# Hyperedge Replacement with Rendezvous

Gnanamalar David[†]
Frank Drewes[††]
Hans-Jörg Kreowski[††]

Universität Bremen
FB Mathematik und Informatik
Postfach 33 04 40
D-2800 Bremen 33

**Abstract**
In this paper, we introduce and study a rendezvous mechanism for parallel replacements of hyperedges by (hyperedge-decorated) graphs that allows some merging of the replacing graphs if the attachment of the replaced hyperedges shares some nodes. The main result shows that the rendezvous mechanism can increase the generative power of table-controlled parallel hyperedge replacement graph grammars (which themselves are more powerful than ordinary hyperedge replacement graph grammars).

## 0 Introduction

Hyperedge replacement (as introduced by Feder [Fed71], Pavlidis [Pav72] and others under various synonyms) is one of the easiest and best studied types of graph rewriting (see, e.g., Bauderon and Courcelle [BC87], Lengauer and Wanke [LW88] and Habel, Kreowski and Vogler [Hab89], [HK87], [HKV89]). Hyperedge replacement graph grammars combine an attractive generative power with interesting structural and decidability properties. To some extend, these nice properties result from the fact that hyperedge replacement is context-free in the sense of Courcelle [Cou87] (see also Habel [Hab89]). On the other hand, the generative power is not only restricted as one must expect from a context-free mode of rewriting, but also because the nature of hyperedge replacement causes certain limitations. For each generated graph language, for example, there is a positive integer $k$ such that no member of the language is $k$-fold connected. Hence the set of all graphs, the set of all complete graphs, the set of all bipartite graphs and other graph languages of this kind cannot be generated by either sequential or parallel hyperedge replacement graph grammars.

In this paper, we introduce a rendezvous mechanism for parallel hyperedge replacement to overcome these limitations. If the attachments of some hyperedges share some nodes (called a "rendezvous"), their replacing graphs may be merged with each other

[†]The author thanks ÖSW, Bochum for its financial support which made his stay in Bremen possible.
[††]Partially supported by the ESPRIT Basic Research WG 3299 (COMPUGRAPH).

according to a predefined rendezvous specification. We show that the rendezvous mechanism can increase the generative power of table–controlled parallel hyperedge replacement graph grammars (which themselves are more powerful than ordinary hyperedge replacement grammars). Actually, we present a hyperedge replacement graph grammar with rendezvous generating the set of all complete graphs and one with a single table for the set of all graphs. The latter one uses only hyperedges of type 1, that is, hyperedges that point to only one node. Moreover, we show that for each table–controlled parallel hyperedge replacement graph grammar there exists a rendezvous table–controlled parallel hyperedge replacement graph grammar of order 2, that is, those using only hyperedges of type 2.

We use a parallel notion of replacement because this seems natural in order to explain the effect of the rendezvous specification. For other approaches to parallelism in the context of formal languages and graph grammars, see, e.g., Herman and Rozenberg [HR75], Ehrig, Kreowski and Taentzer [EK76], [ET92], Bailey, Cuny and Fisher [BCF91], and Nagl [Nag87].

# 1 Hyperedge Replacement

In this section, we recall the basic notions hyperedge replacement is built upon. We choose ordinary unlabelled undirected graphs without loops and multiple edges as basic structures of interest. To generate sets of graphs, these graphs become decorated with hyperedges each of which has a label and is attached to several nodes of the graph. Further, every such (hyperedge–)decorated graph has a sequence of so-called external nodes. These external nodes are needed to define the replacement of a hyperedge by a (decorated) graph. Every hyperedge of a decorated graph serves as a placeholder for a graph or — recursively — for another decorated graph, which can replace the hyperedge by fusing its external nodes with those the hyperedge is attached to.

### 1.1 Definition (graph)

A *graph* is a pair $(V, E)$ where $V$ is a finite set of *nodes* (or *vertices*) and $E$ is a set of 2–elements subsets of $V$, called *edges*. The set of all graphs is denoted by $\mathcal{G}_0$. □

### 1.2 Definition (decorated graph)

Let $N$ be a set of *nonterminal labels* each element $A$ of which is associated with a natural number, its *type*, denoted by $type(A)$.

A (hyperedge–)*decorated graph* (over $N$) is a system $H = (V, E, Y, lab, att, ext)$ where

- $(V, E)$ is in $\mathcal{G}_0$, called the *underlying graph* $U(H)$ of $H$,
- $Y$ is a finite set of *hyperedges*,
- $lab : Y \to N$ is a mapping, called the *labelling*,
- $att : Y \to V^*$ is a mapping (called the *attachment*) such that $|att(y)|^1 = type(lab(y))$ for all $y \in Y$,
- $ext \in V^*$, called the (sequence of) *external nodes*.

[1] Given $v \in V^*$, $|v|$ denotes the length of $v$.

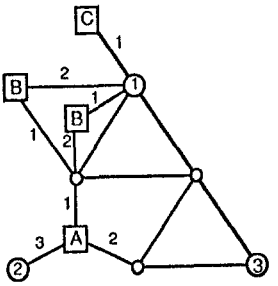
**Remarks.**

1. If $H$ is a decorated graph we denote its components by $V_H$, $E_H$, $Y_H$, $lab_H$, $att_H$ and $ext_H$ (unless they are explicitly named).
2. A set $N$ of nonterminals as above will be called *typed* in the following.
3. We let $type(H) = |ext|$ and for every $y \in Y_H$ we define $type_H(y) = type(lab(y))$.
4. The set of all decorated graphs of type $k$ over $N$ is denoted by $\mathcal{G}_k(N)$, and $\mathcal{G}(N)$ denotes $\bigcup_{k \in \mathbb{N}} \mathcal{G}_k(N)$.
5. For every label $A$ we set $[A] = \{1, \ldots, type(A)\}$. Accordingly, for a hyperedge $y \in Y_H$, let $[y]_H = [lab_H(y)]$.
6. For $A \in N$ the decorated graph $([A], \emptyset, \{y\}, lab, att, ext)$ with $lab(y) = A$ and $att(y) = ext = 1 \ldots type(A)$ is called a *handle* and denoted by $A^\bullet$. □

Two decorated graphs are said to be isomorphic if there is an isomorphism between the underlying graphs preserving external nodes, and there is a bijective mapping between their hyperedges which is consistent with the graph isomorphism.

