Topological Graphs and Combinators - Extended Version

Bruno de Assis Delboni[†1] Mauricio Ayala-Rincón[†,‡ 2] Daniele Nantes-Sobrinho[†3]

Departamentos de † Matemática e ‡ Ciência da Computação
Universidade de Brasília, Brasília, Brazil

Abstract

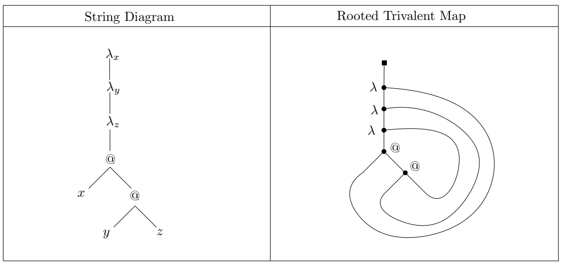
A class of topological graphs is provided that represents combinators and subclasses of lambda-terms. The graph representation takes into account points in the real Cartesian plane $\mathbb{R} \times \mathbb{R}$ which become vertices if they have a specific annotation and edges that are curves incident to two points that might or not be vertices. The graph structure is enriched with an ordering on the edges incident to each vertex built through the use of a notion of orientation. A calculus is provided to generate subclasses of topological graphs that corresponds to `BCI` and `BCK` combinators. As an application, a principal type algorithm on these graphs is provided.

Keywords: Topological graphs, combinatory logic, lambda terms, type inference.

1 Introduction

The objective of this paper is to study the relation between combinators, λ-terms and a new class of graphs, called *topological graphs*, which is proposed here. The approach follows/extends the technique proposed by Zeilberger in [Zei16], where several connections between linear λ-terms and rooted connected trivalent maps were coined out. It was shown that linear λ-term have a tree representation of its structure that can be related to a unique connected rooted trivalent map. This result can be extended to `BCK` λ-terms and rooted maps whose vertices have degree two or three.

The use of diagrams for the representation of λ-terms is well-known and very intuitive. For instance, the linear λ-term $B \equiv \lambda xyz.x(yz)$ has the string diagram (a.k.a, syntactic tree) on the left of the figure, whereas the drawing on the right of the figure is the trivalent rooted map representation of B, following the approach in [Zei16].

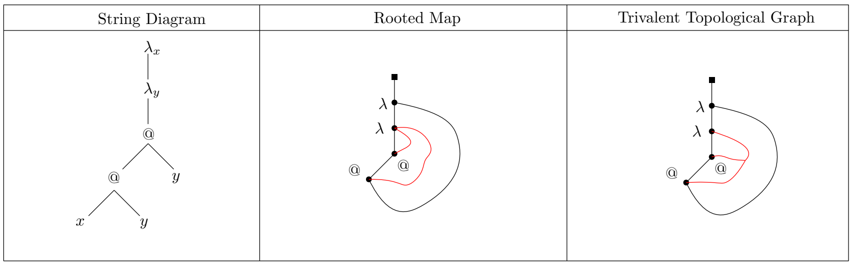


We have observed that for *non-linear, closed λ-terms without variable clashes*, there is also a natural association with connected rooted *non-trivalent maps*. This relation arises from the fact that such a term can be obtained from a `BCK` λ-term in which each free variable is identified with a bounded variable. From this observation, it arises the question: *how to decide whether a non necessarily trivalent connected rooted map represents or not a λ-term?*

To answer this question, a new structure is introduced, called *topological graph*, which is capable of representing non-linear λ-terms. This structure includes, besides (free-)edges and vertices, a new kind of object called `whip`, which is used to represent several occurrences of the same variable. For instance, the figure below

[1] Email: `bruno.delboni@gmail.com` Author supported by a Ph.D. Scholarship from the Brazilian Federal Counsel CNPq.
[2] Email: `ayala@unb.br` Author partially supported by a CNPq high productivity research grant 307672/2017-4.
[3] Email: `dnantes@unb.br` Research supported by the Federal District Foundation FAPDF within grant DE 0193001369/2016.

illustrates the string diagram (left) of $W \equiv \lambda xy.xyy$, the *non-trivalent* rooted map associated to W (center), and the trivalent topological graph (right) that is constructed and explored in this paper. The red "edges" representing the two occurrences of y (center) collapsed in one so called 2-`whip` (red object on the right).



Contributions. The main contributions of this paper are listed below.

- A new general definition of graphs, the topological graphs, which include `whips`, and can represent several occurrences of the same variable (free or bound).
- A constructive notion of graph decomposition which identifies graphs corresponding to λ-terms. This decomposition starts from any free edge of a connected graph, removing it and its incident vertices giving as result either one or two connected subgraphs according to the vertex being associated to an abstraction or application.
- Classification of fully decomposable subclasses of graphs that correspond to so called `BCK`-, `BCI`- and ω-graphs associated respectively to the combinatorial classes `BCK`, `BCI` and `SK`. A calculus is given to generate these kind of graphs which is proved sound and complete modulo isomorphism.
- Finally, a principal type algorithm over ω-graphs is given.

Related Work. As in computation, graphs are ubiquitous in the theory of combinators and in λ-calculus. Syntactically λ-terms have a natural representation as trees in which their occurrences, positions and subterms are organised. Also, statical analyses of λ-terms are naturally performed over this structure; for instance, by subject-construction Theorem the simple type deduction of a λ-term has the same tree structure than the tree structure of the term (e.g., see [Hin97]). Also, regarding dynamic analysis, well-known Böhm's trees are used to provide denotational semantics to computations through λ-calculus (e.g., Chapter 10 in [Bar12]).

The correspondence between closed linear λ-terms and classes of rooted trivalent graphs or maps on compact-oriented surfaces without boundary was introduced by Bodini, Gardy and Jacquot in [BGJ13]. They related unlabelled and labelled rooted combinatorial maps with BCI and BCK λ-systems and explored their combinatorial properties based on these structures. In [ZG15] Zeilberger and Giorgetti proved a correspondence between rooted planar maps and normal planar lambda terms. Further, a more general and conceptual account of this correspondence was formalised by Zeilberger in [Zei16] in which λ-terms with free variables were considered too. The last paper references correspondences among α-equivalence classes of λ-terms and isomorphism classes of these maps relating λ-terms with rooted maps, linear λ-terms with rooted trivalent maps, normal planar λ-terms with rooted planar maps, and normal linear λ-terms with rooted maps. Zeilberger extended Bodini *et al.* bijection to a more general one that relates linear λ-terms with free variables and rooted trivalent maps with a marked boundary of free edges and provides decomposition and forgetful operators that are applied to extract λ-terms from trivalent graphs and to build these graphs from the natural diagrammatic representation of λ-terms. An interesting application of Zeilberger's approach is a representation of the four colors theorem. More recently, Zeilberger's work evolves to the development of connections between type inference and flow problem in maps [Zei18]. When the λ-terms are closed, their representations can be seen as connected rooted *hyper graphs* [LZ04] with vertices of degree either two or three.

For the objectives of this paper, in which the planarity of the representation over general topological surfaces is not a requirement, the proposed topological graphs are build just over the surface $\mathbb{R}^2$ and, as previously mentioned, `whips` are used to deal with non linearity. Indeed, $\mathbb{R}^2$ is used only for simplicity since any two dimensional topological orientable space would be adequate.

Organisation. Sec. 2 gives some basic notions about combinators and introduces topological graphs while Sec. 3 presents the combinatorial graphs, as well as, the subclasses of `BCK` `BCI`- and ω-graphs. Sec. 4 defines operations on ω- graphs, introduces a calculus to generate them and their relation to λ-terms. Sec. 5 introduces a sound and complete principal typing algorithm for graphs, and Sec. 6 concludes. Additional details and complete proofs are included in the appendix.

2 Background

2.1 Combinators

Combinatory logics[HS86], invented by Schönfinkel in 1924 and independently by Curry in 1930, uses combinators to get rid of the complications concerned with operations related to the manipulation of bound variables such as substitution and α-equivalence. Combinators are *constants* that describe, by concatenation, operations without using free variables. Curry's notation for combinators is used and the class of *CL-terms* defined as concatenations of variables and the constants $\mathtt{S}$ and $\mathtt{K}$, called *principal combinators*, as following.

Definition 2.1 [CL-terms] Let $\mathcal{V}$ be a countably set of *variables* disjoint of $\{), (, \mathtt{S}, \mathtt{K}\}$. CL-terms are inductively defined as: • $x \in \mathcal{V}$ is a CL-*term*, • $\mathtt{S}, \mathtt{K}$ are CL-*terms*, • (MN) is a CL-*term*, if M and N are CL-*terms*.

Notation 1 *As usual, application of CL-terms is associated to the left allowing elimination of unnecessary parentheses; for instance, PQR abbreviates $((PQ)R)$, but $(P(QR))$ can be abbreviated just as $P(QR)$.*

The set of all variables occurring in a CL-term Q is denoted as $\mathtt{FV}(Q)$ and, if $\mathtt{FV}(Q) = \emptyset$, then Q is called *closed*. Let P, Q and R be CL-terms, x and y distinct variables. Define the operation of *substitution*, as the act of *replacing a term R for a variable x in P*, denoted as $P[R/x]$, by induction over the CL-terms as
• $x[R/x] := R$, • $y[R/x] := y$, • $(PQ)[R/x] := (P[R/x]Q[R/x])$.

Example 2.2 Consider the CL-terms $Q \equiv x(\mathtt{S}yx)$ and $R \equiv \mathtt{SKK}$, then $\mathtt{FV}(Q) = \{x, y\}$ and $\mathtt{FV}(R) = \emptyset$, $Q[\mathtt{SS}/x] \equiv \mathtt{SS}(\mathtt{S}y(\mathtt{SS}))$ and $R[Q/x] \equiv R$.

