $k$-connected $k$-hypergraphs as defined below are unique.

**Definition 3.1 ($k$-split decomposition).** Let $H$ be a hypergraph.

A $k$-split decomposition of $H$ is any set $S$ where either $S = \{H\}$ or there is some $k$-split decomposition $S'$ of $H$ and some $H' \in S'$ $k$-splitting into hypergraphs $H_1$ and $H_2$ with new hyperedge $\text{new}$, such that

1. $S = S' - \{H'\} \cup \{H_1, H_2\}$ and
2. for all $H'' \in S'$ we have $\text{new} \notin E_{H''}$.

$S$ is total if $\text{split}_k(H') = \emptyset$ for all $H' \in S$.

**Remarks.**

- The second requirement above just means that the new hyperedges occurring in a $k$-split decomposition shall be chosen distinct from each other and from the old ones. So, the result of merging all the elements of $S$ back into one again is always the original hypergraph $H$.

- A $k$-split decomposition is called general if all splittings applied in order to obtain it are general ones.

Because of the size requirement (see Definition 2.7), repeated splitting will always lead to a total $k$-split decomposition finally. By the third part of Lemma 2.10, if $H$ is $k$-connected, we also know what components these total $k$-split decompositions are made of: they consist of $(i, k)$-triples and $k$-boxes. By Corollary 2.17, however, total $k$-split decompositions of $k$-connected $k$-hypergraphs are not unique. In order to get unique decompositions, we want to recollapse certain parts of the total $k$-split decomposition. Of course, there is always a trivial way of doing so. Just collapse all the components back into one, thus ending up with $\{H\}$, which is clearly unique. So, our aim must in fact be a bit more ambitious: we want to find a notion of collapsed $k$-split decompositions where as few components as possible are collapsed, but which yields unique results. By Corollary 2.17, we know that we do at least have to collapse $(i, k)$-triples into $(i, k)$-stars as far as possible. We will show that this is already enough. Thus, our definition of collapsed $k$-split decompositions is the following.

**Definition 3.2 (Collapsed $k$-split decomposition).** Let $H$ be a hypergraph and let $S$ be a total $k$-split decomposition of $H$. A collapsed $k$-split decomposition is a $k$-split decomposition that can be built up from $S$ by repeatedly replacing two $(i, k)$-stars $H_1, H_2 \in S$ satisfying the peak condition by $H_1 \langle H_2 \rangle$ (as long as this is possible).
**Example 3.3 (Collapsed 3-split decomposition).** Consider the hypergraph depicted in Fig. 7. We may first split off the leftmost hypergraph in Fig. 8 (which is a (1, 3)-star) with new hyperedge $e$ and then the middle one (a (0, 3)-star) with new hyperedge $e'$. The remaining hypergraph is the rightmost one, which is a 3-box.

Obviously, every collapsed $k$-split decomposition is a $k$-split decomposition. Because of Lemma 2.14, every collapsed $k$-split decomposition of a $k$-connected $k$-hypergraph consists of $k$-boxes and $(i, k)$-stars.

The elements of a $k$-split decomposition $S$ can be arranged as the nodes of a tree $T(S)$, the split tree associated with $S$, according to the relation "$H_1$ contains a hyperedge of $H_2$", as follows.

**Definition 3.4 (Split tree).** Let $S$ be a $k$-split decomposition of some hypergraph $H$.

The $k$-split tree associated with $S$ is the tree $T(S)$ for which the following holds:

1. $N_{T(S)} = S$, i.e., the nodes of $T(S)$ are the hypergraphs in $S$.
2. If $H' \in S$ then neigh$_{T(S)}(H') = \{ H'' \in S \mid E_{H'} \cap E_{H''} \neq \emptyset \}$.

Note that $T(S)$ is indeed a tree since we required that the new hyperedges introduced when constructing a split decomposition are chosen appropriately. We will show that collapsed $k$-split trees are unique up to similarity as defined below.

**Definition 3.5 (Similarity of split trees).** Let $H$ be a hypergraph.

![Fig. 7. A 3-hypergraph to be decomposed.](image)

![Fig. 8. The collapsed 3-split decomposition of the hypergraph in Fig. 7.](image)
Two split trees \( T \) and \( T' \) of \( H \) are similar if they are isomorphic via some isomorphism \( f \), and for each \( H' \in \mathcal{N}_T \) there is a weak isomorphism \( b \) between \( H' \) and \( f(H') \), with

1. \( b_v \) being the identity on \( V_{H'} \),
2. \( b_e(e) = e \) for all \( e \in E_H \cap E_{H'} \), i.e., \( b_e \) is the identity on all "old" hyperedges, and
3. the weak isomorphisms chosen for different components agree on common hyperedges.

Two split decompositions are similar if the split trees they define are similar.

According to the definition, similarity means that the hypergraphs occurring in both decompositions are not only isomorphic, they are even identical, except for the new hyperedges. (Observe that this does already imply that the two split trees are isomorphic, i.e., the relationship between different components is the same in both decompositions.)

We are now able to formulate the uniqueness theorem.

**Theorem 3.6** (Uniqueness of collapsed \( k \)-split trees). General collapsed \( k \)-split trees of \( k \)-connected \( k \)-hypergraphs are unique up to similarity.

In order to prove Theorem 3.6, we need the notion of a region. We will show that a \( k \)-connected \( k \)-hypergraph \( H \) uniquely divides into nonoverlapping regions, which are \((i,k)\)-stars and \( k \)-boxes. Actually, this is the major part of the work; the proof of the theorem gets rather easy then.

**Definition 3.7** (Regions). Let \( H \in \mathcal{H}_k \) be \( k \)-connected.

1. We define an equivalence relation \( =_H \) on hypergraphs in \( \text{split}_k(H) \). If \( H_1, H_2 \in \text{split}_k(H) \) with new hyperedges \( \text{new}_1 \) and \( \text{new}_2 \) then \( H_1 =_H H_2 \) if and only if \((V_{H_1}, E_{H_1} \setminus \{\text{new}_1\}) = (V_{H_2}, E_{H_2} \setminus \{\text{new}_2\}) \) and \([\text{sources}({\text{new}_1})] = [\text{sources}({\text{new}_2})] \). The equivalence class of \( H_1 \) with respect to \( =_H \) is denoted by \([H_1]_H \).

2. For every \( R \in \text{split}_k(H) \), \([R]_H \in \text{reg}(H) \) if and only if there is some \( TYPE \in \{\text{box}_k \} \cup \{\text{star}_{\leq i} | i \in \mathbb{N}, 0 \leq i \leq k/2 \} \) (the type of \( R \)) such that \( R \in TYPE \) and there is no \( R' \in \text{split}_k(H) \cup \{H\} \) with \( R' \in TYPE \) and \( R \in \text{split}_k(R') \).

**Remark.** Two hypergraphs in \([R]_H \) are more or less equal. They may only differ with respect to the order on the sources of the new hyperedge. Intuitively, a region is an \((i,k)\)-star or a \( k \)-box that splits off the given hypergraph and is maximal in the sense that it is not contained in a larger one of the same type which also splits off.

**Definition 3.8** (Overlapping). Let \( H \) be a hypergraph and let \( H_1, H_2 \in \text{split}_k(H) \) with distinct new hyperedges \( \text{new}_1 \) and \( \text{new}_2 \), respectively.

\( H_1 \) and \( H_2 \) overlap if \( V_{H_1} \cap V_{H_2} \neq \emptyset \) or \([\text{sources}(\text{new}_1)] \cap [\text{sources}(\text{new}_2)] \neq \emptyset \) or \( E_{H_1} \cap E_{H_2} \neq \emptyset \).

Distinct regions overlap if and only if their sets of nodes overlap (hence, we do not have to pay attention to \( E_{H_1} \cap E_{H_2} \), any more), by the following lemma. Furthermore, if one region contained another one (which is indeed impossible for distinct regions, as
we shall show), the new hyperedge of the greater region could not be incident with an inner node of the smaller one.

**Lemma 3.9.** Let $H \in \mathcal{H}_k$ be $k$-connected and let $[[R_0]]_H, [[R_1]]_H \in \text{reg}(H)$ be distinct, where $R_1$ and $R_2$ have distinct new hyperedges $\text{new}_0$ and $\text{new}_1$, respectively.

1. $R_0$ and $R_1$ overlap if and only if $V_{R_0} \cap V_{R_1} \notin [[\text{sources}(\text{new}_0)]] \cap [[\text{sources}(\text{new}_1)]]$.
2. If $V_{R_1} \subseteq V_{R_0}$ then $[[\text{sources}(\text{new}_0)]] \cap \text{inner}(R_1) = \emptyset$.

**Proof.** (1) We have to show that the existence of some hyperedge $e \in E_{R_0} \cap E_{R_1}$ does already imply that $V_{R_0} \cap V_{R_1} \notin [[\text{sources}(\text{new}_0)]] \cap [[\text{sources}(\text{new}_1)]]$. Assume, to the contrary, that $V_{R_0} \cap V_{R_1} \subseteq [[\text{sources}(\text{new}_0)]] \cap [[\text{sources}(\text{new}_1)]]$. Clearly, $e \in E_{R_0} \cap E_{R_1}$ does then imply that $e$ is parallel with $\text{new}_0$ and $\text{new}_1$. Since regions of type $\text{star}_k^0$ are the only ones containing parallel hyperedges, this means that $R_0, R_1 \in \text{star}_k^0$. But then, by maximality, we have $E_{R_0} \cap E_H = \{e \in E_H | [[\text{sources}(e)]] = V_{R_0}\} = E_{R_1} \cap E_H$; so, $R_0 \cong H R_1$, violating the assumption that $[[K_0]]_H \neq [[R_1]]_H$.

(2) Assume that $v \in [[\text{sources}(\text{new}_0)]] \cap \text{inner}(R_1)$. If $V_{R_0} = V_H$ then $H!R_0 \in \text{star}_k^0$, i.e., there are at least two distinct hyperedges $e, e' \in E_H$ which are parallel with $\text{new}_0$. But $v \in \text{inner}(R_1)$ implies both $e, e' \in E_{R_1}$, and $|V_{R_1}| > k$, which is impossible because no regions other than $(0, k)$-stars contain parallel hyperedges. If $V_{R_0} \neq V_H$, there is a node $v' \in V_H - V_{R_0}$ such that $v$ and $v'$ are adjacent. (Otherwise, $[[\text{sources}(\text{new}_0)]] - v$ would also disconnect $H$, contradicting $k$-connectedness.) But since $v \in \text{inner}(R_1)$, this means that $v' \in V_{R_1}$ and, hence, $V_{R_1} \not\subset V_{R_0}$.

Concerning overlapping regions $[[R_0]]_H$ and $[[R_1]]_H$, we have two special cases. The first one is the case where $|V_{R_0}| = k$, i.e., $R_0 \in \text{star}_k^0$, and the second one is the converse, $V_{R_0} = V_H$, i.e., $H!R_0 \in \text{star}_k^0$. Since we do not always want to have to pay attention to these cases, we first prove that overlapping is impossible here. (Of course, the situation is symmetric, i.e., the same holds for $R_1$, since the definition of overlapping is symmetric.)