**1.3 Definition (isomorphic decorated graphs)**
Two decorated graphs $H$ and $H'$ are *isomorphic*, denoted by $H \cong H'$, if there are bijective mappings $f : V_H \to V_{H'}$ and $g : Y_H \to Y_{H'}$ such that $E_{H'} = \{\{f(x), f(y)\} \mid \{x, y\} \in E_H\}$, $lab_H(y) = lab_{H'}(g(y))$ and $f^*(att_H(y)) = att_{H'}(g(y))$ for all $y \in Y_H$, and $f^*(ext_H) = ext_{H'}$, where $f^*$ is the natural extension of $f$ to sequences. □

**1.4 Example (hyperedge–decorated graph)**
The picture below shows a decorated graph consisting of a graph on six vertices having six edges together with four hyperedges. Its external nodes are drawn with numbers inside which indicate their order. The hyperedges are labelled with $A$, $B$, and $C$, and the order on their tentacles is again indicated by numbers. The type of this decorated graph as well as that of the $A$-labelled hyperedge is 3, while the hyperedges labelled with $B$ are of type 2 and the one labelled with $C$ is of type 1.




□

Next we define some frequently used operations on decorated graphs.

**1.5 Definition (operations on decorated graphs)**
Let $H, H' \in \mathcal{G}(N)$.

1. Let $B \subseteq Y_H$. Then the *removal* of $B$ from $H$ yields the decorated graph

$$H - B = (V_H, E_H, Y_H - B, lab, att, ext_H)$$

with $lab(y) = lab_H(y)$ and $att(y) = att_H(y)$ for all $y \in Y_H - B$.

2. The *disjoint union* of $H$ and $H'$ is the decorated graph

$$H + H' = (V_H \mathbin{\mathring{\cup}} V_{H'}, E_H \mathbin{\mathring{\cup}} E_{H'}, Y_H \mathbin{\mathring{\cup}} Y_{H'}, lab, att, ext_H)$$

with $lab(y) = lab_H(y)$ and $att(y) = att_H(y)$ for all $y \in Y_H$ and $lab(y) = lab_{H'}(y)$ and $att(y) = att_{H'}(y)$ for all $y \in Y_{H'}$. Here $\mathring{\cup}$ denotes the disjoint union of sets. (Observe that the disjoint union of decorated graphs is not commutative.)
For $H_1, \ldots, H_n \in \mathcal{G}(N)$, $\sum_{i=1}^{n} H_i$ means $H_1 + H_2 + \ldots + H_n$.

3. Let $f : V_H \to V$ be a mapping for some set $V$. Then the *nodes-set exchange* in $H$ through $f$ yields the decorated graph

$$H/f = (V, E, Y_H, lab_H, att, ext)$$

with $E = \{\{f(x), f(y)\} \mid \{x, y\} \in E_H, f(x) \neq f(y)\}$, $att(y) = f^*(att_H(y))$ for all $y \in Y_H$, and $ext = f^*(ext_H)$, where $f^*$ is the natural extension of $f$ to sequences.

4. Let $\delta$ be a binary relation on $V_H$, let $V$ be the quotient set of $V_H$ through the equivalence relation generated by $\delta$ with natural mapping $nat : V_H \to V$. Then $H/nat$ is called the *nodes fusion* according to $\delta$, and is also denoted by $H/\delta$. If $u, v \in V_H^*$, $u = u_1 \cdots u_n$ and $v = v_1 \cdots v_n$, and $\delta = \{(u_i, v_i) \mid 1 \leq i \leq n\}$, we also write $H/(u = v)$ instead of $H/\delta$. □

With the help of the operations just defined we now explain hyperedge replacement. We will restrict ourselves to the parallel mode of rewriting, i.e., we always replace all hyperedges occurring in a decorated graph simultaneously.

**1.6 Definition (parallel hyperedge replacement)**
Let $H \in \mathcal{G}(N)$ and let $repl : Y_H \to \mathcal{G}(N)$ be a mapping with $type_H(y) = type(repl(y))$ for $y \in Y_H$.

The *replacement* of $Y_H = \{y_1, \ldots, y_n\}$ in $H$ through $repl$ yields the decorated graph

$$\mathrm{REPLACE}(H, repl) = \left( (H - Y_H) + \sum_{i=1}^{n} repl(y_i) \right) \Big/ (att = ext)$$

with $att = att(y_1) \cdots att(y_n)$ and $ext = ext_{repl(y_1)} \cdots ext_{repl(y_n)}$.