There are some special combinations of the principal combinators $\mathtt{S}$ and $\mathtt{K}$, such as

$$\mathtt{B} := \mathtt{S}(\mathtt{KS})\mathtt{K} \qquad \mathtt{C} := \mathtt{S}(\mathtt{BS}(\mathtt{BKS}))(\mathtt{KK}) \qquad \mathtt{I} := \mathtt{SKK}$$

CL-terms built using only $\mathtt{K}$ and $\mathtt{S}$ are called *pure*; otherwise, they are called *applied*. Notice that some combinators can be built from others, for instance $\mathtt{C}$ uses the combinators $\mathtt{S}, \mathtt{K}$ and $\mathtt{B}$, therefore $\mathtt{C}$ is an applied combinator.

CL - A calculus for combinators

The logical theory of combinators, denoted as CL, consists of the ($\mathtt{SK}$)-axioms and the set of equality-rules, presented in Figure 1, where P, Q and R are arbitrary CL-terms.

($\mathtt{SK}$)-axioms:

$$\mathtt{K}PQ = P \qquad \mathtt{S}PQR = PR(QR) \qquad P = P$$

Equality-rules:

$$\frac{P = Q}{Q = P}\,[\mathtt{sym}] \qquad \frac{P = Q \quad Q = R}{P = R}\,[\mathtt{tr}] \qquad \frac{P = Q}{PR = QR}\,[\mathtt{clos_r}] \qquad \frac{P = Q}{RP = RQ}\,[\mathtt{clos_l}]$$

Fig. 1. CL: a calculus for combinators

Example 2.3 Consider the combinators $\mathtt{B} \equiv \mathtt{S}(\mathtt{KS})\mathtt{K}$ and $\mathtt{I} \equiv \mathtt{SKK}$. For all CL-terms M, N and P, $\mathtt{CL} \vdash \mathtt{I}M = M$ and $\mathtt{CL} \vdash \mathtt{B}MNP = M(NP)$ hold; also, $\mathtt{CL} \vdash \mathtt{C}MNP = MPN$ holds.

Different fragments of the combinatory logic can be created based on the identities between combinators used as axioms, these are called ($\mathtt{BCK}$)- and ($\mathtt{BCI}$)- combinatory logic:

($\mathtt{BCK}$)-axioms:	$\mathtt{B}PQR = P(QR)$	$\mathtt{C}PQR = PRQ$	$\mathtt{K}PQ = P$	$P = P$
($\mathtt{BCI}$)-axioms:	$\mathtt{B}PQR = P(QR)$	$\mathtt{C}PQR = PRQ$	$\mathtt{I}P = P$	$P = P$

2.2 Topological Graphs

A theory of *topological graphs* is developed with the aim to establish a correspondence between these graphs and the CL-terms. This correspondence works as a first approach to the intention of relating *topological graphs* to lambda-terms(, which are known to have a relation with combinators.)

Familiarity with the basic notions of topology as in [Mun15] is assumed.

To relate combinators and topological graphs, the notion of graphs proposed by Zeilberger in [Zei16] is extended. The notion of graphs adopted in this work allows "edges" that can be connected to more than two vertices. These edges are called whips and are used to represent multiple occurrences of the same variable in combinators.

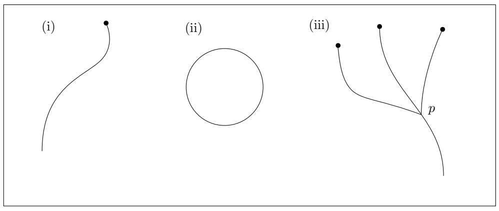
In order to define topological graphs, topological subspaces of $\mathbb{R}^2$ are of particular interest, they contain the "edges" of the topological graphs. The definition below describes objects that are homeomorphic to (i) segments of curves in $\mathbb{R}^2$, or (ii) to circles (S^1) or (iii) to "unions" of curves (whips), see figure in Definition 2.4.

Definition 2.4 [Topological Edges] Let $\mathcal{F} = \{\tau_i\}_{i\in I}$, be a non empty family of topological subspaces of $\mathbb{R}^2$ and a finite set $\mathcal{V} \subseteq \mathcal{G}_{\mathcal{F}} := \bigcup_{i\in I} \tau_i$, which is called the *vertex set* . For $\tau \in \mathcal{F}$, $\Delta\tau := \tau \cap \mathcal{V}$ denotes the set of vertices *incident* with τ, and $\tau^\circ := \tau \setminus \Delta\tau$ be *the interior* of τ. It is said that τ is a *topological edge* of $\mathbb{R}^2$ if and only if (see also figure).

(i) τ is homeomorphic to the interval $[0, 1]$; or

(ii) τ is homeomorphic to $S^1 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$; or

(iii) τ is an n-whip that is a compact and connected topological space with $n \geq 1$ and a unique point $p_\tau \in \tau$ such that $\tau^\circ \setminus \{p_\tau\}$ has $n+1$ connected components homeomorphic to $(0, 1)$ or $[0, 1)$.



A set $\mathcal{T} = \{\tau_i\}_{i\in I}$ of topological edges is called a *conformal* if, in addition, the following property holds: for all topological edge $\tau, \tau' \in \mathcal{T}$, if $\tau \cap \tau'$ is infinite then $\tau = \tau'$. This condition is technical and guarantees that there is no infinite overlap of topological edges.

Remark 2.5 Vertices are points in $\mathbb{R}^2$ depicted as bullets (•). In figure in Definition 2.4, the topological edge (i) has one vertex, (ii) has no vertices and (iii) has three vertices. Every vertex is a point in $\mathbb{R}^2$ but not every point is a vertex. There is a point at the opposite ending of the edge subspace (i). This distinction is fundamental for the notions of edges and free edges.

The notion of *topological graph* differs from the usual notion of graph by allowing *free edges* and whips.

Definition 2.6 [Topological Graph] Let $\mathcal{T} = \{\tau_i\}_{i\in I}$ be a non-empty, finite and conformal set of topological edges, and $\mathcal{V} \subseteq \mathcal{G}_{\mathcal{T}} := \bigcup_{i\in I} \tau_i$ be a finite vertex set. The pair $\mathcal{G} = \langle\mathcal{V}, \mathcal{T}\rangle$ is a *topological graph* if and only if

(i) for each $\tau \in \mathcal{T}$, τ° is a connected topological space.

(ii) if τ is homeomorphic to S^1 then $|\Delta\tau| = 1$, and τ is called *loop*.

(iii) if τ is homeomorphic to $[0, 1]$ then $|\Delta\tau| = 2$ or 1, and τ is called an *edge* or *free edge*(*half edges* in [Zei16]), respectively.

(iv) if τ is an n-whip then $1 \leq |\Delta\tau| \leq n$. In this case, τ is called a *bound* n-whip if and only if $\tau^\circ \setminus \{p_\tau\}$ have no connected component homeomorphic to $[0, 1)$; otherwise, it is called *free* n-whip.

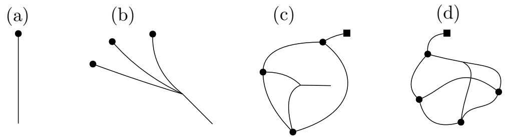
$\mathcal{G}_{\mathcal{T}}$ is said to be the *graph drawing* of $\mathcal{G}$. The *boundary* of $\mathcal{T}$, denoted by $\partial\mathcal{T}$ is the set of all free edges and free n-whips of $\mathcal{T}$.

Remark 2.7 A topological graph $\mathcal{G} = \langle\mathcal{V}, \mathcal{T}\rangle$, by definition above, must have at least one *topological edge* in the sense of Definition 2.4 and $\Delta\tau \cap \mathcal{V} \neq \emptyset$ for each $\tau \in \mathcal{T}$. By abuse of notation, the pair $\langle\emptyset, \{\tau\}\rangle$, for τ a topological edge, is called a *degenerated topological graph.*

Remark 2.8 It is easy to see that, for each $v \in \mathcal{V}$, $\mathcal{T}(v) := \{\tau \in \mathcal{T} \mid v \in \Delta\tau\}$, i.e., the set of edges incident to v, is finite.

Definition 2.9 [Rooted Topological Graph] Let $\mathcal{G} = \langle\mathcal{V}, \mathcal{T}\rangle$ be a topological graph and $r \in \mathcal{V}$ a distinguished vertex. The triple $\langle\mathcal{V}, \mathcal{T}, r\rangle$ is called a *rooted topological graph*, denoted by $\mathcal{G}(r)$, if and only if, there exists a unique topological edge $\tau_r \in \mathcal{T}$ such that $r \in \Delta\tau_r$ and τ_r is neither a loop nor a n-whip. In this case, τ_r is called a *rooted edge*, r is the *root* and $v \in \Delta\tau_r \setminus \{r\}$ is the *rooted vertex.*

Example 2.10 Four (rooted) topological graphs are given. Bullets are vertices and black squares roots.



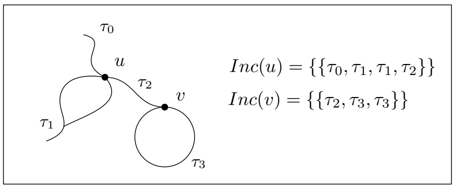
(i) is the simplest topological graph with one vertex and one free edge.
(ii) is a free 3-`whip`.
(iii) is a rooted topological graph with 4 vertices and a free 2-`whip`.
(iv) is a rooted topological graph with 5 vertices and one bound 2-`whip`.

The notions of planarity and connectivity are different because rooted topological graphs are not only concerned about the relation between vertices and edges, but also with the drawing of the graph. Therefore, the notions of vertex *degree*, connectedness and planarity for topological graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ should be established.

Definition 2.11 [Vertex Incidence, Vertex Degree] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a topological graph, $v \in \mathcal{V}$ a vertex and $\tau \in \mathcal{T}(v)$. $\texttt{Inc}(v)$ is defined as the multiset over $\mathcal{T}(v)$ denoting the incidences of each $\tau \in \mathcal{T}(v)$ on v, where $\texttt{Inc}(v)(\tau)$ is defined as: **(i)** if τ is an edge or free edge then $\texttt{Inc}(v)(\tau) = 1$; **(ii)** if τ is a loop then $\texttt{Inc}(v)(\tau) = 2$; **(iii)** if τ is a n-`whip`, then $\texttt{Inc}(v)(\tau)$ is equal the number of connected components of $\tau^\circ \setminus \{p_\tau\}$ that have v as an accumulation point.