**Lemma 3.10.** Let $H$ be a $k$-connected $k$-hypergraph and let $[[R_0]]_H, [[R_1]]_H \in \text{reg}(H)$ be distinct. If $R_0$ and $R_1$ overlap, then $k < |V_{R_0}| < |V_H|$.

**Proof.** We have to show that $R_0, H!R_0 \notin \text{star}_k^0$. Let the corresponding splitsets for $R_0$ and $R_1$ be $S_0$ and $S_1$ and suppose first that $R_0 \in \text{star}_k^0$. By the definition of overlapping, there must be some $v \in S_0 \cap \text{inner}(R_1)$ (so, in particular, $R_1 \notin \text{star}_k^0$). Since $R_0 \in \text{star}_k^0$, there must be at least two distinct hyperedges $e, e' \in E_H$ incident with $v$. But then $R_1$ contains parallel hyperedges (namely, $e$ and $e'$), which is impossible since $R_1 \notin \text{star}_k^0$.

If $H!R_0 \in \text{star}_k^0$, nodes are adjacent in $H$ if and only if they are so in $R_0$. Hence, we have that $R_1$ splits off $R_0$ because it splits off $H$. By Lemma 2.16, the type of $R_0$ equals that of $R_1$ then, which is impossible by maximality of regions.
By the preceding lemma it remains to show that regions in \( \text{reg}(H)^- \) defined by

\[
\text{reg}(H)^- = \{ [R]_H \in \text{reg}(H) | k < |V_R| < |V_H| \}
\]
cannot overlap.

Intuitively, two regions \([R_0]_H, [R_1]_H \in \text{reg}(H)^-\) cannot overlap because, otherwise, we would get that each of them can be split further using the splitset splitting the other one off \(H\), which would contradict the maximality of \((i,k)\)-stars (by Lemma 2.16) or the \((k+1)\)-connectedness of \(k\)-boxes. However, there are the new hyperedges in \(R_0\) and \(R_1\), which do not exist in \(H\). Since these hyperedges (say \(\text{new}_0\) and \(\text{new}_1\)) can directly connect vertices which are not adjacent in \(H\), we can only claim that, e.g., \(R_1\) gets disconnected by deletion of \(V_{R_1} \cap [\text{sources}(\text{new}_0)] \cup [\text{sources}(\text{new}_1)]\), but \(V_{R_1} \cap [\text{sources}(\text{new}_0)]\) alone may not suffice. This, of course, does not help, since these new splitsets may contain more than \(k\) nodes. As Corollary 2.17 states, this kind of situation can really occur. On the other hand, this was proved by looking at two overlapping \((i,k)\)-stars forming a bigger one together, which is not very serious, as long as there are no other situations where this does also occur. In order to show this, we will first prove a – rather technical – lemma stating that, if we assume to have overlapping regions, suitable intersections and unions of their splitsets can be defined such that all of them are splitsets of size \(k\). Later on, we will use this in order to show that, indeed, only \((i,k)\)-stars can overlap; hence, regions cannot overlap at all, because this would contradict their maximality, by Lemma 2.16.

**Lemma 3.11.** Let \(H\) be a \(k\)-connected \(k\)-hypergraph and let \([R_0]_H, [R_1]_H \in \text{reg}(H)^-\) with corresponding splitsets \(S_0\) and \(S_1\).

Furthermore, let \(Q_L = \text{inner}(R_0) - V_{R_1}, Q_R = \text{inner}(R_1) - V_{R_0}, Q_T = V_H - (V_{R_0} \cup V_{R_1})\) and \(Q_B = \text{inner}(R_0) \cap \text{inner}(R_1)\) as depicted in Fig. 9. Then the following hold:

1. Let \(S_L = S_0 \cap S_1, S_T = S_0 - V_{R_1}, S_R = S_1 - V_{R_0}, S_B = S_1 \cap \text{inner}(R_0)\) and \(S_{BR} = S_0 \cap \text{inner}(R_1)\) be the five subsets \(S_0 \cup S_1\) consists of. If both regions overlap, then \(|S_L| = |S_T| = |S_R| = |S_{BR}| \geq 1\).

2. Let \(S_L = S_0 \cup S_T \cup S_B, S_R = S_0 \cup S_T \cup S_{BR}, S_T = S_\cap \cup S_T \cup S_{TR}\) and \(S_B = S_\cap \cup S_B \cup S_{BR}\). For all \(x \in \{L, R, T, B\}\) the set \(S_x\) separates all nodes of \(Q_x\) from all of \(V_H - (S_x \cup Q_x)\).

**Proof.** The second assertion follows directly from the fact that \(S_0\) and \(S_1\) are splitsets, since this means that vertices \(v \in Q_x, v' \in Q_y\) cannot be adjacent for \(x, y \in \{L, R, T, B\}\), \(x \neq y\).

For the first claim we consider three cases, where the first two will be shown to be impossible. By symmetry, we need not consider the cases where only the roles of \(R_0\) and \(R_1\) are exchanged.

1. \(V_{R_0} \subseteq V_{R_1}\). By the second part of Lemma 3.9, this means that \(S_1 \cap \text{inner}(R_0) = \emptyset\). Thus, every two nodes \(v_0 \in \text{inner}(R_0), v_1 \in V_{R_1}\) are adjacent in \(R_1\) if and only if they are adjacent in \(H\). Hence, \(S_0\) separates the nodes of \(\text{inner}(R_0)\) from those of \(V_{R_1} - V_{R_0}\) in \(R_1\) since it does so in \(H\). Due to the fact that \(R_0 \notin \text{split}_0\), we have \(\text{inner}(R_0) \neq \emptyset\). Also, \(V_{R_1} \subseteq V_{R_0}\) since \(V_{R_0} = V_{R_1}\), would mean that \(S_0 = S_1\), i.e., \(R_0 =_H R_1\), by the second
part of Lemma 3.9. Hence, $V_{R_1} - V_{R_0} \neq \emptyset$; so, $R_0$ splits off $R_1$. By Lemma 2.9, this is only possible if $R_1$ is not $(k+1)$-connected, i.e., $R_1 \in \textit{star}_k^i$ for some $i, 1 \leq i \leq k/2$. But then $R_0 \in \textit{star}_k^i$, too, by Lemma 2.16, violating the maximality of regions.

(2) $V_{R_0} - V_{R_1}, V_{R_1} - V_{R_0} \neq \emptyset$ (as opposed to the above) and $S_0 \subseteq V_{R_0}$. Since the new hyperedge of $R_0$ is not incident with any node of $V_{R_0} - V_{R_1}, S_1 \cap V_{R_0}$ separates the nodes in $V_{R_0} \cap \text{inner}(R_1)$ from those in $V_{R_0} - V_{R_1}$ in $R_0$ since $S_1$ does so in $H$. (Observe that both sets are nonempty because of the assumption and the definition of overlapping.) Thus, we get that $S_1 \subseteq V_{R_0}$ because $R_0$ is $k$-connected, i.e., the situation is symmetric: as $S_1$ disconnects $R_0, S_0$ disconnects $R_1$, since $S_0 \subseteq V_{R_0}$ as well as $S_1 \subseteq V_{R_0}$.

If $\textit{new_0}$ and $\textit{new_1}$ are the new hyperedges of $R_0$ and $R_1$, respectively, let

$$R_\cap = (V_{R_0} \cap V_{R_1}, E_{R_0} \cap E_{R_1} \cup \{\textit{new_0}, \textit{new_1}\}).$$

By the first part of Lemma 3.9, there is some $v \in V_{R_0} \cap \text{inner}(R_1)$. Since $v \in \text{inner}(R_1), \textit{all hyperedges of} H \textit{incident with} v \textit{are in} E_{R_1}$. Since there are at least two hyperedges incident with $v$ in $R_0$, but only $\textit{new_0}$ is not in $E_{R_0}$, we have $E_{R_0} \cap E_{R_1} \neq \emptyset$. Hence, $|E_{R_1}| \geq 3$, i.e., $R_\cap$ splits off both $R_0$ and $R_1$. In that case, $R_0$ and $R_1$ cannot be $k$-boxes; so, there are $i, j, 1 \leq i, j \leq k/2$, such that $R_0 \in \textit{star}_k^i$ and $R_1 \in \textit{star}_k^j$. But by Lemma 2.16, $R_\cap$ is an $(i, k)$-star and a $(j, k)$-star; so, $i = j$. Moreover, Lemma 2.16 does also yield that the pairs $R_0 \cap R_\cap$ and $R_1 \cap R_\cap$, $R_\cap$ satisfy the peak condition. Thus, $R_0 \cap R_\cap \cap \textit{peak}_{R_0} = \textit{peak}_{R_1} \cap R_\cap$ since $\textit{peaks}_{R_0} = \textit{peaks}_{R_1} \cup \textit{peaks}_{R_\cap}$. Now we have three cases.

- $R_0 \cap R_1 \cap R_\cap = H$ contradicts maximality of $R_0$ and $R_1$ at once.

- If $V_{R_0} \cup V_{R_1} = V_H$, but there is a hyperedge $e \in E_H$ not in $E_{R_0} \cup E_{R_1}$, we have $\text{[sources}(e)\text{]} \cap (\text{inner}(R_0) \cup \text{inner}(R_1)) = \emptyset$; hence, $\text{[sources}(e)\text{]} \subseteq S_0 \cap S_1$, since $V_{R_0} \cup V_{R_1} = V_H$. But then $S_0 = S_1$ as $|\text{sources}(e)| = k$. However, this implies that $R_0 \cap R_\cap$ do not overlap: by Lemma 2.16, $R_0$ and $H \cap R_\cap$ are uniquely determined by $S_0$, and $R_1$ and $H \cap R_\cap$ are uniquely determined by $S_1$; so, $R_0 = H \cap R_1$ and $R_1 = H \cap R_0$. Surely, $R_0$ and $H \cap R_0$ cannot overlap.
Recognising $k$-connected hypergraphs in cubic time

If there is a node $v \in V_H - (V_{R_0} \cup V_{R_1})$, it is separated from $S_1 - S_0$ by $S_0$ and from $S_0 - S_1$ by $S_1$. Thus, every path from $v$ to a node of $S_1$ must use some node of $S_0$ and vice versa. This implies that $S_0 = S_1$; so, again $\llbracket R_0 \rrbracket_H$ and $\llbracket R_1 \rrbracket_H$ do not overlap, as above.