**Remarks.**

1. Parallel hyperedge replacement is a simple construction where the hyperedges of a decorated graph are removed, the associated decorated graphs are added disjointly and their external nodes are fused with the corresponding nodes formerly attached to the replaced hyperedges.
2. Note that the component graphs replacing the hyperedges are fully embedded into the resulting graph where their external nodes loose this status.
3. If a mapping $repl : Y_H \to \mathcal{G}(N)$ meets the assumption of the definition, it will be referred to as *well-typed*. □

# 2 Parallel hyperedge replacement grammars

In this section hyperedge replacement with rendezvous is introduced. The basic model to which the notion of rendezvous is added is HTOL hyperedge replacement. The letter H refers to the hybrid type of graphs where hyperedges are rewritten while undirected edges remain unchanged. The letter T indicates that the considered grammars provide sets of tables where each table is a set of productions, and OL refers to the type of rewriting of OL systems (see, e.g., Rozenberg and Salomaa [RS80]) which was adapted to hyperedge replacement by Kreowski [Kre92].

We first define rendezvous functions. These are functions between binary relations.

**2.1 Definition (rendezvous function)**
Let $I$, $I'$, $J$, $J'$ be sets.

A *rendezvous function* is a function $r : \mathcal{P}(I \times I') \to \mathcal{P}(J \times J')$.

Here, $\mathcal{P}(S)$ denotes the powerset of a set $S$.

**Remark.**

The constant rendezvous function that always yields some fixed binary relation $B$ is denoted by $B$ itself.[2] In particular, $\emptyset$ denotes the rendezvous function assigning $\emptyset$ to every argument. □

**2.2 Definition (production, table set, rendezvous specification)**

1. A *production* (over $N$) is a pair $p = (A, R)$ with $A \in N$ and $R \in \mathcal{G}(N)$. $A$ is called the *left-hand side* of $p$ and denoted by $lhs(p)$. $R$ is called the *right-hand side* and denoted by $rhs(p)$. A finite set of productions is called a *table*, and a set of such tables is a *table set*.
2. Let $\mathcal{T}$ be a table set. A *rendezvous specification* for $\mathcal{T}$ is a function $\mathcal{R}$ which assigns to every table $P \in \mathcal{T}$ and every pair $(p, p') \in P \times P$ a rendezvous function $\mathcal{R}_P(p, p')$ from $\mathcal{P}([lhs(p)] \times [lhs(p')])$ to $\mathcal{P}(V_{rhs(p)} \times V_{rhs(p')})$.

**Remarks.**

1. For every decorated graph $H$ and all $y, y' \in Y_H$ we denote by $\rho_H(y, y')$ the binary relation
$$\rho_H(y, y') = \{(i, j) \in [y]_H \times [y']_H \mid att_H(y)_i = att_H(y')_j\}.$$
We call $\rho_H(y, y')$ the *rendezvous* of $y$ and $y'$ (in $H$). It is the binary relation describing which of the tentacles of $y$ and $y'$ are attached to the same node in $H$.
2. If $P \in \mathcal{T}$ and $\mathcal{R}_P(p, p') = r$ for some rendezvous function $r$ and all $p, p' \in P$ we may write $\mathcal{R}_P = r$. □

A rendezvous specification will be used as follows. Whenever the hyperedges of a decorated graph are replaced, for each pair of these hyperedges their rendezvous is determined. Afterwards, the hyperedges are replaced by the right-hand sides of the productions. In addition, nodes are fused if they are related to each other by

[2] Of course, this notation is ambigous since it does not distinguish between rendezvous functions having different domains and/or ranges. However, for our purposes it is not necessary to make a difference here.

$\mathcal{R}_P(p,p')(\rho)$, for every pair of productions applied, and where $\rho$ is the rendezvous of the replaced hyperedges.

We now define hyperedge replacement with rendezvous formally.

**2.3 Definition (hyperedge replacement with rendezvous)**
Let $H, H' \in \mathcal{G}(N)$, let $\mathcal{T}$ be a table set and let $\mathcal{R}$ be some rendezvous specification for $\mathcal{T}$.

1. A mapping $b : Y_H \to P$ with $P \in \mathcal{T}$ is called an *L-base* (over $\mathcal{T}$) in $H$ if the mapping *repl* defined by $repl(y) = rhs(b(y))$ for all $y \in Y_H$ is well-typed.
2. For every L-base $b : Y_H \to P$ over $\mathcal{T}$ we define
$$\mathcal{R}(H,b) = \bigcup_{y,y' \in Y_H} \mathcal{R}_P(b(y), b(y'))(\rho_H(y,y')).$$
3. Let $b : Y_H \to P$ an L-base over $\mathcal{T}$ in $H$. Then there is a *direct derivation* $H \underset{P,\mathcal{R}}{\Longrightarrow} H'$ *through* $b$ if
$$H' \cong \mathrm{REPLACE}(H, repl)/\mathcal{R}(H,b),$$
with $repl : Y_H \to \mathcal{G}(N)$ defined by $repl(y) = rhs(b(y))$ for all $y \in Y_H$.
4. A sequence of direct derivations of the form $H_0 \underset{P_1,\mathcal{R}}{\Longrightarrow} H_1 \underset{P_2,\mathcal{R}}{\Longrightarrow} \cdots \underset{P_k,\mathcal{R}}{\Longrightarrow} H_k$ with $P_1, \ldots, P_k \in \mathcal{T}$ is called a *derivation* (of *length* $k$) from $H_0$ to $H_k$ and is denoted by $H_0 \overset{k}{\underset{\mathcal{T},\mathcal{R}}{\Longrightarrow}} H_k$. If $H \cong H'$, this may be denoted by $H \overset{0}{\underset{\mathcal{T},\mathcal{R}}{\Longrightarrow}} H'$. If $k$ in $H \overset{k}{\underset{\mathcal{T},\mathcal{R}}{\Longrightarrow}} H'$ does not matter, it may be replaced by a star symbol. □