Finally, $\texttt{deg}_{\mathcal{G}}(v) = |\texttt{Inc}(v)|$ is defined as the vertex degree of v. For simplicity, the subindex $\mathcal{G}$ is omitted when $\mathcal{G}$ is clear from the context.

Example 2.12 Given a topological graph $\langle \mathcal{V}, \mathcal{T} \rangle$ where $\mathcal{V} = \{u, v\}$ and $\mathcal{T} = \{\tau_0, \tau_1, \tau_2, \tau_3\}$ as in the figure on the right, the multiset $\texttt{Inc}(v)$ counts how many topological edges are incident with v, including multiple incidence by the same topological edge.



In the remaining of this paper it is assumed that $\texttt{Inc}(v)$ is ordered in a counter-clockwise manner.

Definition 2.13 [Adjacent Vertices, Path] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a topological graph and $u, v \in \mathcal{V}$. The vertices u and v are called *adjacent vertices* in $\mathcal{G}$, and denoted by $u \leftrightarrow v$, if and only if $u = v$ or there exists an edge $\tau \in \mathcal{T}$ such that $u, v \in \Delta\tau$. An (u, v)-*path* in $\mathcal{G}$, denoted as $u \overset{*}{\leftrightarrow} v$, is a finite sequence of adjacent vertices $u = u_0, u_1, \ldots, u_n = v$.

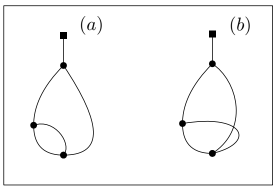
Definition 2.14 [Connected Graph] $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ is a *connected graph* if and only if for each $u, v \in \mathcal{V}$ one has $u \overset{*}{\leftrightarrow} v$.

In the example 2.10, (a), (c) and (d) are connected topological graphs whereas (b) is not since any pair of different vertices in (b) is disconnected (because only edges are considered in the definition).

Remark 2.15 For a rooted topological graph $\mathcal{G}(r)$ connectedness adapts straightforwardly, since $r \in \mathcal{V}$. A degenerated topological graph $\langle \emptyset, \{\tau\} \rangle$ is connected since it does not contain any vertices.

Definition 2.16 [Planar Topological Graph] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a topological graph. $\mathcal{G}$ is *non planar* if and only if there exists $\tau, \gamma \in \mathcal{T} \setminus \partial\mathcal{T}$ such that $\tau \neq \gamma$ and $\tau^\circ \cap \gamma^\circ \neq \emptyset$; otherwise, $\mathcal{G}$ is called *planar*.

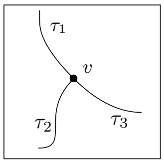
Example 2.17 In the figure, (a) is a planar rooted topological graph while (b) is non planar.



In the sequel, unless stated otherwise, by a *graph* it should be understood a *topological graph*.
The definition of L-function below provides an ordering between the edges of a non-degenerated graph.

Definition 2.18 [L-function] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a non-degenerated graph and $\tau \in \partial\mathcal{T}$ a free edge such that $\Delta\tau = \{v\}$, i.e., v is the only incident vertex to τ. The function L_v maps the integers in $\{0, 1, \ldots, \texttt{deg}(v) - 1\}$ to the the edges $\tau \in \mathcal{T}(v)$ that are incident to v, that is, $L_v : \{0, 1, \ldots, \texttt{deg}(v) - 1\} \to \mathcal{T}(v)$ such that $L_v(0) = \tau$ and $L_v(0), \ldots, L_v(\texttt{deg}(v) - 1)$ in counter-clockwise manner.

Example 2.19 Consider the connected graph $\mathcal{G}$, with a singleton vertex set $\{v\}$ and $\texttt{deg}_{\mathcal{G}}(v) = 3$. Notice that $L : \{0, 1, 2\} \to \{\tau_1, \tau_2, \tau_3\}$, and starting from τ_1, it follows by definition that $L_v(0) = \tau_1$, $L_v(1) = \tau_2$ and $L_v(2) = \tau_3$ (following the counter clockwise orientation of the edges). Starting with edge τ_3 instead, would give $L_v(0) = \tau_3, L_v(1) = \tau_1$ and $L_v(2) = \tau_2$.

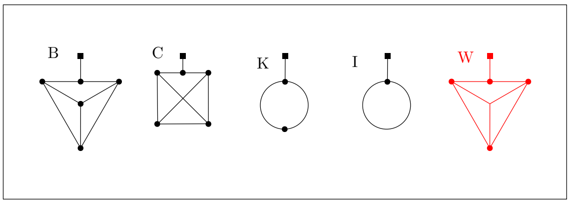


3 Combinatorial Graphs

The subclass of *combinatorial graphs*, for short CL-graphs, is introduced. They consist of rooted and connected topological graphs in which vertices have degree two or three, without **whip**s. Formally,

Definition 3.1 [BCK-graph, BCI-graph] Let $\mathcal{G}(r) = \langle \mathcal{V}, \mathcal{T}, r\rangle$ be a rooted connected graph and $\tau_r \in \texttt{Inc}(r)$. Then $\mathcal{G}(r)$ is called a BCK-graph if and only if $\mathcal{G}(r)$ holds the following properties: (i) for each $v \in \mathcal{V} \setminus \{r\}$, $\texttt{deg}(v) = 2$ or 3; and (ii) there is no n-**whip** in $\mathcal{G}(r)$. In addition, $\mathcal{G}(r)$ is called a BCI-graph if and only if $\mathcal{G}(r)$ is a BCK-graph and for each $v \in \mathcal{V} \setminus \{r\}$, $\texttt{deg}_{\mathcal{G}}(v) = 3$. BCI and BCK graphs are called *combinatorial graphs.*

Example 3.2 According to the Definition3.1, B, C and I are BCI graphs whereas K is a BCK graph. However, W is neither a BCI nor a BCK-graph, since it has a **whip**.



It is well-known that combinators have a correspondence with λ-terms, for instance, $\texttt{B} = \lambda xyz.x(yz)$, $\texttt{C} = \lambda xyz.xzy$, $\texttt{K} = \lambda xy.x$, $\texttt{I} = \lambda x.x$, and $\texttt{W} = \lambda xy.(xy)y$ (the combinator with axiom $\texttt{W}PQ = PQQ$). It is easy to see that the graphs B, C, K, I and W above correspond to the combinators B, C, K and W, respectively.

This subclass of topological graphs is limited: the absence of **whip** does not allow one to represent non-linear λ-terms. Therefore, some extension are necessary.

3.1 Generalised CL-graphs

The class of generalised CL-graphs ($\texttt{CL}^g$-graphs, for short) is introduced. This class contains the CL-graphs and consist of connected graphs containing at least one free edge and vertices of degree either two or three, i.e., the requirement of non-existence of n-**whip**s in CL-graphs is removed. Also,the rooted edge condition is replaced by the existence of free-edges: this gives flexibility for choosing any free edge as the rooted edge.

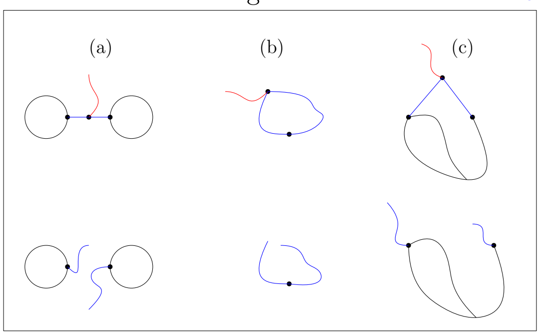
Definition 3.3 [Generalised CL-graph] A graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{T}\rangle$ is called a *generalised* CL*-graph* ($\texttt{CL}^g$-graph) if and only if the following conditions hold: • $\mathcal{G}$ is connected; • each vertex $v \in \mathcal{V}$ is such that $\texttt{deg}(v) = 2$ or $\texttt{deg}(v) = 3$; and, • there exists at least one free edge $\tau \in \partial\mathcal{T}$.

A rooted topological graph $\mathcal{G}(r) = \langle \mathcal{V}, \mathcal{T}, r\rangle$ is called a *rooted* $\texttt{CL}^g$-graph if after removing the root r, $\mathcal{G} = \langle \mathcal{V}, \mathcal{T}\rangle$ is a $\texttt{CL}^g$-graph.

Degenerated graphs are considered as $\texttt{CL}^g$-graphs. Graphs (c) and (d) in Example 2.10, by removing the root, are $\texttt{CL}^g$-graphs.

Lemma 3.4 *Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T}\rangle$ be a $\texttt{CL}^g$-graph and $\tau \in \partial\mathcal{T}$ be a free edge. If $|\mathcal{T}| > 1$ and $\mathcal{G}' = \langle \mathcal{V} \setminus \Delta\tau, \mathcal{T} \setminus \{\tau\}\rangle$ then $\mathcal{G}'$ is a $\texttt{CL}^g$-graph or the union of two $\texttt{CL}^g$-graphs.*

Example 3.5 Notice that the rooted graph in Example 2.10(d) is a $\texttt{CL}^g$-graph, if the root vertex, say r, is removed, and the edge incident to the root, say τ_r, becomes free in (d), denote such a graph as $\mathcal{G} = \langle \mathcal{V}, \mathcal{T}\rangle$. Also, the graph $\mathcal{G}' = \langle \mathcal{V} \setminus \{\Delta\tau_r\}, \mathcal{T} \setminus \{\tau_r\}\rangle$ obtained by eliminating this free edge and the vertex in its boundary is also connected. The figure illustrates Lemma 3.4.