(3) $V_{R_0} - V_{R_1}, V_{R_1} - V_{R_0} \neq \emptyset, S_0 \not\subseteq V_{R_1}$ and $S_1 \not\subseteq V_{R_0}$. By assumption, $S_{TL}, S_{TR} \neq \emptyset$. We first show that $S_{BL}, S_{BR} \neq \emptyset$, too. If $Q_B \neq \emptyset$ then $S_B$ disconnects $H$ by the second part of the lemma. (Remember that we have proved this already.) Thus, $|S_B| \geq k$ by $k$-connectedness and, hence, $S_{BL}, S_{BR} \neq \emptyset$, since $S_{TL}, S_{TR} \neq \emptyset$ and $|S_0| = |S_1| = k$. If $Q_B = \emptyset$ then at least $S_{BL} \cup S_{BR} \neq \emptyset$ due to the fact that $Q_B \cup S_{BL} \cup S_{BR} \neq \emptyset$, by definition of overlapping. Assume that $v \in S_{BR}$. We have $|V_{R_0}| > k$, so, there is some $v' \in Q_L \cup S_{BL} \cup Q_B - Q_L \cup S_{BL}$. If $v' \in S_{BL}$, we are done. Otherwise, $v'$ is separated from $v$ by $S_L$ and, hence, $|S_L| \geq k$. On the other hand, $S_L = S_{BL} \cup S_0 - S_{BR}$; so, again $S_{BL} \neq \emptyset$.

It remains to show that these sets are all the same size.

Without loss of generality, we can assume that $Q_x \neq \emptyset$ for all $x \in \{L, R, T, B\}$. To see this, suppose, e.g., $Q_L = \emptyset$. Then for every $v_0 \in S_{TL}$, we would have to have a node $v_1 \in S_{BL}$ adjacent to $v_0$ by some hyperedge $e \in E_H$ since, otherwise, $S_0 - \{v_0\}$ would separate $v_0$ from every vertex $v \in S_{BL}$. Since $S_0, S_1$ and $S_L$ are splitsets, we have $|\text{sources}(e)| \subseteq S_L$.

This means that if $H'$ with $|V_{H'}| > k$ is any $k$-connected $k$-hypergraph merging with $H$ along $e$ then $H \langle H' \rangle$ is $k$-connected, by Lemma 2.10, and the splitsets under consideration remain the same in $H \langle H' \rangle$, but $Q_L \neq \emptyset$ now (with respect to $H \langle H' \rangle$).

So, we could continue considering $H \langle H' \rangle$ instead of $H$. For $Q_R, Q_T$ and $Q_B$ the situation is similar.

Let $k' = k - |S_\gamma|$. Because of the fact that $|S_0| = k$ we have

$$|S_{TL}| + |S_{BR}| = k' = |S_{TR}| + |S_{BL}|. \tag{2}$$

On the other hand, $S_T$ separates $Q_T$ from the rest (remember that $Q_T \neq \emptyset$); hence, $|S_{TL}| + |S_{TR}| \geq k'$. Similarly, we get $|S_{TL}| + |S_{BL}| \geq k'$ and, thus, $2|S_{TL}| + |S_{TR}| + |S_{BL}| \geq 2k'$, which means that $2|S_{TR}| \geq k'$, by (2). In a similar way, we obtain $2|S_{BR}| \geq k'$; so, in fact, $|S_{TL}| = |S_{BR}| = k'/2$ again by (2). Clearly, the same applies to $S_{TR}$ and $S_{TL}$ and, therefore, $|S_{BL}| = |S_{BR}| = |S_{TL}| = |S_{TR}| = k'/2$. \hfill \qed

We now present the main lemma of this section.

**Lemma 3.12.** Distinct regions of a $k$-connected $k$-hypergraph cannot overlap.

**Proof.** The proof uses Lemma 3.11 in order to show that the existence of overlapping regions would contradict the maximality of $(i, k)$-star regions and the $(k + 1)$- connectedness of regions being $k$-boxes.

Let $\llbracket R_0 \rrbracket_H, \llbracket R_1 \rrbracket_H \in \text{reg}(H)$. By Lemma 3.10, we can assume that $\llbracket R_0 \rrbracket_H, \llbracket R_1 \rrbracket_H \in \text{reg}(H)^-$ with distinct new hyperedges. If $R_0$ and $R_1$ overlapped, we would get the situation of Lemma 3.11; so, let us take the notations from there. There are two cases.
One of $R_0, R_1$, say $R_0$, is a $k$-box. By Lemma 3.11, we have that $V_{R_0} \subseteq S_0 \cup S_B$, because every additional node could be disconnected from others using either $S_L$ or $S_B$. But, since $|S_L| = |S_B| = k$ and $k$-boxes do not contain parallel hyperedges, there can be at most one hyperedge incident with nodes in $S_L$ and one incident with nodes of $S_B$. Hence, $R_0$ contains only three hyperedges, but every $k$-box does at least contain four by definition.

(2) $R_0 \in \text{star}_i^k$ and $R_1 \in \text{star}_j^k$. By Lemma 2.16, we can assume that $R_0$ is an $(i, k)$-triple and $R_1$ is a $j$-triple since, otherwise, we could split off $Q_L, Q_R$ and $Q_B$, using Lemma 3.11 in order to obtain such a situation. Lemma 3.11 yields $i = |S_{BL}| = |S_{BR}| = j$. So, the $k$-hypergraph we can split off using the splitset $S_T$ is weakly isomorphic to the one shown in Fig. 10 (up to relabelling) and is, hence, an $(i, k)$-star properly containing $R_0$ and $R_1$, a contradiction. □

We are now ready to prove Theorem 3.6.

**Proof of Theorem 3.6.** If $\text{reg}(H) = 0$ then $H$ is a $k$-box or an $(i, k)$-star itself and for every collapsed $k$-split decomposition $S$ we have $S = \{H\}$. Otherwise, let $H_1$ be the first element split off to obtain a total $k$-split decomposition for $H$. Because of Lemma 3.12, there is a unique region $[R]_H$ of $H$ such that $H_1 \in \text{split}_k(R)$ or $H_1 =_H R$. If $H_1 \in \text{split}_k(R)$, this implies that $R \in \text{star}_i^k$ for some $i, 0 \leq i \leq k/2$, since $k$-boxes cannot be $k$-split at all. In this case we have $R, R! H_1, H_1 \in \text{star}_i^k$, by Lemma 2.16. Therefore, by the definition of collapsed $k$-split decompositions, every such region $[R]_H$ will be contained in a single component of every collapsed $k$-split decomposition of $H$, as $(i, k)$-stars get merged again as far as possible there. Hence, for every collapsed $k$-split decomposition $S$ of $H$,

$$S! R = \begin{cases} S - \{R\} & \text{if } R' =_H R \text{ for some } R' \in S, \\ S - H' \cup H'! R & \text{otherwise, where } H' \in S \text{ and } R \in \text{split}_k(H'), \end{cases}$$

is well-defined. Clearly, $S! R$ is a collapsed $k$-split decomposition of $H! R$. As it has fewer hyperedges than $H$, we can now proceed, by induction on $|E_H|$, to conclude that, if $S'$ is another collapsed $k$-split decomposition of $H$, $S! R$ and $S'! R$ are unique up to similarity; hence, so are $S$ and $S'$. □

Fig. 10. Overlapping $(i, k)$-stars are not maximal.
4. Computing collapsed $k$-split decompositions

In this section we will be concerned with the problem of computing collapsed $k$-split decompositions. Since we want to use Theorem 3.6 in order to construct for a given $k$-hypergraph a derivation tree, we have to be able to find a collapsed $k$-split decomposition first. However, there is no need for us to perform this task in all its generality. As we shall see later, the hypergraphs of a fixed hyperedge-replacement language (generated by a hyperedge-replacement grammar of order $k$) can only have a finite set of $k$-boxes as components. This means that we can assume that there is only a finite number of hypergraphs (up to weak isomorphism) that can occur in the total $k$-split decomposition we have to construct.

In order to develop our algorithm, we need the following lemma.

**Lemma 4.1.** Let $H \in \mathcal{H}$ and let $S$ be a split decomposition of $H$. Then there is a sequence $H_1, \ldots, H_n$ of hypergraphs such that $S = \{H_1, \ldots, H_n\}$ and, for $i = 1, \ldots, n-1$, we have that $H_i$ splits off $H'_i$, where $H'_i = H_i \setminus H_{i+1}$ for $i = 1, \ldots, n-1$, and $H_n = H_n$.

**Proof.** By induction on $|H|$, it suffices to show that there is some $H_1 \in S \setminus \text{split}_k(H)$ if $|S| > 1$. By definition of splitting (and of split decompositions), this is the case if there is some $H_1 \in S$ with $|E_{H_1} - E_H| = 1$. Since every splitting introduces exactly one new hyperedge present in both components, i.e., $\sum_{H \in S} |E_{H'} - E_H| = 2(|S| - 1)$, there are at least two such components in $S$.  \[ \square \]

Let us now show how total $k$-split decompositions can be computed in time $O(|H|^2)$. (Observe that $O(|H|^2) = O((|V_H| + k|E_H|)^2) = O((|V_H| + |E_H|)^2)).$

**Lemma 4.2** (Computing total $k$-split decompositions). Let $B$ be the closure under weak isomorphism of a finite subset of $\text{box}_k$.

There is an algorithm that produces for every $k$-connected $k$-hypergraph $H$ a general and total $k$-split decomposition $S$ with $S \cap \text{box}_k \subseteq B$, or rejects if there is no such $k$-split decomposition. The algorithm runs in time $O(|H|^2)$.\(^4\)

**Proof.** The labelling of hyperedges has no influence on the problem; so, let us assume that $H$ is unlabelled. Due to Theorem 3.6 and the definition of collapsed $k$-split decompositions, we have that $S \cap \text{box}_k$ is unique up to the ordering on the sources of new hyperedges, which is arbitrary. Thus, if there is a general and total $k$-split decomposition $S$ with $S \cap \text{box}_k \subseteq B$ then any arbitrary one has this property. By Lemma 4.1, this means that we may construct $S$ by splitting any $(i, k)$-triple or $k$-box

\(^4\) Hopcroft and Tarjan (cf. [10]) gave an algorithm for the case $k = 2$ which is better in two respects: it applies to all 2-connected graphs, and it runs in linear time. It would be interesting to see whether their algorithm can be generalised to our case, but we will not try here, since the rest of our algorithm is cubic anyway.
$H_1$ off $H$, then splitting another component $H_2$ off $H!H_1$, and so on. Since $|H| > |H!H_1| > \ldots$, the length of this loop is bounded from above by $|H|$. Therefore, it is sufficient to show that we can find $H_1$ in time $O(|H|)$. We divide this task into two steps. First we try to find some parallel hyperedges. If we succeed, we split off a $(0,k)$-triple. If not, we continue searching for another component.

The search for parallel hyperedges can be done in linear time if we take, e.g., the following representation for $E_H$. Let $Q = \{ \{sources(e)\} | e \in E_H\}$ and consider a list which includes for every $q \in Q$ one entry containing the set $\{sources(e) | e \in E_H$ and $[sources(e)] = q\}$. Clearly, we can find parallel hyperedges in linear time now by looking for an entry including more than one source list.