**2.4 Definition (hyperedge-replacement grammars with rendezvous)**
1. A *hybrid table-controlled OL hyperedge replacement grammar with rendezvous* (an *RHTOL HR grammar*) is a system $G = (N, \mathcal{T}, \mathcal{R}, Z)$ where $N$ is a typed set of nonterminals, $\mathcal{T}$ is a finite table set, $\mathcal{R}$ is a rendezvous specification for $\mathcal{T}$, and $Z \in \mathcal{G}_0(N)$, called the *axiom*. If $Z = S^\bullet$ for some $S \in N$ we may denote $G$ by $(N, \mathcal{T}, \mathcal{R}, S)$.
2. Given such a grammar, the *generated graph language* consists of all graphs derivable from $Z$, i.e.,
$$L(G) = \{H \in \mathcal{G}_0 \mid Z \overset{*}{\underset{\mathcal{T},\mathcal{R}}{\Longrightarrow}} H\}.$$

**Remarks.**
1. $G$ is said to be *of order* $k$ if $k$ is the least natural number satisfying $type(A) \leq k$ for all $A \in N$.
2. If $\mathcal{C}$ is a class of RHTOL HR grammars we denote by $\mathcal{L}(\mathcal{C})$ the set $\{L(G) \mid G \in \mathcal{C}\}$ and by $\mathcal{C}_i$ the set of all grammars in $\mathcal{C}$ of order not greater than $i$.
3. $\mathcal{HTOL}$ denotes the class of RHTOL HR grammars with trivial rendezvous specification. That is, $\mathcal{R}_P = \emptyset$ for all tables. $\mathcal{RHOL}$ denotes the class of RHTOL HR grammars with a single table and $\mathcal{HOL}$ denotes the intersection of $\mathcal{HTOL}$ and $\mathcal{RHOL}$. □

The classes $\mathcal{HTOL}$ and $\mathcal{HOL}$ compare to the respective classes introduced by Kreowski [Kre92]. Kreowski proves in particular that $\mathcal{L}(\mathcal{HOL})$ is properly included in $\mathcal{L}(\mathcal{HTOL})$. HOL HR grammars in turn have the same language generative power as ordinary sequential hyperedge replacement grammars.[3]

The construction used for the rendezvous mechanism reminds of earlier work on amalgamation of graph productions by Boehm, Fonio and Habel [BFH87] (see also Degano and Montanari [DM87]). However, whereas amalgamation is used to build new productions out of given ones (and new derivations out of given ones), the rendezvous mechanism influences the result of a (parallel) rewriting step. Whereas the main theorem obtained by Boehm, Fonio and Habel is that amalgamation does not change the generated graphs the idea behind rendezvous is to increase the generative power of hyperedge replacement. So, rendezvous and amalgamation are actually not as closely related as it could seem at first sight.

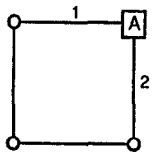
# 3 Examples

In this section we present some examples of RHTOL HR grammars. The first grammar generates all square grids, the second one all complete graphs and the third one all graphs.

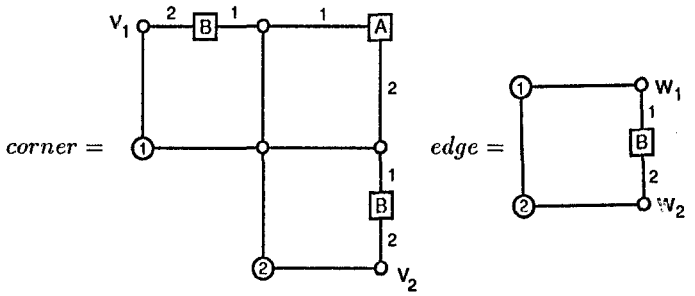
### 3.1 Example (generation of square grids)

The RHTOL HR grammar $GRID = (\{A, B\}, \{P_1, P_2\}, \mathcal{R}, Z)$, which generates the set of all square grids, is defined as follows.

$Z =$




$P_1 = \{p_c, p_e\}$ with $p_c = (A, corner)$, $p_e = (B, edge)$ and



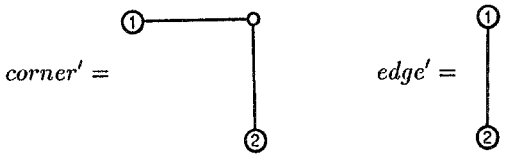

$$\mathcal{R}_{P_1}(p_c, p_e)(\rho) = \begin{cases} \{(v_i, w_1)\} & \text{if } \rho = \{(i,1)\} \\ \emptyset & \text{otherwise,} \end{cases}$$

[3]To be precise, one ought to say that $\mathcal{HOL}$ compares to the set of all hyperedge-replacement grammars where only *canonical derivations* in the sense of Kreowski [Kre87] are allowed. From the viewpoint of generated languages, however, this makes no difference.

$$\mathcal{R}_{P_1}(p_c, p_e)(\rho) = \begin{cases} \{(w_1, w_2)\} & \text{if } \rho = \{(1,2)\} \\ \emptyset & \text{otherwise} \end{cases}$$