(a) is the rooted version of *Dumbbell graph*: removing the free edge and its incident vertex, two (disjoint) connected graphs are obtained.

(b) represents the K combinator (Example 3.2). Note that by removing the red free edge, a connected graph is obtained.

(c) by removing its free edge one obtains a disconnected graph: **whip**s do not connect vertices.

3.2 Graph Decomposition and ω-graph

Here a decomposition operation on topological graphs is introduced. This operation allows us to develop a (topological) graph characterization of non-linear λ-terms. ω-graphs are obtained from a "decomposition property" imposed on $\mathtt{CL}^g$-graphs.

To start, a *graph decomposition* operation is defined on non-degenerated graphs $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ w.r.t. a free edge $\tau \in \partial\mathcal{T}$. Therefore, graph decomposition operation is non-deterministic since $\mathcal{G}$ may have more than one vertex in $\partial\mathcal{T}$ and a choice can be made for τ. The graph decomposition consists of the application of rules ($\mathtt{D-abs}$) or ($\mathtt{D-app}$), based on the degree of vertices and their connectivity properties. The intuition of the rules is given below and after that, they are formally introduced (in Definition 3.6). Starting from a choice of $\tau \in \partial\mathcal{T}$, such that $\Delta\tau = \{v\}$:

- ($\mathtt{D-abs}$): this rule is applied when v has degree 2, or it has degree 3 and the L-function is associated to edges such that $L_v(1), L_v(2)$ are not in $\partial\mathcal{T}$, and if v and τ are removed from $\mathcal{G}$ the resulting graph is connected or $L_v(1)$ is a loop. See, for instance, the first decomposition step in Figure 2.
- ($\mathtt{D-app}$): this rule is applied when the v has degree 3, and rule ($\mathtt{D-abs}$) cannot be applied. That is, one has either $L_v(1)$ or $L_v(2)$ in $\partial\mathcal{T}$ (first ($\mathtt{D-app}$) in Figure 2), or when removing τ and v the resulting graph is not connected. Each application of this rule decomposes the graph into two subgraphs.

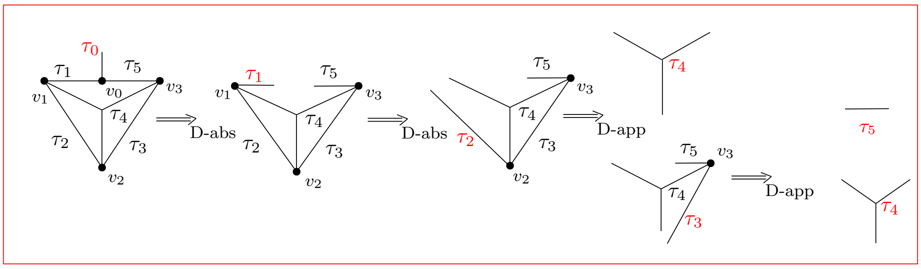


Fig. 2. A $\mathtt{CL}^g$- graph decomposition.

Definition 3.6 [Decomposition step] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a non-degenerated graph and $\tau \in \partial\mathcal{T}$ a free edge such that $\Delta\tau = \{v\}$. Suppose that $L_v(1)$ is not a bound n-$\mathtt{whip}$, a *decomposition step*, denoted as $\Longrightarrow_D$, is defined via the application of the one of the following rules, where $\mathcal{T}' = \mathcal{T} \setminus \{\tau\}$ and $\mathcal{V}' = \mathcal{V} \setminus \Delta\tau$:

($\mathtt{D-abs}$) $\quad (\boldsymbol{\mathcal{T}, \mathcal{V}, \tau}) \Longrightarrow_D (\boldsymbol{\mathcal{T}', \mathcal{V}', L_v(1)})$, if $\mathtt{deg}_{\mathcal{G}}(v) = 2$; or $\mathtt{deg}_{\mathcal{G}}(v) = 3, L_v(1), L_v(2) \notin \partial\mathcal{T}$ and $\langle \mathcal{T}', \mathcal{V}' \rangle$ is a connected graph or $L_v(1)$ is a loop.

In this case v is called an $\mathtt{abs}$-*node* and denoted as (λ)-node in $\langle \mathcal{V}, \mathcal{T} \rangle$ w.r.t. τ.

($\mathtt{D-app}$) $\quad (\boldsymbol{\mathcal{T}, \mathcal{V}, \tau}) \Longrightarrow_D (\boldsymbol{\mathcal{T}_i, \mathcal{V}_i, L_v(i)})$, if the following conditions hold:

(i) $\mathtt{deg}_{\mathcal{T}}(v) = 3, \mathcal{V}' = \mathcal{V}_1 \dot\cup \mathcal{V}_2$,

(ii) if $\mathcal{V}_1 \neq \emptyset$ and $\mathcal{V}_2 \neq \emptyset$ then, for each $w_1 \in \mathcal{V}_1, w_2 \in \mathcal{V}_2$, w_1 and w_2 are not connected in $\langle \mathcal{T} \setminus \{\tau\}, \mathcal{V} \setminus \{v\} \rangle$;

(iii) if $\mathcal{V}_i = \emptyset$ then $L_v(i) \in \partial\mathcal{T}$ for some $i \in \{1, 2\}$;

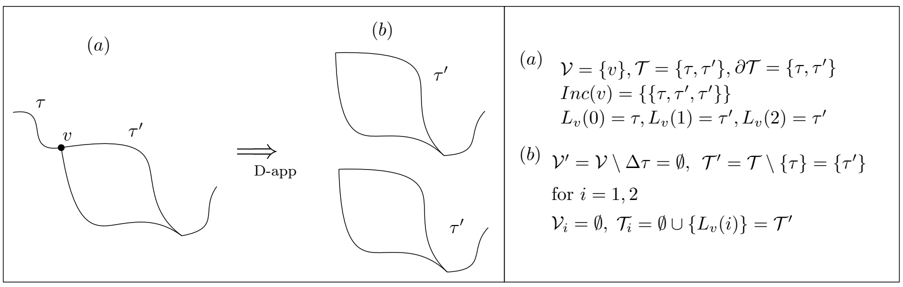
(iv) $\mathcal{T}_1 \cap \mathcal{T}_2$ does not contain a bound n-$\mathtt{whip}$;

where $\mathcal{T}_i = \{\tau' \in \mathcal{T}' \mid \tau' \cap \mathcal{V}_i \neq \emptyset\} \cup \{L_v(i)\}$, for $i = 1, 2$, and in this case, v is called an $\mathtt{app}$-*node* and denoted as (@)-node in $\langle \mathcal{V}, \mathcal{T} \rangle$ with relation to τ.

When executing the decomposition steps, condition (iv) in rule $\mathtt{D-app}$, avoids n-$\mathtt{whip}$s being bound. As it will be shown in the paper, this guarantees no collapsing variables in the corresponding λ-term. Condition (iii) in rule $\mathtt{D-app}$ forces that rule $\mathtt{D-abs}$ cannot be applied.

Example 3.7 This example illustrates how rule $\mathtt{D-app}$ works.

The $\mathtt{CL}^g$-graph in (a) becomes a duplicated degenerated graph in (b). Notice that $\mathtt{deg}_{\mathcal{T}}(v) = 3$, $\mathcal{V}' = \mathcal{V} \setminus \{v\} = \emptyset = \mathcal{V}_1 \dot\cup \mathcal{V}_2$, by vacuity, vertices in $\mathcal{V}_1$ and $\mathcal{V}_2$ are not connected in $\langle \mathcal{V}', \mathcal{T}' = \mathcal{T} \setminus \{\tau\} \rangle$. Since $\mathcal{V}_1 = \emptyset$, one has $\{\tau' \in \mathcal{T}' \mid \tau' \cap \mathcal{V}_1 \neq \emptyset\} = \emptyset$, therefore, by definition, $\mathcal{T}_1 = \{L_v(1)\} = \{\tau'\}$. Similarly, $\mathcal{T}_2 = \{L_v(2)\} = \{\tau'\}$.



As usual $\overset{+}{\Longrightarrow}_D$ and $\overset{*}{\Longrightarrow}_D$ denote the transitive and reflexive-transitive closure of $\Longrightarrow_D$. A *normal form* of $(\mathcal{T}, \mathcal{V}, \tau)$ is a triple $(\mathcal{T}', \mathcal{V}', \tau')$ such that $(\mathcal{T}, \mathcal{V}, \tau) \overset{*}{\Longrightarrow}_D (\mathcal{T}', \mathcal{V}', \tau')$ and no rule can be applied to $(\mathcal{T}', \mathcal{V}', \tau')$.

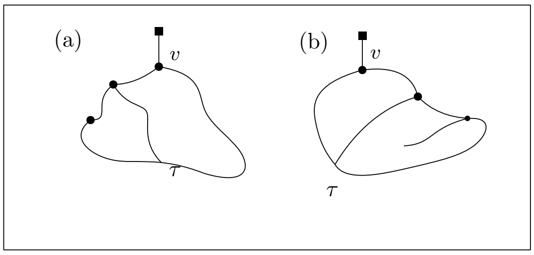
Lemma 3.8 *The relation given by the decomposition step $\Longrightarrow_D$ is terminating.*

Theorem 3.9 (D-consistency) *Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a non degenerated $\mathtt{CL}^g$-graph and $\tau \in \partial\mathcal{T}$ a free edge. If $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_D (\mathcal{T}', \mathcal{V}', \tau')$ then $\tau' \in \mathcal{T}'$ and $\mathcal{G}' = \langle \mathcal{V}', \mathcal{T}' \rangle$ is a $\mathtt{CL}^g$-graph or a degenerated graph.*

In [Zei16] decomposition was also proposed, but the distinguishing feature here is the capability to deal with multiple occurrences of the *same* (free-)edges corresponding to the same variable, using n-`whips`.