As for the other components, let $\mathcal{C}$ be a finite set of representatives of the weak isomorphism classes of unlabelled hypergraphs in $B \cup \bigcup_{1 \leq i \leq k/2} \{H' \mid H'$ an $(i,k)$-triple\}. Let the *degree* of a vertex $v$ in a hypergraph $H'$ be given by $\text{deg}_{H'}(v) = |\{e \in E_H \mid v \in [sources(e)]\}|$. Furthermore, let $\text{deg}_{\text{max}} = \max_{H' \in \mathcal{C}, v \in V_{H'}} \text{deg}_{H'}(v)$ and $s_{\text{max}} = \max_{H' \in \mathcal{C}} |E_{H'}|$. If $H_1 \in \text{split}_k(H)$ is weakly isomorphic to some hypergraph in $\mathcal{C}$ and $v \in \text{inner}(H_1)$, we have that

1. $\text{deg}_{H'}(v) = \text{deg}_{H_1}(v) \leq \text{deg}_{\text{max}}$, since every $e \in E_H$ incident with an inner node of $H_1$ in $H$ must be in $E_{H'}$ and

2. every vertex $v' \in V_{H_1}$ is reachable from $v$ in $H_1$ on a $vv'$-path $p$ with $\text{int}(p) \subseteq \text{inner}(H_1)$. (In particular, it is then reachable on the same path in $H$ since it does not use the new hyperedge.)

The latter holds because $H_1$ is $(k+1)$-connected and $|V_{H_1} \setminus \text{inner}(H_1)| = k$. We now define a subhypergraph $sub_H(v)$ of $H$, for every $v \in V_{H}$, which we can restrict our attention to if we want to decide whether some $H_1$ as above, with $v \in \text{inner}(H_1)$, splits off. Let $sub_H(v)$ be given by the following:

(i) $v \in sub_H(v)$.

(ii) For every path $p = v_1 \ldots e_nv_n$ in $H$ such that $\max_{v' \in \text{int}(p)} \text{deg}_{H'}(v') \leq \text{deg}_{\text{max}}$ and $n \leq s_{\text{max}}, v_1, \ldots, v_n \in V_{sub_H(v)}$ and $e_1, \ldots, e_n \in E_{sub_H(v)}$.

(iii) $sub_H(v)$ contains no other nodes and/or hyperedges.

Clearly, $|sub_H(v)|$ is bounded by a constant and $sub_H(v)$ can be computed in constant time by considering all relevant paths. Also, the set

$$\text{outer}_H(v) = \{v' \in V_{sub_H(v)} \mid \text{deg}_{H}(v) > \text{deg}_{\text{max}}\}$$

can be computed while computing $sub_H(v)$. (Observe that these vertices can only occur as end vertices of the path $p$ in (ii).) Let $H_1$ be weakly isomorphic to some hypergraph in $\mathcal{C}$ now. Due to (1) and (2), we have that it splits off $H$ with splitset $V$ and $v \in \text{inner}(H_1)$ if and only if $H_1 \in \text{split}_k(sub_H(v))$ and $V_{H_1} \cap \text{outer}_H(v) \subseteq V$.

There is only a finite number of hypergraphs of a given size (up to (weak) isomorphism). Therefore, the set $\{ sub_H(v) \mid H$ an (unlabelled) hypergraph and $v \in V_{H}\}$ is finite up to isomorphism. Hence, we may test whether some $H_1$ weakly isomorphic to a hypergraph in $\mathcal{C}$ splits off $sub_H(v)$ in constant time, and a loop over all $v \in V_{H}$ yields the desired result: either we find such an $H_1 \in \text{split}_k(sub_H(v))$ or we get that there is none and reject. $\square$
Theorem 4.3 (Computing collapsed k-split decompositions). Let $B$ be the closure under weak isomorphism of a finite subset of box$_k$. There is an algorithm that produces for every k-connected k-hypergraph $H$ a general and collapsed k-split decomposition $S$ with $S \cap$ box$_k \subseteq B$, or rejects if there is no such k-split decomposition. The algorithm runs in time $O(|H|^2)$.

Proof. Let $S_T$ be a total k-split decomposition of $H$, which we can compute in quadratic time, by Lemma 4.2. By definition, we can construct $S$ from $S_T$ by merging as far as possible all $(i, k)$-stars that satisfy the peak condition. As we observed in the proof above, $S_T$ has at most a linear number of elements. While constructing $S_T$ we can easily compute some extra information. We remember every new hyperedge constructed, together with the two components it appears in and its peaks with respect to both of these components. Also, we memorise the type of each component. Now we just have to consider each new hyperedge the two hypergraphs it is part of being $(i, k)$-stars both, decide whether the peak condition is satisfied (which is easy since we remembered the peaks) and merge them if necessary. Note that we need not update the information about the peaks since they stay the same. Merging the two components, thus, takes constant time since it merely consists of deleting the hyperedge along which the merging shall take place. Since the number of new hyperedges is linear, we can thus construct $S$ from $S_T$ in time $O(|H|)$. \qed

5. Hyperedge-replacement grammars

We now define our notion of hyperedge-replacement grammars by means of merging. It is obvious that the procedure of merging two hypergraphs $H$ and $H'$ along a hyperedge $e$ is strongly related to the usual notion of hyperedge replacement (cf. [9]). There we would say that $e$ is replaced by $(V_H, E_H - \{e\})$ in $H$. So, $H'$ serves as a kind of production in this situation: the hyperedge $e$ is the "left-hand side" and $(V_H', E_H' - \{e\})$ is the "right-hand side". However, this view depends on $H$. If $H''$ is another hypergraph, it may merge with $H'$ along a hyperedge $e' \neq e$. In order to define grammars, we need productions whose left-hand sides are fixed, i.e., independent of the context in which the production is applied. Therefore, productions are defined as hypergraphs together with a distinguished hyperedge.

Definition 5.1 (Production, cf. [18]). A hyperedge-replacement production is a pair $p = (e, R)$, where $R$ is a hypergraph and $e \in E_R$. The hyperedge $e$ is the left-hand side $\text{lhs}(p)$ of $p$ and $(V_R, E_R - \{e\})$ is its right-hand side, denoted by $\text{rhs}(p)$.

The underlying hypergraph $U(p)$ of $p$ is defined by $U(p) = R$.

Remark. Vogler [18] does not distinguish between graphs and productions. He considers a sort of graphs having a special, so-called virtual hyperedge. So, our productions do in fact compare with his notion of graphs (generalised to the hypergraph case).
We consider two productions \( p \) and \( p' \) isomorphic if \( U(p) \equiv U(p') \) via an isomorphism \( b \) with \( b(\text{lhs}(p)) = \text{lhs}(p') \). In this case we use the notation \( b(p) \) to denote \( p' \).

**Definition 5.2 (Hyperedge replacement, cf. [9]).** Let \( H \) be a hypergraph, let \( e \in E_H \) and let \( p = (e', R) \) be a hyperedge-replacement production with \( \text{lab}(e) = \text{lab}(e') \) and \( |\text{sources}(e)| = |\text{sources}(e')| \). Then the hyperedge replacement \( H[p \leftarrow e] \) is defined by

\[
H[p \leftarrow e] = b(H) \langle b'(U(p)) \rangle,
\]

where \( b \) and \( b' \) are any isomorphisms such that the merging is defined and \( b_E(e) = b_E(e') \).

**Remark.** Note that, in contrast to merging, hyperedge replacement yields unique results only up to isomorphism. Nevertheless, we will mostly write \( H' = H[p \leftarrow e] \) instead of \( H' \equiv H[p \leftarrow e] \).

Using the notion of hyperedge replacement, we define hyperedge-replacement grammars as usual.

**Definition 5.3 (Hyperedge-replacement grammars, cf. [9]).** Let \( \mathcal{P} \) be a set of productions and let \( H_1, \ldots, H_n \) be hypergraphs for some \( n \geq 0 \).

1. \( H_1 \) directly derives \( H_2 \) by a production \( p \in \mathcal{P} \) in a derivation step \( H_1 \Rightarrow_H H_2 \), if \( H_2 = H_1[p \leftarrow e] \) for some \( e \in E_{H_1} \). If the production we actually use does not matter, we also write \( H_1 \Rightarrow_H H_2 \) or even \( H_1 \Rightarrow H_2 \), if \( \mathcal{P} \) is understood from the context.

2. A derivation (of length \( n \)) in \( \mathcal{P} \) is a sequence of derivation steps

\[
H_1 \Rightarrow_H H_2 \Rightarrow_H \cdots \Rightarrow_H H_{n+1}.
\]

\( H_{n+1} \) is then said to be derivable from \( H_1 \) (in \( \mathcal{P} \)), and this is denoted by \( H_1 \Rightarrow^n \mathcal{P} H_{n+1} \), where we may again omit \( \mathcal{P} \). If the length of the derivation does not matter, we write \( H_1 \Rightarrow^* \mathcal{P} H_{n+1} \).

3. A hyperedge-replacement grammar \( G \) is a pair \( (\mathcal{P}_G, A_G) \), where
- \( \mathcal{P}_G \) is a finite set of productions, and
- \( A_G \) is any hypergraph, called the axiom of \( G \).

\( G \) is of order \( k \) if \( |\text{sources}(e)| \leq k \) for all \( e \in \{\text{lhs}(p) \mid p \in \mathcal{P}_G \} \).

A hypergraph is derivable in \( G \) if it is derivable from \( A_G \) in \( \mathcal{P}_G \). \( \mathcal{L}(G) \), the language generated by \( G \), is the set of all these hypergraphs, i.e.,

\[
\mathcal{L}(G) = \left\{ \Pi \in \mathcal{H} \mid A_G \Rightarrow^*_\mathcal{P}_G \Pi \right\}.
\]

Since hyperedge-replacement is context-free (in a way to be made precise below), we may infer from a set \( \mathcal{P} \) of productions new ones whose application comprises the effect of whole derivations into one step. As usual, we get these new productions by applying \( \mathcal{P} \) to the right-hand sides of productions in \( \mathcal{P} \) themselves.
Definition 5.4 (P-form). (1) Let \( p = (e, R) \) and \( p' \) be productions and let \( e' \in E_{\text{rhs}}(p) \). Then the hyperedge replacement \( p[e' \leftarrow p'] \) is defined if \( R[e' \leftarrow p'] \) is. If \( R[e' \leftarrow p'] = b(R) \left< b'(U(p')) \right> \) (where \( b \) and \( b' \) are the required isomorphisms), we define
\[
p[e' \leftarrow p'] = (b(e), R[e' \leftarrow p']).
\]
Accordingly, the notions of derivation steps and derivations are extended to productions.

(2) Let \( \mathcal{P} \) be a set of productions. The set \( \mathcal{P}^* \) of all \( \mathcal{P} \)-forms is the transitive closure of \( \mathcal{P} \) under \( \Rightarrow \) given by
\[
\mathcal{P}^* = \left\{ p \mid p' \Rightarrow p \text{ for some } p' \in \mathcal{P} \right\}.
\]

Remark. Note that \( p[e' \leftarrow p'] \) is only defined for \( e' \in \text{E}_{\text{lhs}}(p) \), i.e., \( e' \neq \text{lhs}(p) \).

One of the most important properties of hyperedge-replacement languages is their context-freeness.

Fact 5.5 (Context-freeness lemma, cf. [8]). Let \( H \) and \( H' \) be hypergraphs.