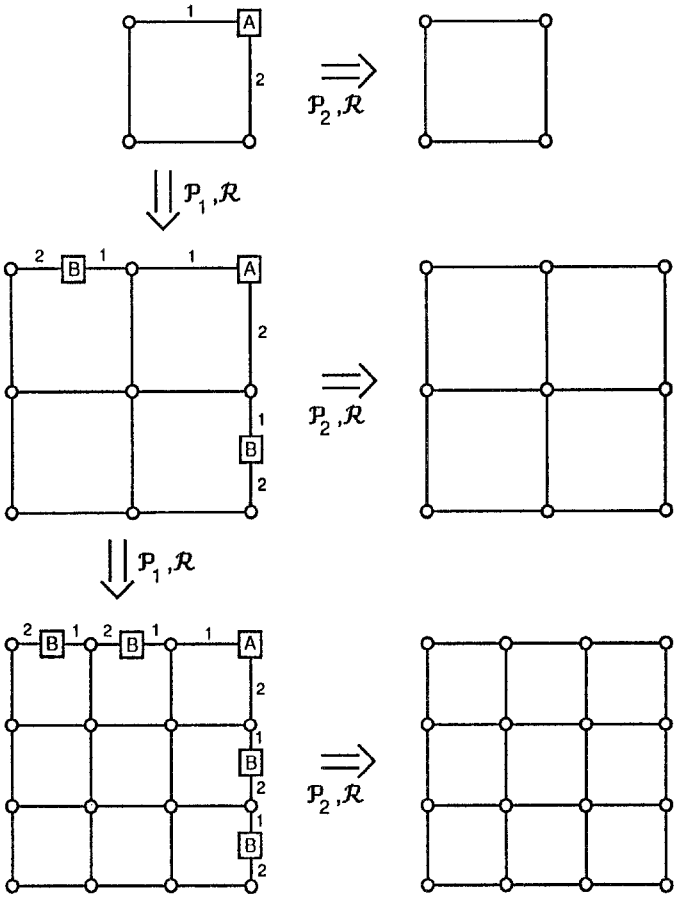
and $\mathcal{R}_{P_1} = \emptyset$ otherwise.

$P_2 = \{p'_c, p'_e\}$ with $p'_c = (A, corner')$, $p'_e = (B, edge')$ and

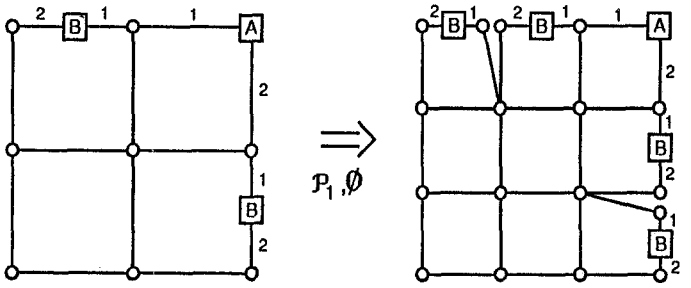



and $\mathcal{R}_{P_2} = \emptyset$.

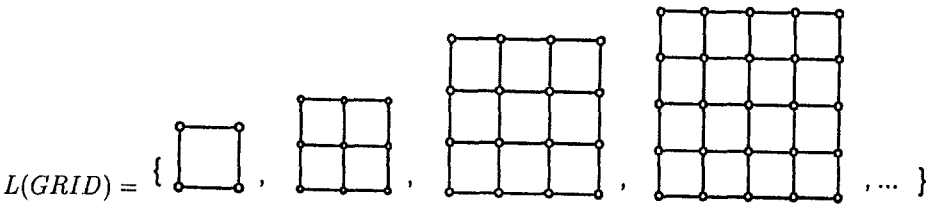
The first three derivations in *GRID* are illustrated below.



If we take the empty rendezvous specification $\emptyset$ instead of $\mathcal{R}$ (i.e., if we consider the derivations without rendezvous) we get instead derivations of the following kind.



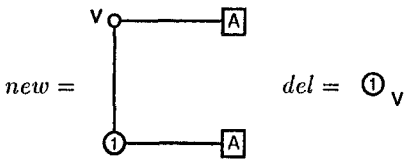

The language generated by *GRID* is the set of all square grids:




□

**3.2 Example (generation of all complete graphs)**
In order to generate the set of all complete graphs the RHTOL HR grammar $K$ given by $K = (\{A\}, \{P_1, P_2\}, \mathcal{R}, A)$ can be used. Its components are defined as follows.

$P_1 = \{p_{new}\}$, $P_2 = \{p_{del}\}$ where $p_{new} = (A, new)$ and $p_{del} = (A, del)$ with




and $\mathcal{R}_{P_1} = \{(v, v)\}$, $\mathcal{R}_{P_2} = \emptyset$.

This grammar generates the set of all complete graphs in the following way. In every intermediate graph, all nodes carry at least one A-labelled hyperedge. Each such hyperedge generates a new node v which is connected with the old one by an edge. By the rendezvous specification, all the new nodes are merged into one, which is thus connected with every old one. □

**3.3 Example (generation of all graphs)**
Using basically the same idea as above we obtain $ALL = (\{A\}, \{P_1, P_2\}, \mathcal{R}, A)$ which generates the set of all graphs. We let $P_2$ be as in $K$, $P_1 = \{p_{new}, p'_{new}\}$, with $p_{new}$ as above and $p'_{new} = (A, new')$ where


$new' =$


and, again, $\mathcal{R}_{P_1} = \{(v, v)\}$, $\mathcal{R}_{P_2} = \emptyset$. □

# 4 Internal Rendezvous

As mentioned in the introduction one of the main limitations of hyperedge replacement stems from the fact that, if two hyperedges are replaced by some graphs, the newly introduced parts cannot be connected directly. The examples in Section 3 indicate that the rendezvous mechanism enables us to overcome this disadvantage. On the other hand, the examples do not fully exploit the power of the new notion because we never merge "old" parts, that is, external nodes of the right-hand sides. In fact, rendezvous in its most general form seems to exceed our initial intention: Not only are we able to introduce dependencies between new nodes created from different hyperedges in a parallel step; it is also possible to let the graph shrink by fusing old nodes. This is why it is quite unclear how the membership problem can be solved for arbitrary RHTOL HR languages. Because of these reasons the class of RHTOL HR grammars in which only internal nodes (that is, nodes that are not external ones) are merged by the rendezvous specification is of special interest.