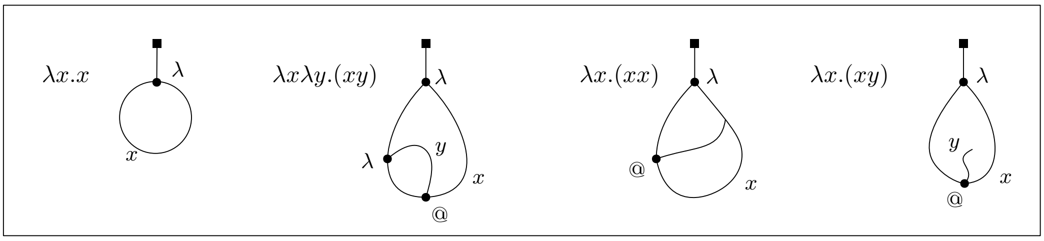
Definition 3.10 [ω-graph] Let $\mathcal{G}(r) = \langle \mathcal{V}, \mathcal{T}, r \rangle$ be a rooted $\mathtt{CL}^g$-graph and $\tau_r \in \mathtt{Inc}(r)$. Then $\mathcal{G}(r)$ is called an ω-graph if and only if each normal form of $(\mathcal{T}, \mathcal{V} \setminus \{r\}, \tau_r)$ with relation to $\Longrightarrow_D$ is a degenerated graph.

Example 3.11 Graph in Figure 2 is an ω-graph. In the figure, (a) is an ω-graph whereas (b) is not since it has a bound `whip` as $L_v(1)$.



From λ-terms to ω-graphs

Construction of a graph and a string diagram from a given λ-term is similar. The rooted topological graphs for $\lambda x.x$, $\lambda x \lambda y.(xy)$, and $\lambda x.(xy)$ are obtained by associating bound variables to edges connected to a (λ)-node and free variables correspond to free edges. However, with this extension one can draw the topological graph corresponding to $\lambda x.(xx)$ using a 2-`whip` to represent the two bounded occurrences of variable x.



ω-graphs correspond to λ-terms without variable clashes; e.g. $(x\lambda x \lambda y.(xy))$ does not have an associated ω-graph. Therefore, $\Longrightarrow_{\mathcal{D}}$ can be used as an ω-graph checker. The graph in Figure 2 corresponds to $\lambda xy.((xy)y)$.

The notions of `BCI`- `BCK`- and ω-graph illustrate the importance of vertex orientation; the reflection of `BCI`- and `BCK`-graphs results in `BCI`- `BCK`-graphs the same is not true for ω-graphs, one can introduce bound n-`whip`.

4 Operations on combinatorial graphs

This section establishes operations on ω-graphs that generate only ω-graphs. To start, some standards on graph drawing are established. Let $\mathcal{G}(r)$ be a rooted graph, the choice of one of the three drawings in Figure 3 makes explicit the existence of a particular edge, or vertex or `whip`, as the interest relies on one of these objects. $\mathcal{G}_{\mathcal{T}}$ denotes the omitted parts of the drawing of $\mathcal{G}(r)$.

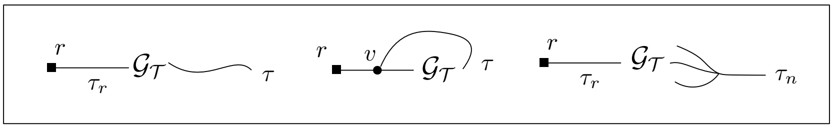
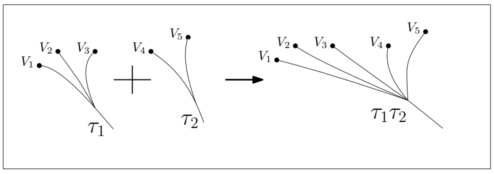


Fig. 3. First: r is the root and τ_r is the only edge connected to r, called *rooted edge*, τ a free edge. Second: v is a vertex called *rooted vertex*, τ is an edge connected to v. Third: τ_n is a n-free `whip`

Definition 4.1 [`whip`-junction] Let $\mathcal{G}_1(r_1) = \langle \mathcal{V}_1, \mathcal{T}_1, r_1 \rangle$ and $\mathcal{G}_2(r_2) = \langle \mathcal{V}_2, \mathcal{T}_2, r_2 \rangle$ be rooted graphs, τ_1 and τ_2 be n and m-`whips` in $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. $\tau_1\tau_2$ is called *a junction of* τ_1 *and* τ_2 if and only if $\tau_1\tau_2$ is an $(n+m)$-`whip` respecting the incidence of each vertex belonging to $\Delta\tau_i$, $i = 1, 2$, and $\Delta(\tau_1\tau_2) = \Delta\tau_1 \cup \Delta\tau_2$.

Example 4.2

Let τ_1, τ_2 be a 3-`whip` and 2-`whip`, respectively. The junction of $\tau_1\tau_2$ is depicted in the figure on the right. Note that $\tau_1\tau_2$ contains the combined number of incidences on each vertex from τ_1 and τ_2.



$\mathbb{G}(\omega)$: an ω-graph generator

The calculus $\mathbb{G}(\omega)$ introduced in Figure 4 generate ω-graphs. Rules operate on triples of the form $(\partial\mathcal{T}, \mathcal{V}, r)$ consisting of a set of free edges, a set of vertices, and a root vertex, respectively.

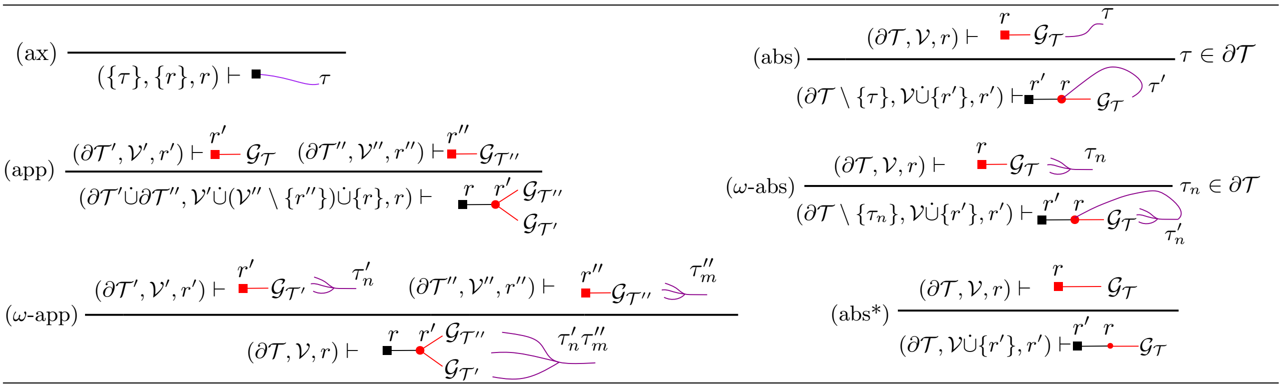


Fig. 4. $\mathbb{G}(\omega)$: a calculus to generate ω-graphs.

Intuitively, rule `(app)` takes two rooted graphs $\mathcal{G}_{\mathcal{T}'}(r')$ and $\mathcal{G}_{\mathcal{T}''}(r'')$ and produces a new rooted graph by colliding the vertex r'' into r' and connecting a new root vertex r to r'. Rule `(abs)` takes as premise a rooted graph $\mathcal{G}_{\mathcal{T}'}(r)$ with a free edge τ and produces a new graph by connecting τ to r and adding a new root r' connected to r. Rule `(abs*)` is analogous to `(abs)` but no free edge is connected to r; this corresponds to vacuous abstractions, when one looks at the correspondence with λ-terms. Rule `(`ω`-app)` takes as premises rooted graphs $\mathcal{G}_{\mathcal{T}'}(r')$ and $\mathcal{G}_{\mathcal{T}''}(r'')$ and produces a new graph in which r' and r'' are collapsed in one vertex, say r', a new root r is added and the n- and m-`whips`, τ'_n and τ''_m are gathered together using the *junction* operator, obtaining an $(n+m)$-`whip` $\tau'_n\tau''_m$. This rule includes multiple junctions, that is, for n_i-`whips` say $\tau'_{n_1}, \ldots, \tau'_{n_k} \in \partial\mathcal{T}'$ and m_j-`whips`, say $\tau''_{m_1}, \ldots, \tau''_{m_k} \in \partial\mathcal{T}''$, pairwise distinct, one can conclude junctions $\tau'_{n_i}\tau''_{m_i}$ for each $i = 1, \ldots, k$. The description of the remaining rules is self-explanatory and is omitted.

Notation 2 $(\partial\mathcal{T}, \mathcal{V}, r) \vdash_{\mathbb{G}} \mathcal{G}_{\mathcal{T}}$ *denotes a derivation using rules in* $\mathbb{G}(\omega)$. *When the calculus is clear from context,* $\mathbb{G}$ *is dropped from* $\vdash_{\mathbb{G}}$. *In the following, rules (*ω*-app) and (*ω*-abs) are referenced as* ω*-rules.*

Example 4.3 Figure 5 illustrates derivations of two ω-graphs, using rules of the calculus $\mathbb{G}(\omega)$. The first derivation produces the combinatorial graph for `K` whereas the second for `W` (see Example 3.2).