(1) If \( \{e_1, \ldots, e_n\} \subseteq E_H, n \geq 0 \), then, for all productions \( p_1, \ldots, p_n \) and every permutation \( \pi \) on \( \{1, \ldots, n\} \),
\[
H[e_1 \leftarrow p_1] \cdots [e_n \leftarrow p_n] = H[e_{\pi(1)} \leftarrow p_{\pi(1)}] \cdots [e_{\pi(n)} \leftarrow p_{\pi(n)}].
\]
(Hence we write \( H[(e_i \leftarrow p_i)]_{i=1}^n \) in the following, where \( H[(e_i \leftarrow p_i)]_{i=1}^n = H \) for \( n=0 \).)

(2) If \( \mathcal{P} \) is a set of productions and \( n \in \mathbb{N} \) then \( H \Rightarrow^*_\mathcal{P} H' \) if and only if there are hyperedges \( e_1, \ldots, e_m \in E_H \) and productions \( p_1, \ldots, p_m \) such that
- \( H' = H[e_1 \leftarrow p_1] \cdots [e_m \leftarrow p_m] \),
- \( p_1, \ldots, p_m \in \mathcal{P}^* \) via derivations of length \( n_1, \ldots, n_m \), and
- \( m + \sum_{i=1}^m n_i = n \).

Thus, derivations for hypergraphs can be divided into subderivations originating from the replaced hyperedges in an already derived hypergraph. In particular, this means that there are derivation trees for derivations in hyperedge-replacement grammars (see [8]).

For every hyperedge-replacement grammar \( G \), there is a hyperedge-replacement grammar \( G' \) with \(|E_A| \geq 3\) for all \( R \in \{A_i \} \cup \{U(p) \mid p \in \mathcal{P}_A\} \), such that
\[
\mathcal{L}(G') = \{ H \in \mathcal{L}(G) \mid |E_H| \geq 3 \}
\]
(see [8]⁵). Recall that we are interested in recognising \( k \)-connected \( k \)-hypergraphs, and that there are no more than a finite number of nonisomorphic \( k \)-connected

⁵ In that setting, this means that there are neither "chain productions" (the ones with \(|E_{U(p)}| = 2\) nor "empty productions" (those with \(|E_{U(p)}| = 1\).
k-hypergraphs with fewer than 3 hyperedges. Thus, we can assume, without loss of
generality, that all productions we have to deal with have at least three hyperedges in
their right-hand sides, and also \(|E_{A_i}| \geq 3\).

Observe that this makes every derivation step \(H \Rightarrow H'\) reversible by \(k\)-splitting,
i.e., \(H \equiv H': R\) for some isomorphic copy \(R\) of \(U(p)\) (see Lemma 2.10).
So, derivations in hyperedge-replacement grammars are very closely related to split
decompositions of the derived graph. In particular, we may define derivation trees as
a special sort of split trees.

**Definition 5.6** (derivation tree, cf. [8]). Let \(G\) be a hyperedge-replacement grammar
such that for all \(R \in A_G \cup \{U(p) | p \in A_G\} \) \(|E_R| \geq 3\), and let \(p = (e, H)\) be a production.

Let \(T\) be a split tree for \(H\) such that there is some distinguished root \(R_T \in N_T\) and for
all \(H' \in N_T - \{R_T\}\) with \(E_{H'} \cap E_{\text{pred}(H)} = \{e'\}\), there is some \(p' \in A_G\) with \((e', H') \equiv p'\).

1. \(T\) is a derivation tree for \(H\) in \(G\) if \(R_T \equiv A_G\).
2. \(T\) is a derivation tree for \(p\) over \(A_G\) if there is some \(p' \in A_G\) with \((e, R_T) \equiv p'\).

Using the context-freeness lemma it is not so hard to show that a hypergraph \(H\) is
derivable in \(G\) if and only if there is a derivation tree for \(H\) in \(G\) (see [8]). Similarly,
a production \(p\) is in \(\mathcal{P}^*\) if and only if there is a derivation tree for \(p\) over \(\mathcal{P}\) (which is, in
turn, the case if and only if there is a derivation tree for \(p\) over \(\mathcal{P}^*\)).

To end this section, let us show a lemma which is important for our recognition
algorithm. The general idea underlying this recognition algorithm is to reduce the
question whether \(H \in \mathcal{L}(G)\) to the question whether the general and collapsed \(k\)-split
tree for \(H\) can be transformed into a derivation tree for \(H\) in \(G\). One difficulty in this
approach is that derivation trees have a root (the axiom), whereas split trees do not. If
we once have this root, we can apply a bottom-up algorithm to the tree, but the root
may be hard to find since many components can be isomorphic to the axiom (up to
relabelling). Lemma 5.7 enables us to consider a set of \(\mathcal{P}\)-forms instead of \(\mathcal{L}(G)\) where
we can choose an arbitrary root instead of searching for the root.

The lemma and its proof involve some relabelling of hyperedges. We will have to
"mark" certain hyperedges. So, let, for every label \(A, \tilde{A}\) be a copy, i.e., a new label, and
denote the marked version of a hyperedge \(e\) by \(\langle e \rangle\), that means, \(\text{sources}(\langle e \rangle) = \text{sources}(e)\) and \(\text{lab}(\langle e \rangle) = \text{lab}(e)\). For a hypergraph \(H\) with \(e_1, \ldots, e_n \in E_H\), let

\[ H\langle e_1, \ldots, e_n \rangle = (V_H, E_H \cup \{\langle e_1 \rangle, \ldots, \langle e_n \rangle \} - \{e_1, \ldots, e_n\}) \]

and for a production \(p\) with \(e \in E_{\text{rhs}(p)}\) define

\[ p\langle e \rangle = (\langle e \rangle, U(p)\langle e, \text{lhs}(p) \rangle). \]

So, in the case of productions we mark \(\text{lhs}(p)\) and \(e\), and \(\langle e \rangle\) instead of \(\langle \text{lhs}(p) \rangle\) is the
left-hand side of the resulting production.
Lemma 5.7. Let $G$ be a hyperedge-replacement grammar. Then there is a finite set $\mathcal{P}$ of productions such that for every hypergraph $H$ all of whose hyperedges are unmarked, and for every $e \in E_H$,

$$H \in \mathcal{L}(G) \text{ if and only if } (\langle e \rangle, H \langle e \rangle) \in \mathcal{P}^*.$$ 

Remark. By the above, we can decide whether $H \in \mathcal{L}(G)$ by choosing any hyperedge $e \in E_H$ and deciding whether the production $(\langle e \rangle, H \langle e \rangle)$ is in $\mathcal{P}^*$. Thus, we can make use of the fact that the component corresponding to the root of a derivation tree for a given production $p$ rather than a hypergraph, is uniquely determined. It is just the component containing $lhs(p)$.

Proof. The construction yielding $\mathcal{P}$ is based on the following idea. Consider the derivation tree for a hypergraph $H \in \mathcal{L}(G)$ with a derivation $A_G \xrightarrow{p_1} \cdots \xrightarrow{p_n} H$. The hyperedge $e$ we want to make the left-hand side of $(\langle e \rangle, H \langle e \rangle)$ stems from the right-hand side of some production $p_i$ in the derivation tree (if it is not a hyperedge of the axiom). If $e$ becomes the left-hand side of a derived production, we have to take a derivation starting with $p_i$, but where $e$ instead of $lhs(p_i)$ is the left-hand side. Then, of course, $lhs(p_i)$ must become a “normal” hyperedge. This means that if $p_j$ is the node preceding $p_i$ in the derivation tree, $p_i$ cannot be applied to $p_j$ any more. Instead, we would like to apply $p_j$ to $p_i$. So, we have to repeat the whole procedure with $p_j$: take the hyperedge of $p_j$ which was formerly replaced by $p_i$ and make this one the left-hand side. Again, $lhs(p_j)$ must become an ordinary hyperedge then, and everything goes on with the node preceding $p_j$. Eventually, when we arrive at the root $A_G$ of the derivation tree, this procedure stops by making the hyperedge $e'$ of $A_G$ which was replaced in the original derivation the left-hand side of a production $(e', A_G)$.

Intuitively, what happens with the derivation tree is that the path leading from $A_G$ to $p_i$ gets reversed by swapping the roles of left-hand sides and replaced hyperedges. Of course, we must not only introduce new productions which can be applied together with the old ones – we have to take care not to mix up the original productions with the new ones. This we do using our copied set of labels. Any hyperedge $e$ which gets turned from a normal one into a left-hand side, or vice versa, gets marked. So, let, for $p \in \mathcal{P}_G$,

$$\mathcal{P}_p = \{ p \langle e \rangle | e \in E_{\text{rhs}(p)} \}$$

and define

$$\mathcal{P} = \mathcal{P}_G \cup \bigcup_{p \in \mathcal{P}_A} \mathcal{P}_p \cup \{(\langle e \rangle, A_G \langle e \rangle) | e \in E_{A_G} \}.$$ 

To prove that $\mathcal{P}$ satisfies our needs, observe first that for productions $p$ and $p'$ with $e \in E_{\text{rhs}(p)}$ and $e' \in E_{\text{rhs}(p')}$ we have

$$p[e \leftarrow p'] \langle e' \rangle = p' \langle e' \rangle [\langle lhs(p') \rangle \leftarrow p \langle e \rangle],$$

(4)
by definition of hyperedge-replacement and commutativity of merging. For the same reason we get, for a hypergraph \( H \) with \( e \in E_H \),

\[
\langle \langle e' \rangle, H[e \rightarrow p'] \langle \langle e' \rangle \rangle \rangle = p' \langle \langle e' \rangle \rangle \langle \langle \text{lhs}(p') \rangle \rangle \rightarrow \langle \langle e' \rangle, H \langle \langle e \rangle \rangle \rangle.
\]  
(5)

In a first step we show that, for all productions \( p \) and all hyperedges \( e \in E_{rhs(p)} \):

if \( p \) is a \( P \)-form then \( \langle p \rangle \) is a \( P \)-form.

(6)

We proceed by induction on the length of derivations. If \( p \in P^\ast \), by context-freeness there is some \( p' \in P \) with hyperedges \( e_1, \ldots, e_n \in E_{rhs(p')} \) and \( P \)-forms \( p_1, \ldots, p_n \) having shorter derivations, such that \( p = p'[(e_i \leftarrow p_i)_{i=1,\ldots,n}] \). There are two cases:

1. If \( e \in E_{rhs(p')} \) then \( e \neq e_i \) for \( i = 1, \ldots, n \). Hence,

\[
p \langle e \rangle = p' \langle e \rangle \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \in P^\ast,
\]

since \( p' \langle e \rangle \in P \) and \( p_1, \ldots, p_n \in P^\ast \subseteq P^\ast \).

2. Otherwise, we must have \( e \in E_{rhs(p)} \) for some \( j, 1 \leq j \leq n \). Using (4) we get

\[
p'[(e_j \leftarrow p_j)] \langle e \rangle = p_j \langle e \rangle \langle (\text{lhs}(p_j)) \rangle \rightarrow p' \langle e_j \rangle.
\]

By induction hypothesis, \( p_j \langle e \rangle \) is a \( P \)-form; so, \( p'[(e_j \leftarrow p_j)] \langle e \rangle \in P^\ast \) since \( p' \langle e_j \rangle \in P \).