**4.1 Definition (hyperedge replacement with internal rendezvous)**
The class $\mathcal{IHTOL}$ is the set of all RHTOL HR grammars with internal rendezvous (IHTOL HR grammars). An RHTOL HR grammar $(N, \mathcal{T}, \mathcal{R}, Z)$ is in $\mathcal{IHTOL}$ if we have for all $P \in \mathcal{T}$, $p_1, p_2 \in P$, $\rho \subseteq [lhs(p_1)] \times [lhs(p_2)]$ and $(v_1, v_2) \in \mathcal{R}_P(p_1, p_2)(\rho)$ that $v_i$ is an internal node of $p_i$ (for $i = 1, 2$).

We have the following theorem.

**4.2 Theorem**
Let $G \in \mathcal{IHTOL}$ be such that every right-hand side of a production in $G$ contains at least one internal node. Then the membership problem for G is decidable.

**Proof.** In each parallel step in G the size (= number of nodes) of the generated graph increases because only internal nodes are affected by the rendezvous and every right-hand side contains (at least) one internal node. (The "worst case" is that all the internal nodes get merged, yielding one new node.) But this means that an easy modification of the well-known algorithm to recognize context-sensitive string languages can be used to recognize $L(G)$. This algorithm produces all derivations that yield graphs of the input graph's size and decides whether one of the generated graphs is isomorphic to the input graph. □

The reader should notice that all our examples are of the form required by Theorem 4.2. It is an interesting question whether the theorem can be extended to all IHTOL HR grammars, without the requirement of producing at least one new node in each step.

# 5 The generating power of RHTOL HR grammars

In this section we investigate the power of RHTOL HR grammars. In particular, we prove two theorems. The first one states that there are languages which can be generated by grammars in $\mathcal{IHOL}_1$, i.e., very restricted IHTOL HR grammars that generate languages which cannot be generated by HTOL HR grammars. As a second result of this section we prove that every language generated by an HTOL HR grammar can also be generated by an RHTOL HR grammar of order 2.

The theorem below follows from the fact that — as shown in Example 3.3 — there is a grammar $G \in \mathcal{IHTOL}_1$ generating $\mathcal{G}_0$.

**5.1 Theorem**

$$\mathcal{L}(\mathcal{IHOL}_1) \setminus \mathcal{L}(\mathcal{HTOL}) \neq \emptyset$$

**Proof.** For every $G = (N, \mathcal{T}, \mathcal{R}, Z) \in \mathcal{IHTOL}$ let $G_\cup = (N, \{P\}, \mathcal{R}', Z)$, where $P = \bigcup \mathcal{T}$ and

$$\mathcal{R}'_P(p,p') = \begin{cases} \mathcal{R}_{P'}(p,p') & \text{if } p,p' \in P' \text{ for some } P' \in \mathcal{T} \\ \emptyset & \text{if there is no such } P' \in \mathcal{T}. \end{cases}$$

Obviously, $G_\cup$ is in $\mathcal{IHOL}$ and we have $L(G) \subseteq L(G_\cup)$ since every derivation in $G$ is also admissible in $G_\cup$. Applying this to the IHTOL HR grammar $ALL$ from Example 3.3 we get $L(ALL) \subseteq L(ALL_\cup) \in \mathcal{IHOL}_1$. Since $L(ALL) = \mathcal{G}_0$ this means $L(ALL) = L(ALL_\cup) = \mathcal{G}_0$. It remains to show that there is no $G' \in \mathcal{HTOL}$ generating $\mathcal{G}_0$. By the above for all $G' \in \mathcal{HTOL}$ we have $L(G') \subseteq L(G'_\cup) \in \mathcal{HOL}$. However, it is well-known that there is no hyperedge-replacement grammar generating $\mathcal{G}_0$ [Hab89], that is, no grammar in $\mathcal{HOL}$ generates $\mathcal{G}_0$. □

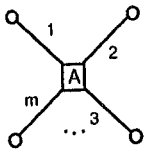
Our next theorem states that for every HTOL HR grammar there is an equivalent RHTOL HR grammar of order 2.

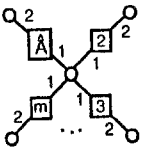
**5.2 Theorem**

$$\mathcal{L}(\mathcal{HTOL}) \subseteq \mathcal{L}(\mathcal{RHTOL}_2)$$

**Proof.** Let $G = (N, \mathcal{T}, \mathcal{R}, Z)$ be of order $k$. In the following we will define the components of the RHTOL HR grammar $G' = (N', \mathcal{T}', \mathcal{R}', Z')$ satisfying $L(G') = L(G)$. Let $N' = \{\hat{A} \mid A \in N\} \mathring{\cup} \{1, \ldots, k\}$, where all labels are of type 2.

The basic idea of the proof is as follows. Consider any $A$-labelled hyperedge $y$ of type $m$ occurring in $Z$ or in a production, such as the one depicted in the left-hand side below.




This hyperedge will be replaced by the "bipartite" decorated graph in the right-hand side above. Here, the middle node is an auxiliary, new node. The hyperedge labelled $\hat{A}$ (which is attached to the first attached node of $y$) now gets replaced as $y$ got before, but with the slight modification that only the first external node of the replacing right-hand side remains an external node. All others become ordinary nodes. Hence additional identifications are necessary: the $i$th attached node of $y$ has to be fused with the node which was the $i$th external one of the right-hand side before. This is easily done by a rendezvous relation for a $(1,1)$-rendezvous between the $\hat{A}$-labelled and the $i$-labelled hyperedge.