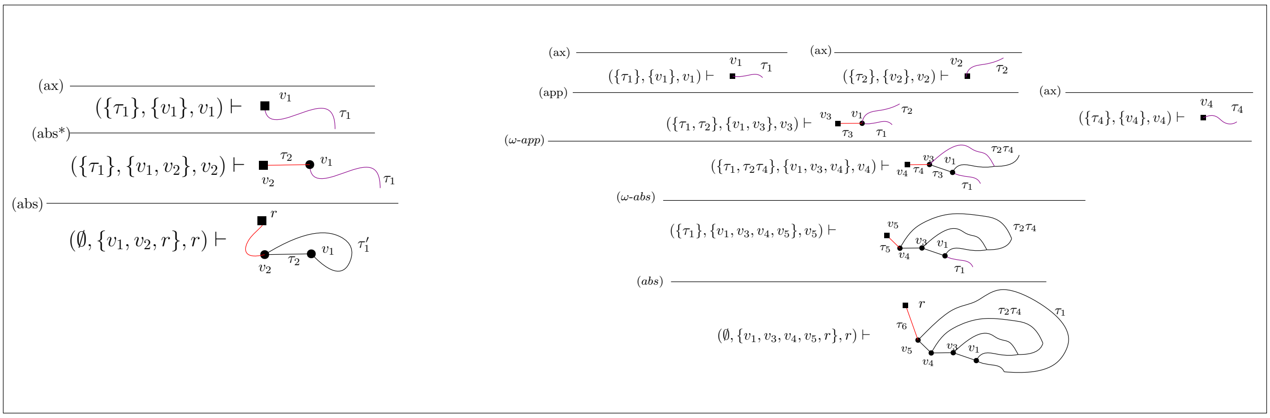


Fig. 5. Red edges are new edges included and purple object are the free edges and free `whips`

The following result concern correctness of the calculus $\mathbb{G}(\omega)$ regarding the generation of ω-graphs.

Definition 4.4 [Topological Graph Isomorphism] Let $\langle\mathcal{V},\mathcal{T},r\rangle$ and $\langle\mathcal{V}',\mathcal{T}',r'\rangle$ be ω-graphs and $\varphi:\mathcal{T}\to\mathcal{T}'$ and $\psi:\mathcal{V}\to\mathcal{V}'$ mappings between edges and vertices, respectively. We say that the 2-map cell (φ,ψ) is an isomorphism if and only if the following properties hold:

(i) φ and ψ are bijections and $\psi(r)=r'$;

(ii) $\varphi(\tau)$ is homeomorphic to τ, for all $\tau\in\mathcal{T}$;

(iii) $\psi(\Delta\tau)=\Delta\varphi(\tau)$, for all $\tau\in\mathcal{T}$;

(iv) $\varphi(\texttt{Inc}(v))=\texttt{Inc}(\psi(v))$, for all $v\in\mathcal{V}$. Besides, $\varphi(\texttt{Inc}(v))$ and $\texttt{Inc}(\psi(v))$ have the same clockwise orientation.

Theorem 4.5 (Correctness) *Let $\mathcal{G}(r)=\langle\mathcal{V},\mathcal{T},r\rangle$ be a rooted graph.*

(i) *(Soundness) If $(\partial\mathcal{T},\mathcal{V},r)\vdash\mathcal{G}_{\mathcal{T}}$ then $\mathcal{G}(r)$ is an ω-graph. Moreover, if only rules (ax), (app) and (abs) rules are then $\mathcal{G}(r)$ is a BCI-graph and, if the ω-rules are not used, then $\mathcal{G}(r)$ is a BCK-graph.*

(ii) *(Completeness) If $\mathcal{G}(r)$ is an ω-graph, then there exists an ω-graph $\langle\mathcal{V}',\mathcal{T}',r'\rangle$ isomorphic to $\mathcal{G}(r)$ such that $(\partial\mathcal{T}',\mathcal{V}',r')\vdash\mathcal{G}_{\mathcal{T}'}$.*

Proof. [sketch] (i) The proof is by induction on the derivation ∇ of $(\partial\mathcal{T},\mathcal{V},r)\vdash\mathcal{G}_{\mathcal{T}}$, by analysing the last rule applied in ∇. If **The rule is** (app), by definition one has $\partial\mathcal{T}=\partial\mathcal{T}'\cup\partial\mathcal{T}''$, $\mathcal{V}=\mathcal{V}'\cup(\mathcal{V}''\setminus\{r''\}\cup\{r\})$ and there exist derivations ∇' and ∇'' of $(\partial\mathcal{T}',\mathcal{V}',r')\vdash\mathcal{G}_{\mathcal{T}'}(r')$ and $(\partial\mathcal{T}'',\mathcal{V}'',r'')\vdash\mathcal{G}_{\mathcal{T}''}(r'')$, respectively. By IH, $\mathcal{G}_{\mathcal{T}'}(r')$ and $\mathcal{G}_{\mathcal{T}''}(r'')$ are ω-graphs. Notice that $\mathcal{G}_{\mathcal{T}}(r)$ is obtained by colliding the roots of $\mathcal{G}_{\mathcal{T}'}(r')$ and $\mathcal{G}_{\mathcal{T}''}(r'')$ into vertex r' and maintaining all other connections and adding a new root vertex r and a new edge,say τ_r, connecting r and r'. So $\texttt{deg}_{\mathcal{T}}(r')=3$ and r' and connects all other vertices in $\mathcal{V}'$ and $\mathcal{V}''$. Therefore, $\texttt{deg}_{\mathcal{T}}(v)=2$ or 3 for all $v\in\mathcal{V}\setminus\{r\}$ and $\mathcal{G}_{\mathcal{T}}$ is connected.

It remains to show that all normal forms of $(\mathcal{T},\mathcal{V},\tau_r)$ w.r.t. $\Longrightarrow_D$ correspond to degenerated graphs. First, notice that no bound **whip** is added. Also, since $\texttt{deg}_{\mathcal{T}}(r')=3$ one could apply rules (D-app) or (D-abs). However, $\langle\mathcal{V}\setminus\{r'\},\mathcal{T}\setminus\{\tau_r\},L_v(1)\rangle$ is disconnected; therefore, the rule applied should be (D-app) and $(\mathcal{T},\mathcal{V},\tau_r)\Longrightarrow_D(\mathcal{T}_i,\mathcal{V}_i,L_v(i))$, which corresponds to ω-graphs $\mathcal{G}_{\mathcal{T}'}$ and $\mathcal{G}_{\mathcal{T}''}$, respectively for $i=1,2$, and their normal forms w.r.t. $\Longrightarrow_D$ correspond also to degenerated graphs.

□

The following results demonstrates that there is an one-to-one correspondence between the class of λ-terms without variable clash modulo α-equivalence and free variables renaming and the class of ω-graphs modulo topological graph isomorphism.

Theorem 4.6 (ω-graph to λ-term) *For each ω-graph exists only one λ-term M without variable-clashes, modulo α-conversion and free variables renaming and a surjective function $f:\mathcal{T}\mapsto Subterms(M)$ such that for each $v\in\mathcal{V}\backslash\{r\}$ where $\tau:=L_v(0),\tau':=L_v(1)$ and (when $\texttt{deg}_{\mathcal{T}}(v)=3$) $\tau'':=L_v(2)$, the following properties hold for f:*

(i) *For each $\gamma,\gamma'\in\partial\mathcal{T}$ distinct, $f(\gamma),f(\gamma')$ are distinct variables.*

(ii) *if v is an* app*-node then, $f(\tau)=(f(\tau')f(\tau''))$*

(iii) *if v is an* abs*-node and $\texttt{deg}_{\mathcal{T}}(v)=3$ then, $f(\tau)=\lambda f(\tau'')\cdot f(\tau')$ where $f(\tau'')$ is a variable.*

(iv) *if v is an* abs*-node and $\texttt{deg}_{\mathcal{T}}(v)=2$ then, $f(\tau)=\lambda x_v\cdot f(\tau')$, where x_v is a variable.*

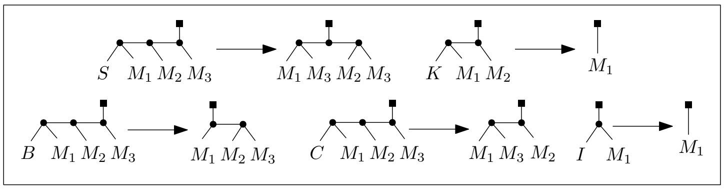
Proof. The proof follows from theorem 4.5 and using induction on the calculus $\mathbb{G}(\omega)$. □

Theorem 4.7 (λ-term to ω-graph) *For each λ-term M without variable clash there exist an ω-graph $\mathcal{G}(r)=\langle\mathcal{V},\mathcal{T},r\rangle$ and a surjective function $f:\mathcal{T}\mapsto Subterms(M)$ such that f has the properties listed in the theorem 4.6 and $f(\tau_r)=M$.*

Proof. The proof is by induction on λ-term construction, applying the the calculus $\mathbb{G}(\omega)$ and using the theorem 4.5. □

From the combinatorial graphs it is possible to create a terminating graph rewriting system which mimics the behavior of BCK and BCI axioms. For the graph rewrite rules, the notion of *graph redex* of a graph G is defined as a particular subgraph of G. These subgraphs simulate the action of *combinatorial graphs* over topological graphs. In the definition below the combinators B, C, K, S and I are identified with their corresponding combinatorial graphs.

Definition 4.8 [Graph Redex] Let B, C, K, S and I be the combinatorial graphs and M_1, M_2 and M_3 connected topological graphs with empty frontier. The graph rewrite rules over the topological graphs are defined as



It is easy to see that considering only B, C, K and I one has a terminating rewriting graph system.

Proposition 4.9 *Let M, N be graphs such that $M \longrightarrow N$. If M is a BCK-graph (BCI-graph) then N is a BCK-graph (BCI-graph).*

Proof. [sketch] Let M be a BCK-graph such that $M \longrightarrow N$, so M has a B, C, K or I subgraph. For simplicity, suppose it is a B-redex, then by the Theorem 4.5, M has a proof tree in the graph calculus $\mathbb{G}(\omega)$ in which ω-rules are not used. Let $(((\mathtt{B}M_1)M_2)M_3)$ be the B-redex of M, then M_1, M_2 and M_3 are BCK-graphs and also have proofs in $\mathbb{G}(\omega)$. By Theorem 4.5, $((M_1M_2)M_3)$ is a BCK-graph, therefore N is obtained from M by replacing $(((\mathtt{B}M_1)M_2)M_3)$ by $((M_1M_2)M_3)$ and then N is a BCK-graph. The other cases are analogous □

5 PTG: a typing algorithm for graphs

An algorithm for simple type assignment is given. It is assumed reader's familiarity with the notion of typability (e.g., see [Hin97]).