Now \( p \langle e \rangle \in P^\ast \), as required, for

\[
p \langle e \rangle = p' \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \langle e \rangle
\]

\[
= p'[(e_j \leftarrow p_j)] \langle e \rangle \langle (e_i \leftarrow p_i)_{i=1,\ldots,n,i \neq j} \rangle
\]

and \( p_1, \ldots, p_n \in P^\ast \subseteq P^\ast \).

This ends the proof of (6). To prove the lemma, consider some hypergraph \( H \) with no marked hyperedge and some hyperedge \( e \in E_H \).

Suppose that \( H \in \mathcal{L}(G) \). By context-freeness there are hyperedges \( e_1, \ldots, e_n \in E_{A_0} \) and productions \( p_1, \ldots, p_n \in P^\ast \) such that \( H = A_0 \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \).

If \( e \in E_{A_0} \) then

\[
\langle \langle e \rangle, H \langle e \rangle \rangle = \langle \langle e \rangle, A_0 \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \langle e \rangle \rangle
\]

\[
= \langle \langle e \rangle, A_0 \langle e \rangle \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \rangle
\]

\[
= \langle \langle e \rangle, A_0 \langle e \rangle \rangle \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \in P^\ast
\]

since \( \langle \langle e \rangle, A_0 \langle e \rangle \rangle \in P \) and \( p_1, \ldots, p_n \in P^\ast \subseteq P^\ast \).

If \( e \in E_{rhs(p)} \) for some \( j, 1 \leq j \leq n \), then \( p_j \langle e \rangle \in P^\ast \) by (6). Hence, we have

\[
\langle \langle e \rangle, H \langle e \rangle \rangle = \langle \langle e \rangle, A_0 \langle (e_i \leftarrow p_i)_{i=1,\ldots,n} \rangle \langle e \rangle \rangle
\]

\[
= \langle \langle e \rangle, A_0 \langle e_j \rangle \rangle \langle (e_i \leftarrow p_i)_{i=1,\ldots,n,i \neq j} \rangle
\]

\[
= p_j \langle e \rangle \langle (\text{lhs}(p_j)) \rangle \rightarrow \langle \langle e_j \rangle, A_0 \langle e_j \rangle \rangle \langle (e_i \leftarrow p_i)_{i=1,\ldots,n,i \neq j} \rangle
\]

\[\text{by (5)}\]

\[
\in P^\ast.
\]
For the other direction, let \( \langle e \rangle, H \langle e \rangle \in \mathcal{P}^* \). Again, we proceed by induction on the length of derivations. Let \( p \in \mathcal{P} \) and \( e_1, \ldots, c_n \in E_{rhl(p)} \), such that there are productions \( p_1, \ldots, p_n \in \mathcal{P}^* \) with \( \langle e \rangle, H \langle e \rangle \equiv p[(e_i \leftarrow p_i)]_{i=1, \ldots, n} \).

(1) If \( p = \langle e \rangle, A_G \langle e \rangle \rangle \) then \( e_1, \ldots, e_n \) are unmarked; so, \( p_1, \ldots, p_n \in \mathcal{P}^* \) (by definition of \( \mathcal{P} \) all productions \( p \in \mathcal{P}^* \) have marked left-hand sides). This does immediately imply that \( H = A_G[(e_i \leftarrow p_i)]_{i=1, \ldots, n} \in \mathcal{L}(G) \).

(2) Otherwise, let \( p = p' \langle e \rangle \) for some \( p' \in \mathcal{P}_G \). Since \( \langle e \rangle \) is the only marked hyperedge in \( H \langle e \rangle \) but \( p \) contains \( \langle lhl(p') \rangle \), in addition, we must have \( \langle lhl(p') \rangle = e_j \) for some \( j, 1 \leq j \leq n \). Furthermore, \( lhl(p_j) \) must then be the only marked hyperedge in \( p_j \). Let \( lhl(p_j) = \langle e' \rangle \) and let \( H' = (V_{U(p_j)}, E_{rhl(p_j)} \cup \{e'\}) \). By the above, \( H' \) does not contain any marked hyperedge and the induction hypothesis applies, as \( \langle e' \rangle, H' \langle e' \rangle \rangle = p_j \in \mathcal{P}^* \). So, \( H' \in \mathcal{L}(G) \) and we get

\[
H \langle e \rangle = U(p)[(e_i \leftarrow p_i)]_{i=1, \ldots, n} = U(p') \langle e, lhl(p') \rangle [(e_i \leftarrow p_i)]_{i=1, \ldots, n} = U(p') \langle lhl(p') \rangle [(e_i \leftarrow p_i)]_{i=1, \ldots, n} \langle e \rangle
\]

and hence,

\[
H = U(p') \langle lhl(p') \rangle [(e_i \leftarrow p_i)]_{i=1, \ldots, n} = U(p') \langle lhl(p') \rangle \langle (e_j \leftarrow p_j), H' \rangle [(e_i \leftarrow p_i)]_{i=1, \ldots, n, i \neq j} = H' \langle e_j \leftarrow p_j \rangle [(e_i \leftarrow p_i)]_{i=1, \ldots, n, i \neq j} \in \mathcal{P}^*_G,
\]

since \( H' \in \mathcal{L}(G) \), \( p' \in \mathcal{P}_G \) and \( p_i \in \mathcal{P}^*_G \) for \( i = 1, \ldots, n, i \neq j \). \( \Box \)

6. Recognising \( k \)-connected \( k \)-hypergraphs

The main theorem of this paper is the following one.

Theorem 6.1 (Cubic time recognition). Let \( G \) be a hyperedge-replacement grammar of order \( k \).

There is an algorithm running in time \( O(|H|^3) \) which decides for every \( k \)-connected \( k \)-hypergraph \( H \) whether \( H \in \mathcal{L}(G) \).

Note that a derivation in a hyperedge-replacement grammar of order \( k \) deriving a \( k \)-connected \( k \)-hypergraph cannot use any production \( p \) whose right-hand side contains a \( k' \)-hyperedge for \( k' \neq k \). Hyperedges with more than \( k \) sources cannot be replaced any more if they once appear in a derived hypergraph, and if productions are used whose right-hand sides contain \( k' \)-hyperedges, \( k' < k \), then these must eventually be replaced by some \( k \)-hypergraph. Hence, the final result \( k' \)-splits and both components contain at least one inner node, i.e., the derived hypergraph is not \( k \)-connected.
Note also that, due to Lemma 2.16, if \( H_1 \in \text{star}_k^1 \) and \( H_1 \Rightarrow H_2 \) by a hyperedge replacement \( H_1 [e \in p] \), then \( H_2 \in \text{star}_k^1 \) if and only if \( U(p) \in \text{star}_k^1 \) and \( e \) and \( \text{lhs}(p) \) satisfy the peak condition.

The general idea we use to prove Theorem 6.1, is much the same as the one Vogler uses (cf [18]). Therefore, we will discuss the coarse-grained structure only informally and concentrate on the new things. By Lemma 5.7, it suffices to show that the set \( \{ p \in \mathcal{P}^* | U(p) \text{ a k-connected k-hypergraph} \} \) can be recognised in cubic time for every finite set \( \mathcal{P} \) of productions of order \( k \). As we already observed, the underlying hypergraph of every production in a derivation finally yielding a k-connected k-hypergraph must itself be a k-connected k-hypergraph. Thus, we may assume that \( \mathcal{P} \) has only productions whose underlying hypergraphs are k-connected k-hypergraphs, since other productions must not be applied anyway. Furthermore, we can assume that \( U(p) \) is a k-box or an \((i,k)\)-triple for some \( i, 0 \leq i \leq k/2 \), because \( \mathcal{P} \) can be turned into such a form: by (3), we may assume that \( |E_{U(p)}| \geq 3 \) for all \( p \in \mathcal{P} \). As long as there is some \( p = (e, R) \in \mathcal{P} \), which is neither an \((i,k)\)-triple nor a k-box, consider hypergraphs \( R_1 \) and \( R_2 \) such that \( R \) \( k \)-splits into \( R_1 \) and \( R_2 \) with new hyperedge \( e' \). We choose the label of \( e' \) distinct from all others appearing in \( \mathcal{P} \). Suppose that \( e \in E_{R_1} \). Then replacing \( p \) by \( (e, R_1) \) and \( (e', R_2) \) does not change \( \mathcal{P}^* \) [up to those productions containing a hyperedge labelled with the new label \( \text{lab}(e') \)] and repeating this process of dividing productions into smaller ones does obviously lead to a production set of the required type.

Because of the context-freeness lemma, a production \( p \) is a \( \mathcal{P} \)-form if and only if there is a derivation tree for \( p \) over \( \mathcal{P} \). If this derivation tree exists, it is a total \( k \)-split tree for \( U(p) \), because of our assumptions about \( \mathcal{P} \). Thus, we can collapse it, obtaining a derivation tree for \( p \) over \( \mathcal{P}^* \) with nodes in \( \text{box}_k \cup \bigcup_{0 \leq i \leq k/2} \text{star}_k^i \). Let us call this kind of derivation tree a collapsed derivation tree for \( p \) over \( \mathcal{P} \). If \( U(p) \) is \( k \)-connected, Theorem 3.6 applies. So, the general, collapsed \( k \)-split tree for \( U(p) \) and its collapsed derivation tree are similar, i.e., they differ in at most two respects.

(1) The direction of the new hyperedges, i.e., the order on their sources, can be chosen arbitrarily for the collapsed split tree. This does not hold for derivation trees in general since the components must be isomorphic to productions.

(2) \( T \) lacks the labelling of left-hand sides, i.e., all new hyperedges are labelled \( \tau \).

To handle the first problem, we complete the production set by introducing equivalent rules, for all possible redirections of hyperedges. This means that we replace every production \( p \) by the set \( C(p) \) of all productions obtainable from \( p \) by substituting each hyperedge \( e \in E_{U(p)} \) by some hyperedge \( e_\pi \), such that \( \pi \) is a permutation on \( \{1, \ldots, k\} \), \( \text{lab}(e_\pi) = (\text{lab}(e), \pi) \) and \( \text{sources}(e_\pi) = e_{\pi(1)} \cdots e_{\pi(k)} \). Call the set of productions we thus get from \( \mathcal{P} \) the complete form \( C(\mathcal{P}) \), i.e., \( C(\mathcal{P}) = \bigcup_{p \in \mathcal{P}} C(p) \). The following lemma is just a more general version of the corresponding one by Vogler (see [18]).

**Lemma 6.2 (Equivalence of a production set and its complete form).** Let \( p \) be a production and let \( p' \in C(p) \). Then \( p \) is a \( \mathcal{P} \)-form if and only if \( p' \) is a \( C(\mathcal{P}) \)-form.
We do not prove this lemma, because the proof is almost the same as in the case treated by Vogler.