For each decorated graph $H = (V, E, Y, lab, att, ext)$ let $f(H) = (V', E, Y', lab', att', ext)$, where

$$V' = V \mathbin{\mathring{\cup}} \{v_y \mid y \in Y\}$$

$$Y' = \bigcup_{y \in Y} \{y_i \mid i \in [y]_H\}$$

$$att'(y_i) = v_y att(y)_i \text{ and}$$

$$lab'(y_i) = \begin{cases} \widehat{lab(y)} & \text{if } i = 1 \\ i & \text{if } 2 \leq i \leq type_H(y). \end{cases}$$

Now define for each production $p = (A, R) \in P \in \mathcal{T}$ a production $\hat{p} = (\hat{A}, \hat{R})$ as follows. If $f(R) = (V, E, Y, lab, att, ext)$ then

$$\hat{R} = (V \mathbin{\mathring{\cup}} \{mid\}, E, Y, lab, att, mid\, ext_1)$$

Furthermore, let $q_i = (i, i^\bullet - Y_{i\bullet})$, for $i = 2, \ldots, k$. For every $P \in \mathcal{T}$ we set

$$\hat{P} = \{\hat{p} \mid p \in P\} \cup \{q_i \mid i = 2, \ldots, k\}.$$

Now $\mathcal{T}' = \{\hat{P} \mid P \in \mathcal{T}\}$. For every $\hat{p} \in \hat{P} \in \mathcal{T}'$ and for all $q_i$, $2 \leq i \leq m$, where $ext_{rhs(p)} = v_1 \ldots v_m$ and $ext_{rhs(q_i)} = v\, v'$, we define

$$\mathcal{R}'_P(\hat{p}, q_i)(eq) = \begin{cases} \{(v_i, v'), (v_1, v)\} & \text{if } eq = \{(1,1)\} \\ \emptyset & \text{otherwise.} \end{cases}$$

For all other pairs $p, p' \in P$ we let $\mathcal{R}'_P(p, p') = \emptyset$. Finally, let $Z' = f(Z)$.

It remains to prove that $L(G') = L(G)$. For this, it suffices to show that for all decorated graphs $H, H' \in \mathcal{G}(N)$ there is a direct derivation $H \underset{\mathcal{T},\mathcal{R}}{\Longrightarrow} H'$ if and only if $f(H) \underset{\mathcal{T}',\mathcal{R}'}{\Longrightarrow} f(H')$, since by definition of $f$ we have $f(H) = H$ if $Y_H = \emptyset$.

Since $\mathcal{R}'$ is empty except for the rendezvous $\{(1,1)\}$ two replacements of hyperedges $y_1, y_2 \in Y_{f(H)}$ are not affected by the rendezvous unless they stem from the same hyperedge in $H$. Hence by context-freeness of hyperedge replacement we may without loss of generality assume that $H$ is a handle $A^\bullet$. Let $f(R) = (V, E, Y, lab, att, ext)$ and let $p = (A, R) \in P$ be the production applied to $A^\bullet$ in order to derive $H'$. Then $H' \cong R$, hence $f(H') \cong f(R)$. Let $b : Y_{f(H)} \to \hat{P}$ be defined by

$$repl(y) = \begin{cases} \hat{p} & \text{if } lab_{f(H)}(y) = \hat{A} \\ q_i & \text{if } lab_{f(H)}(y) = i. \end{cases}$$

With $m = type(A)$, $ext_H = v_1 \dots v_m$, and $Y_H = \{y\}$ we have

$$f(H) \underset{T,\emptyset}{\Longrightarrow} (V \mathring{\cup} \{v_y, v_2, \dots, v_m\}, E, Y, lab, att, v_1 \dots v_m)$$

through $b$. Since the rendezvous of each two hyeredges of $f(H)$ is $\{(1,1)\}$, $\mathcal{R}'(f(H), b)$ identifies $v_y$ with $v_1$ and every $v_i$ with $ext_i$, for $i = 2, \dots, m$. (Observe that $v_1 = ext_1$, anyway.) Thus

$$f(H) \underset{T,\mathcal{R}'}{\Longrightarrow} (V, E, Y, lab, att, ext_1 \dots ext_m) = f(R) \cong f(H')$$

as asserted.

For the other direction let $f(H) = f(A^\bullet) \underset{T',\mathcal{R}'}{\Longrightarrow} H''$. Then $Y_{f(H)} = \{y_1, \dots, y_m\}$, where $y_1$ is labelled by $\hat{A}$ and for each $i$, $2 \le i \le m$, $y_i$ is labelled by $i$. So the L-base $b'$ used in this direct derivation must have the form

$$b'(y_i) = \begin{cases} \hat{p} & \text{if } i = 1 \\ q_i & \text{if } i > 1 \end{cases}$$

for some $\hat{p}$ and $\hat{P}$ with $\hat{p} \in \hat{P} \in T'$. Hence $b$ defined by $b(y) = p$ is an L-base over $T$ in $H = A^\bullet$ since $p \in P \in T$. Now $H \underset{T,\mathcal{R}}{\Longrightarrow} H'$ through $b$, for some $H' \in \mathcal{G}(N)$. By the first part of the proof this means $f(H) \underset{T',\mathcal{R}'}{\Longrightarrow} f(H')$ through $b'$, so $H'' \cong f(H')$. □

Observe that the proof of Theorem 5.2 relies heavily on the fact that external nodes are merged by the rendezvous specification. There seems to be no way to achive this effect using an IHTOL$_2$ HR grammar. We have the following corollary summerizing the (proper) inclusions known so far.