Definition 5.1 [Types, Type substitution] Let $\mathbb{A}$ be an countably infinite set of *type variables*. The set of *Types* built over $\mathbb{A}$, denoted by $\mathtt{Type}(\mathbb{A})$, is defined inductively as: **(i)** each $a \in \mathbb{A}$ is a type, called *atom*; **(ii)** if ρ, σ are types, then $(\rho \to \sigma)$ is a type, called *composite type*.

When the atoms set is not relevant or there is no ambiguity, instead $\mathtt{Type}(\mathbb{A})$, it would be used simply Type. Also, for brevity, composite types $(\rho \to \sigma)$ are denoted simply as $\rho\sigma$.

A function $\mathbf{t} : \mathtt{Type}(\mathbb{A}) \to \mathtt{Type}(\mathbb{A})$ is called *type substitution* if and only if $\mathbf{t}$ satisfies:

(i) there exist only finitely many atoms $a \in \mathbb{A}$ such that $\mathbf{t}a \neq a$; and

(ii) $\mathbf{t}(\rho \to \sigma) = (\mathbf{t}\rho \to \mathbf{t}\sigma)$ for all composite type $(\rho \to \sigma)$.

The notion of typed graphs defined below and exemplified in Figure 6 relies on L-functions and on the notions of abs- and app-nodes.

Definition 5.2 [Typed Graph] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a $\mathtt{CL}^g$ - graph, $\tau \in \partial\mathcal{T}$ a free-edge and $E : \mathcal{T} \to \mathtt{Type}$ be a function from edges to types. The pair $\langle \mathcal{G}, E \rangle$ is called a *typed graph* w.r.t. τ if and only if for $\tau_i := L_v(i)$, where $0 \leq i < \mathtt{deg}_{\mathcal{T}}(v)$, the following conditions hold.

- For each vertex $v \in \mathcal{V}$ such that $\mathtt{deg}_{\mathcal{T}}(v) = 3$: **(i)** if v is a abs-node then $E(\tau_0) = E(\tau_2) \to E(\tau_1)$ and, **(ii)** if v is a app-node then $E(\tau_1) = E(\tau_2) \to E(\tau_0)$.
- For each vertex $v \in \mathcal{V}$ such that $\mathtt{deg}_{\mathcal{T}}(v) = 2$, it holds $E(\tau_0) = \rho \to E(\tau_1)$, where ρ is an arbitrary type.

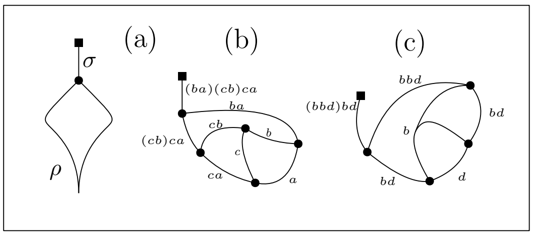


Fig. 6. (a) is not typable; (b), (c) are typed graphs

Finally, a (principal type) PTG-algorithm (Algorithm 1) is introduced that assigns types to $\mathtt{CL}^g$-graphs: it takes as input a $\mathtt{CL}^g$-graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$, a free-edge $\tau \in \partial\mathcal{T}$ and a partial function $E : \mathcal{T} \to \mathtt{Type}$ and associates to each edge in $\mathcal{T}$ a type. The algorithm works as Hindley's simple typing algorithm, but assigning new type variables and composite types for each edge of the graph. Type assignment should satisfy abstraction type requirements in abs-nodes of the graph, as well as application requirements in app-nodes of the graph, which is guaranteed by resolving the associated first-order unification problems.

Algorithm 1 Principal Type for Graphs - PTG$(\mathcal{T}, \mathcal{V}, \tau, E)$

Input: A $\mathtt{CL}^g$-graph $\langle \mathcal{V}, \mathcal{T}\rangle$, a free-edge $\tau \in \partial\mathcal{T}$, E is a partial function from edges to types.
Output: The partial function E.
Begin ▷ a is chosen always as a fresh atom
if $\langle \mathcal{V}, \mathcal{T}, \tau\rangle$ is a degenerated graph **then**
    if $E(\tau)$ is not defined **then** **return** $E \cup \{\tau \mapsto a\}$
    else **return** E
else
    if $(\mathcal{T}, \mathcal{V}, \tau)$ is a normal form w.r.t. $\Longrightarrow_{\mathcal{D}}$ **then return** FAIL
    else
        Let v be the only vertex incident to τ in $\langle \mathcal{V}, \mathcal{T}\rangle$
        if v is an abs-node **then**
            Let $(\mathcal{T}', \mathcal{V}', \tau')$ be a triplet such that $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$
            $E' \leftarrow$ PTG$(\mathcal{T}', \mathcal{V}', \tau', E)$
            if $E' = \emptyset$ or FAIL **then return** E'
            else
                if $\mathtt{deg}_{\mathcal{T}}(v) = 2$ **then**
                    return $E' = (E' \setminus \{\tau \mapsto E'(\tau)\}) \cup \{\tau \mapsto (a \to E'(L_v(1)))\}$
                else
                    return $E' = (E' \setminus \{\tau \mapsto E'(\tau)\}) \cup \{\tau \mapsto (E'(L_v(2)) \to E'(L_v(1))\}$
        else
            Let $(\mathcal{T}_1, \mathcal{V}_1, \tau_1)$ and $(\mathcal{T}_2, \mathcal{V}_2, \tau_2)$ be direct successors of $(\mathcal{T}, \mathcal{V}, \tau)$ w.r.t. $\Longrightarrow_{\mathcal{D}}$
            $E_1 \leftarrow$ PTG$(\mathcal{T}_1, \mathcal{V}_1, \tau_1, E)$
            if $E_1 = \emptyset$ or FAIL **then** **return** E_1
            else
                $E_2 \leftarrow$ PTG$(\mathcal{T}_2, \mathcal{V}_2, \tau_2, E_1)$
                if $E_2 = \emptyset$ or FAIL **then** **return** E_2
                else
                    Let γ_i be $E(L_v(i))$, $i = 1, 2$
                    if γ_1 is a composite of the form $\rho \to \sigma$ **then**
                        $\mathbf{t} \leftarrow$ Unify(ρ, γ_2) and $\gamma \leftarrow \sigma$
                    else
                        $\mathbf{t} \leftarrow$ Unify$(\gamma_1, \gamma_2 \to a)$ and $\gamma \leftarrow a$
                    if $\mathbf{t} =$ FAIL **then** **return** $\emptyset$
                    else
                        return $\mathbf{t}(E_2 \setminus \{\tau \mapsto E_2(\tau)\}) \cup \{\tau \mapsto \mathbf{t}(\gamma)\}$

Example 5.3 Figure 7 illustrates the execution of Algorithm PTG for the graph in Figure 2 (cf. Figure 6 (c)).

Theorem 5.4 (Termination of PTG) PTG *terminates for each valid instance* $(\mathcal{T}, \mathcal{V}, \tau, E)$.

Proof. Let μ be the following measure, $\mu(\mathcal{T}, \mathcal{V}, \tau, E) = |\mathcal{T}|$. It is clear that $(\mathcal{T}, \mathcal{V}, \tau, E) \Longrightarrow_{\mathtt{PTG}} (\mathcal{T}', \mathcal{V}', \tau', E')$ implies that $|\mathcal{T}'| < |\mathcal{T}|$ for $\mathcal{T}' = \mathcal{T} \setminus \{\tau\}$. Then, $\mu(\mathcal{T}, \mathcal{V}, \tau, E) < \mu(\mathcal{T}', \mathcal{V}', \tau', E')$ and the result follows. □

In the proof below the relation $\Longrightarrow_{\mathtt{PTG}}$ is used in the following sense: $(\mathcal{T}, \mathcal{V}, \tau, E) \Longrightarrow_{\mathtt{PTG}} (\mathcal{T}', \mathcal{V}', \tau', E')$ means that to compute PTG$(\mathcal{T}, \mathcal{V}, \tau, E)$ one needs to compute PTG$(\mathcal{T}', \mathcal{V}', \tau', E')$. Soundness and completeness are obtained by induction over $|\mathcal{T}'|$ using some auxiliary lemmas.

Theorem 5.5 (Soundness and Completeness of PTG) *Let* $\langle \mathcal{V}, \mathcal{T}\rangle$ *be a* $\mathtt{CL}^g$*-graph,* $\tau \in \partial\mathcal{T}$ *a free-edge,* $T : \mathcal{T} \mapsto$ Type *a partial function*

(i) *(Soundness) If* $E =$ PTG$(\mathcal{T}, \mathcal{V}, \tau, T) \neq \emptyset$ *and* $E \neq$ FAIL *then* $(\langle \mathcal{V}, \mathcal{T}\rangle, E)$ *is a typed* $\mathtt{CL}^g$*-graph w.r.t.* τ.

(ii) *(Completeness) If* $E =$ PTG$(\mathcal{T}, \mathcal{V}, \tau, \emptyset) \neq$ FAIL *and* $(\langle \mathcal{V}, \mathcal{T}\rangle, T)$ *is a typed graph w.r.t.* τ *then there exists a type substitution* $\mathbf{t}$ *such that* $T(\tau) = \mathbf{t}(E(\tau))$.

6 Conclusion

A new notion of topological graphs which contain free edges and whips has been introduced: the former corresponds to free variables whereas the latter to multiple occurrences of the same variable in λ-terms. Topological graphs whose vertices have degree 2 or 3 and contains at least one free edge are called $\mathtt{CL}^g$-graphs. When $\mathtt{CL}^g$-graphs does not have whips they called BCK-graphs, BCK-graphs whose vertices (different from root) have degree three are called BCI-graphs. It is introduced a notion of graph decomposition that checks if a topological graph is an ω-graph: $\mathtt{CL}^g$-graphs whose normal forms w.r.t. the graph decomposition relation are degenerated graphs. In addition, a sound and complete calculus to generate ω-graphs was proposed. To conclude, a sound and complete principal typing algorithm on $\mathtt{CL}^g$-graphs was given. As future work, it is planned to explore graph transformations via the proposed graph rewriting rules.