Let \( p' \in C(p) \). By Lemma 6.2 and the uniqueness of collapsed \( k \)-split trees, we have \( p \in \mathcal{P}^* \) if and only if the new hyperedges of the collapsed split tree \( T \) of \( U(p') \) [or \( U(p) \), if we identify \( \text{lab}(e) \) with \( (\text{lab}(e), \text{id}) \)] can be labelled in such a way that \( T \) becomes a collapsed derivation tree for \( p' \) over \( C(\mathcal{P}) \). Hence, all we need is an algorithm recognising the collapsed components, i.e., those productions whose underlying hypergraphs are \((i, k)\)-stars. Then we can decide whether \( p \in \mathcal{P}^* \) by computing the collapsed \( k \)-split tree of \( U(p) \) and transforming it into a collapsed derivation tree over \( C(\mathcal{P}) \). The first task can be performed in quadratic time, as we already know, and the latter can be done by computing in a bottom-up way the set of possible labels for each of the left-hand sides. [Observe that \( k \)-boxes, which may also occur in \( T \), are no problem at all, because every such \( k \)-box must be isomorphic to one of the finitely many productions of \( C(\mathcal{P}) \).] Finally, we say “yes” if (and only if) \( \text{lab}(\text{lhs}(p')) \) turns out to be a possible label of \( \text{lhs}(p') \).

Up to this point, everything has been similar to the investigations Vogler made, so—as in the special case \( k = 2 \)—the running time of our algorithm is essentially determined by the time we need to recognise the \((i, k)\)-stars occurring in \( T \). [Observe that \( \sum_{H \in N_i} |V_H| \) and \( \sum_{H \in N_i} |E_H| \) are linear in \( |U(p)| \), as the number of components of a \( k \)-split decomposition is at most linear in \( |H| \) (see Lemma 4.1). Hence, testing all the \((i, k)\)-stars in \( T \) one after the other will not take more time than recognising one \((i, k)\)-star which is as large as \( U(p') \).]

What remains to be done is to show that the language \( \{ p \in \mathcal{P}^* \mid U(p) \in \text{star}_i \} \) can be recognised in cubic time for all \( i, \ 0 \leq i \leq k/2 \). For \( i = 0 \), this means recognising a semilinear set, by Parikh’s Theorem (see [7]), which is a problem solvable in linear time [6]; so, let us assume that \( i > 0 \). Intuitively, although a hyperedge in an \((i, k)\)-star is incident with \( k \) vertices, it does not incorporate more information than just a normal edge (i.e., a 2-hyperedge), for \([\text{sources}(e)]\) is determined by any two nodes from the different peaks of \( e \), and we can remember the ordering on \( \text{sources}(e) \) using more detailed labels. Therefore, we can give an equivalent grammar with \( k = 2 \) in which the underlying hypergraphs of all productions are \((1, 2)\)-triples (so-called triangles) whose nodes represent the peaks of the original productions. This is what we will do in order to prove the following lemma.

**Lemma 6.3** (Cubic time recognition of \((i, k)\)-star productions). Let \( \mathcal{P} \) be some finite set of productions, and let \( i \) be a natural number, \( 1 \leq i \leq k/2 \).

There is an algorithm running in time \( |U(p)|^3 \) which decides, for every production \( p \), whether \( U(p) \in \text{star}_i \) and \( p \in \mathcal{P}^* \).

**Proof.** First of all, observe that it is sufficient to give an algorithm applying to inputs \( p \) with \( U(p) \in \text{star}_i \), because it is very easy to determine whether a given hypergraph is an \((i, k)\)-star. Just choose an arbitrary hyperedge and start “walking” around the star, thereby checking whether the hypergraph satisfies the definition of \((i, k)\)-stars.
Because of Lemma 2.16, it suffices to consider some fixed \( i \), \( 1 \leq i \leq k/2 \), and we can assume that the underlying hypergraphs of productions in \( \mathcal{P} \) are \((i, k)\)-triples. To prove the lemma, we will give a construction transforming \( \mathcal{P} \) into an equivalent production set whose underlying hypergraphs are triangles. This kind of production set does essentially compare with a set of context-free string productions (see [18]). This proves our cubic bound, by the well-known theorem of Cocke, Kasami and Younger (cf. [20, 11]).

Construction. Let \( s_{pi} \) denote the set of all hyperedge-replacement productions \( p \) with \( U(p) \in \text{star}_i^k \). Our construction works by removing the centres of the \((i, k)\)-triples and replacing the peaks by only one vertex each. Consider some production \( p \in s_{pi} \) and let \( R = U(p) \). Since we know that every hyperedge \( e \in E_R \) is incident with every node of the centre, we can as well delete these nodes from their source lists, and use more detailed labels to remember the numbers of the sources from \( \text{centre}_R \).

Similarly, we can represent the two peaks adjacent to \( e \) by two nodes connected by an ordinary edge, where we again remember the source numbers in the label. This is possible if we use an ordering \( g(q) \) of \( q \) for every \( q \in \text{peaks}_R \cup \{ \text{centre}_R \} \) that helps us to store the information \( g(q)_1 = \text{sources}(e)_{j_1}, g(q)_2 = \text{sources}(e)_{j_2}, \ldots, g(q)_q = \text{sources}(e)_{j_q} \) as the list \( j_1 j_2 \ldots j_q = \text{num}_e(g(q)) \). Of course, we have to consider all such orderings, which means that a production will become replaced by the set of all productions modified this way. So, every \((i, k)\)-triple production will be replaced by a set of productions whose underlying hypergraphs are triangles.

Let us call a function \( g \) an ordering for \( R \) if it assigns an ordering \( g(q) \) of \( q \) to every \( q \in \text{peaks}_R \cup \{ \text{centre}_R \} \). For every such ordering \( g \) and every hyperedge \( e \in E_R \) with \( \text{peaks}_R(e) = \{q_1, q_2\} \) such that \( \min\{\text{num}_e(q_1 \cup q_2)\} = q_1 \), we define \( f_g(e) \) by

\[
\begin{align*}
\bullet & \quad \text{sources}(f_g(e)) = q_1, q_2, \text{ and} \\
\bullet & \quad \text{lab}(f_g(e)) = (\text{lab}(e), \text{num}_e(g(q_1)), \text{num}_e(g(\text{centre}_R)), \text{num}_e(g(q_2))).
\end{align*}
\]

We let \( f_g(R) = (\text{peaks}_R, f_g(E_R)) \) and \( f_g(p) = (f_g(\text{lhs}(p)), f_g(R)) \).

As an example, consider the \((2, 7)\)-star in Fig. 11. If we order the nodes of its centre from the left to the right and those of the peaks from the inside to the outside, we get the triangle depicted in Fig. 12.

Let

\[
f(\mathcal{P}) = \{ f_g(p) \mid p \in \mathcal{P} \text{ and } g \text{ an ordering for } U(p) \}.
\]

Hence, we are done if we can prove the following claim.

**Claim.** Let \( p \in s_{pi} \) and let \( g \) be an ordering for \( U(p) \). Then \( p \) is a \( \mathcal{P} \)-form if and only if \( f_g(p) \) is an \( f(\mathcal{P}) \)-form.

Let us first show that, if we have \( p, p' \in s_{pi} \), a hyperedge \( e \in E_{U(p)} \), and an ordering \( g \) for \( U(p) \) then the following hold:

(i) \( p \equiv p' \) if and only if there is an ordering \( g' \) for \( U(p') \), such that \( f_g(p) \equiv f_{g'}(p') \).

---

\[6\] This is just a convention about the direction the resulting edge.
(ii) \( p \rightarrow p' \) is defined and \( e \) and \( \text{lhs}(p') \) satisfy the peak condition if and only if there is an ordering \( g' \) for \( U(p') \) such that \( f_g(p) [ f_g(e) \leftarrow f_g(p') ] \) is defined.

(iii) If \( e \) and \( \text{lhs}(p') \) satisfy the peak condition then there are orderings \( h \) and \( h' \) such that \( f_{h'}(p \leftarrow p') = f_h(p) [ f_h(e) \leftarrow f_h(p') ] \).

To prove the above statements, let \( R = U(p) \) and \( p' = (e', R') \).

Proof of (i): If \( p = b(p') \) for an isomorphism \( b \), it is straightforward to prove that \( V \rightarrow b_{V}^{-1} (g(b_{V}(V))) \) for \( V \in \text{peaks}(R) \cup \{ \text{centre}(R) \} \) defines an ordering for \( R' \) with the desired property. On the other hand, if two such orderings and an isomorphism \( b \) between \( f_g(p) \) and \( f_{g'}(p') \) are given, then \( g(q) \rightarrow g'(b_{V}(q)) \), for \( q \in \text{peaks}(R) \cup \{ \text{centre}(R) \} \), yields an isomorphism from \( R \) to \( R' \), because \( b \) preserves edge labels.

Proof of (ii): If \( p \rightarrow p' \) is undefined then \( \text{lab}(e) \neq \text{lab}(e') \) and, hence, \( \text{lab}(f_g(e)) \neq \text{lab}(f_{g'}(e')) \), i.e., \( f_g(p) [ f_g(e) \leftarrow f_g(p') ] \) is also undefined. The same holds if \( e \) and \( e' \) do not satisfy the peak condition. By definition, this means that \( \text{num}_e(\text{peaks}(e)) \neq \text{num}_e(\text{peaks}(e')) \); hence, \( \text{lab}(f_g(e)) \neq \text{lab}(f_{g'}(e')) \). On the other hand, if \( p \rightarrow p' \) is defined and \( e \) and \( e' \) satisfy the peak condition, we have \( \text{lab}(e) = \text{lab}(e') \), \( \text{num}_e(\text{peaks}(e)) = \text{num}_e(\text{peaks}(e')) \) and \( \text{num}_e(\text{centre}(R)) = \text{num}_e(\text{centre}(R')) \). Therefore, we may define \( g' \) in such a way that

\[
\text{num}_e(g'(\text{peaks}(e'))) = \text{num}_e(g(\text{peaks}(e)))
\]

and

\[
\text{num}_e(g'(\text{centre}(R))) = \text{num}_e(g(\text{centre}(R))).
\]

Then, by definition, \( \text{lab}(f_{g'}(e')) = \text{lab}(f_g(e)) \), implying that \( f_g(R) [ f_g(e) \leftarrow f_g(R') ] \) is defined.

Proof of (iii): It suffices to show that if \( e \) and \( e' \) satisfy the peak condition and \( g \) is an ordering for \( R \) and \( R' \) then

\[
f_g(R) \leftarrow f_g(R') = f_{g'}(R \leftarrow R').
\]
Let $E_R \cap E_{R^*} = \{e\}$. We have

$$f_g(R) \langle f_g(R') \rangle = (V_{\mathcal{L}(R)} \cup V_{\mathcal{L}(R')}, E_{\mathcal{L}(R)} \cup E_{\mathcal{L}(R')} - \{e\})$$

$$= (\text{peaks}_R \cup \text{peaks}_{R'}, f_g(E_R \cup E_{R'} - \{e\}))$$

$$= (\text{peaks}_{R \cup R'}, f_g(E_{R \cup R'}))$$

$$= f_g(R \cup R').$$

This ends the proof of (i)–(iii). Note that, on the right-hand side of the equivalence in (ii), there is no peak condition to be satisfied. This is because all triangles do automatically satisfy this condition.