**5.3 Corollary**

1. $\mathcal{L}(\mathcal{HOL}) \subset \mathcal{L}(\mathcal{HTOL}) \subset \begin{matrix} \mathcal{L}(\mathcal{IHTOL}) \\ \mathcal{L}(\mathcal{RHTOL}_2) \end{matrix} \subseteq \mathcal{L}(\mathcal{RHTOL})$

2. $\mathcal{L}(\mathcal{HOL}) \subset \mathcal{L}(\mathcal{IHOL}) \subseteq \mathcal{L}(\mathcal{RHOL}) \subseteq \mathcal{L}(\mathcal{RHTOL})$.

**Proof.** The properness of $\mathcal{L}(\mathcal{HOL}) \subseteq \mathcal{L}(\mathcal{HTOL})$ was shown by Kreowski [Kre92]. Theorem 5.1 yields both $\mathcal{L}(\mathcal{HTOL}) \subset \mathcal{L}(\mathcal{IHTOL})$ and $\mathcal{L}(\mathcal{HOL}) \subset \mathcal{L}(\mathcal{IHOL})$ and $\mathcal{L}(\mathcal{HTOL}) \subset \mathcal{L}(\mathcal{RHTOL}_2)$ is the statement of Theorem 5.2. □

# 6 Conclusion

This paper presents the very first steps of the investigation of hyperedge replacement with rendezvous. The rendezvous mechanism allows to overcome certain limitations of the generative power of hyperedge replacement grammars without rendezvous. The following are open problems for further consideration:

1. The summarizing Corollary 5.3 leaves the following questions: Are $\mathcal{L}(\mathcal{RHOL})$ and $\mathcal{L}(\mathcal{IHTOL})$ properly included in $\mathcal{L}(\mathcal{RHTOL})$ or not? Is $\mathcal{L}(\mathcal{HTOL})$ included in $\mathcal{L}(\mathcal{RHOL})$ or even in $\mathcal{L}(\mathcal{IHOL})$?
2. The generative power of ordinary hyperedge-replacement graph grammars increases with the order. Is this also the case if the grammars employ the rendezvous mechanism? The authors conjecture that this should be true for $\mathcal{IHTOL}$ but false for $\mathcal{RHTOL}$. (For the latter a proof similar to the one for Theorem 5.2 may be possible but the extension is not trivial.)
3. Is there a sort of "context-freeness lemma" for $\mathcal{IHTOL}$? Which are other interesting special cases of the rendezvous mechanism that increase the generative power of hyperedge replacement but do not destroy all the theory?
4. More explicitly speaking, which of the rich structural and decidability results of hyperedge-replacement grammars and languages can be carried over or adapted to the case of grammars with rendezvous or subclasses of it?

# References


[BC87] M. Bauderon, B. Courcelle. *Graph Expressions and Graph Rewriting. Mathematical Systems Theory*, 20, 83–127, 1987.

[BCF91] D. Bailey, J. Cuny, C. Fischer. *Programming with Very Large Graphs. Lecture Notes in Computer Science*, volume 532, 84–97, 1991.

[BFH87] P. Boehm, H.-R. Fonio, A. Habel. *Amalgamation of Graph Transformations: A Synchronization Mechanism. Journal of Computer and System Sciences*, 34, 377–408, 1987.

[Cou87] B. Courcelle. *An Axiomatic Definition of Context-Free Rewriting and its Application to NLC Graph Grammars. Theoretical Computer Science*, 55, 141–1812, 1987.

[DM87] P. Degano, U. Montanari. *A Model of Distributed Systems based on Graph Rewriting. Journ. ACM*, 34, 411–449, 1987.

[EK76] H. Ehrig, H.-J. Kreowski. *Parallel Graph Grammars.* In A. Lindenmayer, G. Rozenberg, editors, *Automata, Languages, Development*, 425–442. North Holland, Amsterdam, 1976.

[ET92] H. Ehrig, G. Taentzer. *From Parallel Graph Grammars to Parallel High-Level Replacement Systems.* In G. Rozenberg, A. Salomaa, editors, *Lindenmayer Systems*, 283–304. Springer, 1992.

[Fed71] J. Feder. *Plex Languages. Inform. Sci.*, 3, 225–241, 1971.

[Hab89] A. Habel. *Hyperedge Replacement: Grammars and Languages.* PhD thesis, Bremen, 1989.

[HK87] A. Habel, H.-J. Kreowski. *Some Structural Aspects of Hypergraph Languages Generated by Hyperedge Replacement. Lecture Notes in Computer Science*, volume 247, 207–219, 1987.

[HKV89] A. Habel, H.-J. Kreowski, W. Vogler. *Metatheorems for Decision Problems on Hyperedge Replacement Graph Languages. Acta Informatica*, 26, 657–677, 1989.

[HR75] G.T. Herman, G Rozenberg. *Developmental Systems and Languages.* North Holland/American Elsevier, New York, 1975.

[Kre87] H.-J. Kreowski. *Is Parallelism Already Concurrency? Part 1: Derivations in Graph Grammars. Lecture Notes in Computer Science*, volume 291, 343–360, 1987.

[Kre92] H.-J. Kreowski. *Parallel Hyperedge Replacement.* In G. Rozenberg, A. Salomaa, editors, *Lindenmayer Systems*, 271–282. Springer, 1992.

[LW88] T. Lengauer, E. Wanke. *Efficient Analysis of Graph Properties on Context-Free Graph Languages. Lecture Notes in Computer Science*, volume 317, 379–393, 1988.

[Nag87] M. Nagl. *Set Theoretic Approaches to Graph Grammars. Lecture Notes in Computer Science*, volume 291, 41–54, 1987.

[Pav72] T. Pavlidis. *Linear and Context-Free Graph Grammars. Journ. ACM*, 19(1), 11–23, 1972.

[RS80] G. Rozenberg, A. Salomaa. *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.