To prove the claim, we proceed by induction on the length of the derivation. Because of the context-freeness lemma (and by Lemma 2.16) we have that $p \in \mathcal{S}_p$ if and only if there is a production $p' \in \mathcal{P}$ with hyperedges $e_1, \ldots, e_n \in E_{\mathcal{R}_{\mathcal{H}}(p)}$ and there are productions $p_1, \ldots, p_m \in \mathcal{P}^*$ having shorter derivations such that

- $p = p'[(e_i \mapsto p_i),_{1 \leq i \leq m}]$, and
- $U(p), U(p_1), \ldots, U(p_m)$ are $(i, k)$-stars such that $e_j$ and $\text{lhs}(p_j)$ satisfy the peak condition for all $j$, $1 \leq j \leq m$.

By (i)–(iii), this is the case if and only if there is a production $p' \in \mathcal{P}$ with hyperedges $e_1, \ldots, e_n \in E_{\mathcal{R}_{\mathcal{H}}(p')}$ and there are productions $p_1, \ldots, p_m \in \mathcal{P}^*$ having shorter derivations such that

- $f_{\mathcal{S}_p}(p) = f_{\mathcal{S}_p}(p')[(f_{\mathcal{S}_p}(e_i) \mapsto f_{\mathcal{S}_p}(p_i)),_{1 \leq i \leq m}]$ for some additional orderings $\theta', \theta_1, \ldots, \theta_n$, and
- $U(p), U(p_1), \ldots, U(p_m)$ are $(i, k)$-stars such that $e_j$ and $\text{lhs}(p_j)$ satisfy the peak condition for all $j$, $1 \leq j \leq m$.

(More precisely, because of (i) we have $p \in \mathcal{S}_p$ and $p = p'[(e_i \mapsto p_i),_{1 \leq i \leq m}]$ if and only if $p \in \mathcal{S}_p$ and $f_{\mathcal{S}_p}(p) = f_{\mathcal{S}_p}(p')[(e_i \mapsto p_i),_{1 \leq i \leq m}]$, which is, by (ii) and (iii), the case if and only if $f_{\mathcal{S}_p}(p) = f_{\mathcal{S}_p}(p')[(f_{\mathcal{S}_p}(e_i) \mapsto f_{\mathcal{S}_p}(p_i)),_{1 \leq i \leq m}]$.)

By induction hypothesis (and definition of $f(\mathcal{P})$), the above is true if and only if there is some $\hat{p} \in f(\mathcal{P})$ and there are hyperedges $\hat{e}_1, \ldots, \hat{e}_n \in E_{\mathcal{R}_{\mathcal{H}}(\hat{p})}$ and productions $\hat{p}_1, \ldots, \hat{p}_m \in f(\mathcal{P})^*$ such that $U(\hat{p}_i) \in \text{star}_{\frac{1}{2}}$ for $i = 1, \ldots, m$ and $f_{\mathcal{S}_p}(p) = \hat{p}[(\hat{e}_i \mapsto \hat{p}_i),_{1 \leq i \leq m}]$. This is – by context-freeness – the case if and only if $f_{\mathcal{S}_p}(p)$ is an $f(\mathcal{P})$-form as claimed.

This completes the proof of Lemma 6.3, hence the proof of Theorem 6.1. □

7. Discussion

We have seen that every hyperedge-replacement language of $k$-connected $k$-hyper graphs can be recognised in cubic time if it is generated by a hyperedge-replacement grammar of order $k$. This means that, in contrast to the algorithm by Lautemann [14], the running time does not depend on the order of the grammar (i.e., the degree of the polynomial does not). However, the constant factors, which will appear when $k$ becomes larger, seem to get quite big, since the number of hyperedge labels we used in
our construction increases along with \( k! \). But since these hyperedge labels were needed to remember the ordering on the sources of a hyperedge, we will probably not be able to find a much better construction, for this ordering on the sources is really necessary for the power of hyperedge-replacement grammars.

The algorithm is a proper generalisation of the one Vogler developed for cyclically connected graphs generated by edge-replacement [18]. In particular, the class of languages it can recognise includes all context-free string languages, because strings may be identified with cycle graphs having one unlabelled edge connecting the right and the left end. However, our algorithm is restricted to \( k \)-hypergraphs as inputs. For \( k = 2 \) this is no real disadvantage since 2-hypergraphs are just graphs and one is usually interested in graphs anyway. (Also, as mentioned below, this does just exclude 0- and 1-hyperedges, but no \( k' \)-hyperedges for \( k' > 2 \).)

It would be quite interesting to see whether one can weaken the restriction that only \( k \)-hypergraphs are looked at. In particular, one can ask whether languages of \( k \)-connected graphs generated by a hyperedge-replacement grammar of order \( k \) are also recognisable in polynomial time. [Observe that edges cannot be subject to replacement (except for (0, 2)-star productions) since this would destroy \( k \)-connectedness. So, every edge in a \( k \)-connected hypergraph is automatically “terminal” if \( k > 2 \).] A special case which might be easier to treat than this general question is whether one can allow for \( k \)-hyperedges to be replaced by complete graphs, as in the proof of Lemma 2.6. Unfortunately, the answer to this question is not clear at all, because it might be hard to find out which ones of the edges of a graph belong together. As an example, consider the 3-hypergraphs shown in Fig. 13. Replacing the hyperedges by triangles, we get the graph of Fig. 14 in both cases. Thus, we cannot decide whether, e.g., the edges of the triangle in the middle stem from the same hyperedge or not. (It might, perhaps, be an important observation that the two hypergraphs in Fig. 13 are isomorphic, but it does not seem to help directly.)

For the more general question the author would like to give the following conjecture.

---

**Fig. 13.** 3-hypergraphs

**Fig. 14.** The result of replacing 3-hyperedges in Fig. 13 by 3-cliques.
Conjecture 7.1. There is some $k \in \mathbb{N}$ such that there are NP-complete graph languages of $k$-connected graphs which can be generated by a hyperedge-replacement grammar of order $k$.

As opposed to the above, one can also ask for allowing $k'$-hyperedges to appear in a derived $k$-connected hypergraph, for $k' > k$. Clearly, our algorithm does also apply in this slightly more general situation if the considered hyperedge-replacement grammar is of order $k$. This is because a $k'$-hyperedge can only be part of a $k$-box (if we extend the definition of $k$-boxes appropriately), since it makes all its $k'$ sources adjacent. Thus, it is very easy to extend the uniqueness theorem in this respect and to use the generalised version for the recognition algorithm.

Our result indicates once more that connectedness is a very vital property for hyperedge-replacement languages to be tractable. If we just reduce the required connectedness by 1, we immediately run into NP-completeness for any arbitrary $k > 1$:

For every $k > 1$ there is a $(k - 1)$-connected hyperedge-replacement language of $k$-hypergraphs which is NP-complete.

This can be seen as follows. Using the NP-completeness result by Lange and Welzl [12], an easy argument shows that there are NP-complete languages of connected graphs which can be generated by an edge-replacement grammar [5]. Hence, if we want to have hyperedges of rank $k$, we need just add $k - 2$ additional nodes to every production (and the axiom) of this grammar and include these new nodes as the last items in all source lists. Thus we obtain a hyperedge-replacement grammar generating a language as claimed.

Courcelle [3] defines strongly context-free hypergraph languages. These are sets of hypergraphs the derivation trees of which can be described by monadic second-order formulas. Also, he studies a subset of this class of languages, the so-called regular hypergraph languages. For these, only productions of a particular form and $(0, k)$-star productions are allowed.

It seems that hyperedge-replacement languages generated by using only $k$-box productions – let us call them $k$-box languages – are strongly context-free, because of the uniqueness theorem. All regions (as defined in Section 3) of a hypergraph of such a language are leaves of its derivation tree. Splitting off these regions, we find that the regions of the resulting hypergraph are the “parents” of the leaves in the derivation tree, and so on. Courcelle obtains a quadratic parsing algorithm for the strongly context-free case, which is also what we get for $k$-box languages or $k$-box languages where $(0, k)$-star productions are allowed in addition, since we do not have to use the CYK-algorithm in this case.

On the other hand, $k$-box productions do also have certain similarities with the productions allowed in Courcelle’s regular hypergraph grammars. (For these he provides a linear parsing algorithm.) As mentioned above, regular hypergraph grammars may also use $(0, k)$-star productions. However, even strongly context-free grammars cannot use arbitrary $(i, k)$-star productions since the class of strongly context-
free hypergraph languages does not include all context-free string languages (see [3]). Finding out what the relation between these notions really are is an interesting task.

The notion of a split tree which we used to obtain the uniqueness result is closely related to that of a decomposition tree, as defined and investigated by other authors (see, e.g. [17, 13]). The difference is just that splitting introduces new hyperedges not present in a decomposition tree. Let $S$ be a split decomposition of a hypergraph $H$. If we define $\tilde{H}'=(V_{H'},E_{H'}\cap E_H)$ for all $H'\in S$, we obtain a decomposition tree $D$ of $H$ by setting $N_D=\{\tilde{H}'\mid H'\in S\}$ and $\text{neigh}_D(H')=\{\tilde{H}''\mid H''\in \text{neigh}_{T(S)}(H')\}$. However, the restriction under which (hyper)graphs and their decomposition trees are usually studied is bounded tree-width, meaning that one can always find a decomposition tree whose components are “small”, i.e., which have only a bounded number of nodes. On the other hand, our $k$-split trees may contain arbitrarily large components, but the intersection of two components’ node sets is always of size at most $k$.

By definition, similarity of split trees $T$ and $T'$ means that the corresponding decomposition trees are equal. So, the uniqueness theorem for collapsed $k$-split trees carries over to decomposition trees in the sense that collapsed decomposition trees (i.e., the decomposition trees corresponding to collapsed $k$-split trees) are unique for $k$ connected $k$-hypergraphs. However, since the components of decomposition trees do not have the new hyperedges, the components do not look that nice any more. So, it is quite unclear whether this observation has any interesting consequences for the study of decomposition trees.

A third method of decomposing graphs is given by the so-called simplicial decompositions (cf. [4]). These are mostly studied in connection with infinite-graph theory. Roughly speaking, the notion of simplicial decomposition allows a graph to be decomposed into two induced subgraphs if their union is just the original graph and their intersection is a complete graph. So, decomposing a graph is possible if (and only if) there is a clique in the graph the removal of whose nodes disconnects it. At first sight, this seems to be related to $k$-splitting because $k$-hyperedges may be seen as complete subgraphs, i.e., $k$-cliques (see the proof of Lemma 2.6). But there is an important difference. Whereas splitting introduces a new $k$-clique (i.e., a $k$-hyperedge), simplicial decomposition requires that it is already present in the graph to be decomposed. In other words, if we merge two hypergraphs, the result will not be simplicially decomposable in general. For example, all $(i,k)$-stars are prime with respect to simplicial decomposition since they do not contain a $k$-clique (i.e., a set $[\text{sources}(e)]$ for some hyperedge $e$) that disconnects the graph.

Acknowledgment

I thank Bruno Courcelle and Walter Vogler for numerous useful and interesting hints on the style and the contents of this paper.
References