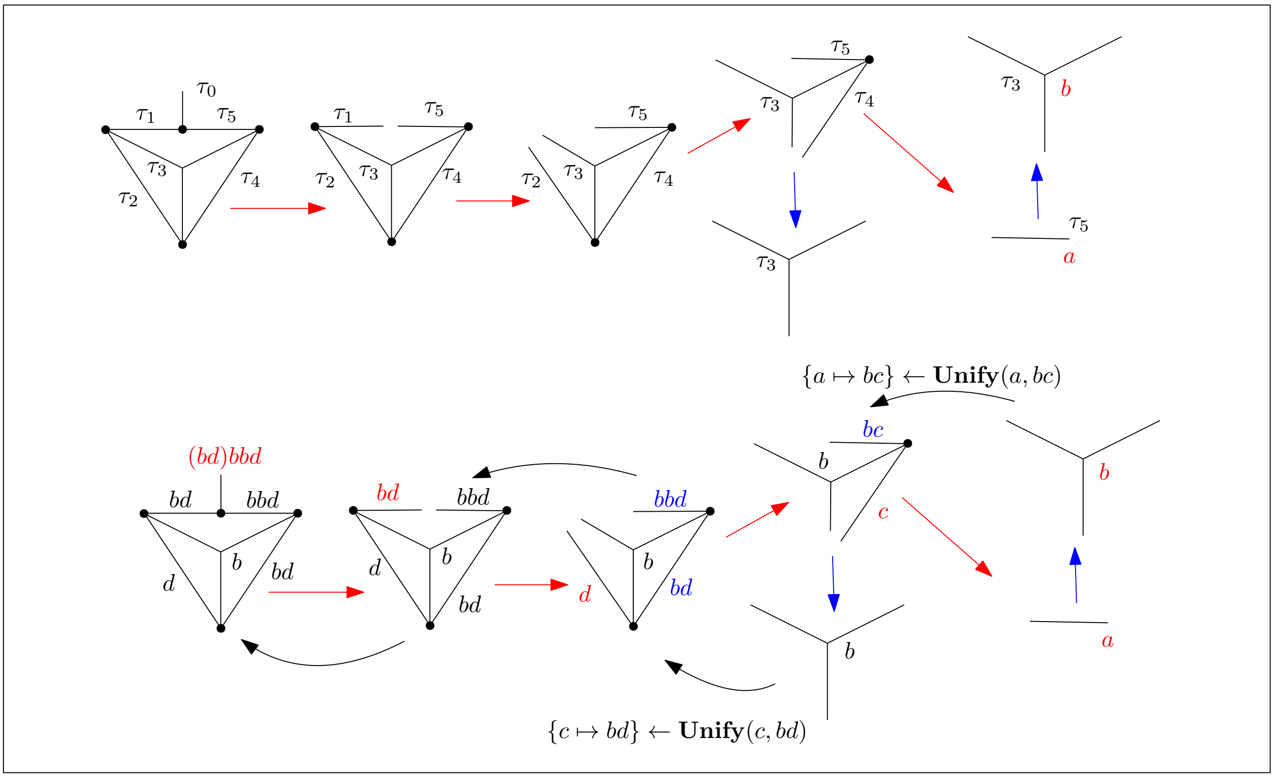


Fig. 7. PTG: Principal Type Algorithm. The figure at the top illustrates the process of decomposition; when a degenerated topological graphs are reached, a new type variable is assigned to them. The figure must be read from the left to the right, following the oriented arrows and respecting the priority of colors, RED, BLUE, BLACK. The figure at the bottom continues the one above and must be read from right to left, following the oriented arrows with the same priority. Each red letter in a degenerated topological graph is a new assignment, and if it is in a non-degenerated graph, it is the result of an application or abstraction, depending whether it is connected to an `app`-node or an `abs`-node.Each blue letter is the result of applying the unification algorithm to all assigned topological edges in this instance.

References

[Bar12] H.P. Barendregt. *The Lambda Calculus, its Syntax and Semantics.* Number 40 in Studies in Logic. College Publications, 2012.

[BGJ13] O. Bodini, D. Gardy, and A. Jacquot. Asymptotics and random sampling for BCI and BCK lambda terms. *Theor. Comput. Sci.*, 502:227–238, 2013.

[Hin97] J. R. Hindley. *Basic Simple Type Theory.* Cambridge University Press, 1997.

[HS86] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and λ-Calculus.* Number 1 in London Mathematical Society Student Texts. Cambridge University Press, 1986.

[LZ04] S. K. Lando and A. K Zvonkin. *Graphs on Surfaces and their Applications.* Springer, 2004.

[Mun15] J. R. Munkres. *Topology.* Pearson, 2nd edition, 2015.

[Zei16] N. Zeilberger. Linear lambda terms as invariants of rooted trivalent maps. *J. Funct. Program.*, 26:e21, 2016.

[Zei18] N. Zeilberger. A theory of linear typings as flows on 3-valent graphs. In *Proc. of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, pages 919–928. ACM, 2018.

[ZG15] N. Zeilberger and A. Giorgetti. A correspondence between rooted planar maps and normal planar lambda terms. *Logical Methods in Computer Science*, 11(3), 2015.

A General Topology: basic concepts

Definition A.1 [Topological Space] Let M be a non-empty set and $\tau \subseteq \mathcal{P}(M)$, where $\mathcal{P}(M)$ is the power set of M. We say that (M, τ) is a *topological space* iff τ have the following properties:

(i) $\emptyset \in \tau$;

(ii) if $\{\mathcal{U}_\lambda\}_{\lambda \in L} \subseteq \tau$, then $\bigcup \mathcal{U}_\lambda \in \tau$

(iii) given $\mathcal{U}_{\lambda_1}, \ldots, \mathcal{U}_{\lambda_n} \in \tau$, the finite intersection $\bigcap_{i=1}^{n} \mathcal{U}_{\lambda_i} \in \tau$.

The collection τ is called a *topology* for M. When there is no confusion, τ is omitted and M is called a topological space.

If M is a topological space with topology τ, we say that a subset $\mathcal{U}$ of M is an *open set* of M if $\mathcal{U}$ belongs to the collection τ. Therefore, a topological space is a set M together with a collection of subsets (open sets) of M such that $\emptyset$ and M are both open, arbitrary unions and finite intersections of open sets are open.

Example A.2 The collection of all open intervals $(a, b) = \{x \mid a < x < b\}$, in real line $\mathbb{R}$, is called the *standard topology* for $\mathbb{R}$. The product of this topology with itself, i.e., the collection of all products $(a, b) \times (c, d)$ of open intervals in $\mathbb{R}$, is called the *standard topology* on $\mathbb{R} \times \mathbb{R} = \mathbb{R}^2$.

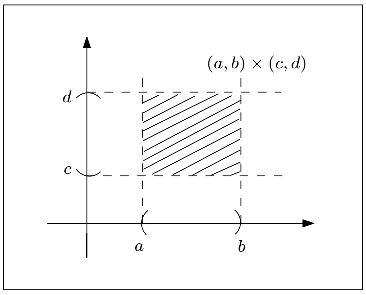


Fig. A.1. Standard Topology for $\mathbb{R}^2$

Definition A.3 Let M be a topological space with topology τ. If Y is a subset of M, the collection

$$\tau_Y = \{Y \cap \mathcal{U} \mid \mathcal{U} \in \tau\}$$

is a topology on Y, called *subspace topology.* With this topology, Y is called a *subspace* of X; its open sets consist of all intersection of open sets of X with Y.

Consider the subset $Y = [0, 1]$ of the real line $\mathbb{R}$. The subspace topology has as basis all sets of the form $(a, b) \cap Y$, where (a, b) is an open interval in $\mathbb{R}$.

A collection $\mathcal{A}$ of subsets of a space M is said to *cover* M, if the union of elements of $\mathcal{A}$ is equal to M. A space M is said to be *compact* if every open covering $\mathcal{A}$ of M contains a finite subcollection that also covers M.

Definition A.4 [Disconnected Space] Let (M, τ) be a topological space. (M, τ) is a *disconnected space* iff there exist $\mathcal{U}$ and $\mathcal{U}' \in \tau$ such that $M = \mathcal{U} \cup \mathcal{U}'$ and $\mathcal{U} \cap \mathcal{U}' = \emptyset$; otherwise, (M, τ) is called a *connected space.*

Example A.5 The interval $[0, 1] \subseteq \mathbb{R}$ is a connected topological space considering the restriction to $[0, 1]$ of the standard topology for $\mathbb{R}$. Similarly, the set $S^1 := \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$ is a connected topological space considering the standard topology for $\mathbb{R}^2$.

Let A be a subset of a topological space M and x be a point of M. It is said that x is a *point of accumulation* of A if every neighborhood of x intersects A in some other point different from x itself.

Definition A.6 Let X and Y be topological spaces and $f : X \to Y$ be a bijection. If both f and its inverse $f^{-1} : Y \to X$ are continuous, then f is called a *homeomorphism.*

B Examples of the decomposition process

Example B.1 [ω-graphs] Decompositions of two CL^g-graphs are presented in the Figure B.1. Both them are ω-graphs since every normal form is a degenerated graph.

The first ω-graph is a closed trivalent one, as described by Zeilberger [Zei16], and the rule $\Longrightarrow_{\mathcal{D}}$ decomposes it such as it is done in [Zei16]. Notice that in CL^g-graphs without n-`whip`s, it is unnecessary the labeling of all edges because all normal forms are distinct.

The second ω-graph is not a closed trivalent map, as defined in [Zei16], because there is an n-`whip` in its center, labeled as τ_4. In this graph all edges and `whip`s were labeled in order to highlight repetitions in the decomposition, specifically after the first application of rule `D-app` when τ_4 becomes a degenerated graph. The red label points to the current orientation edge.

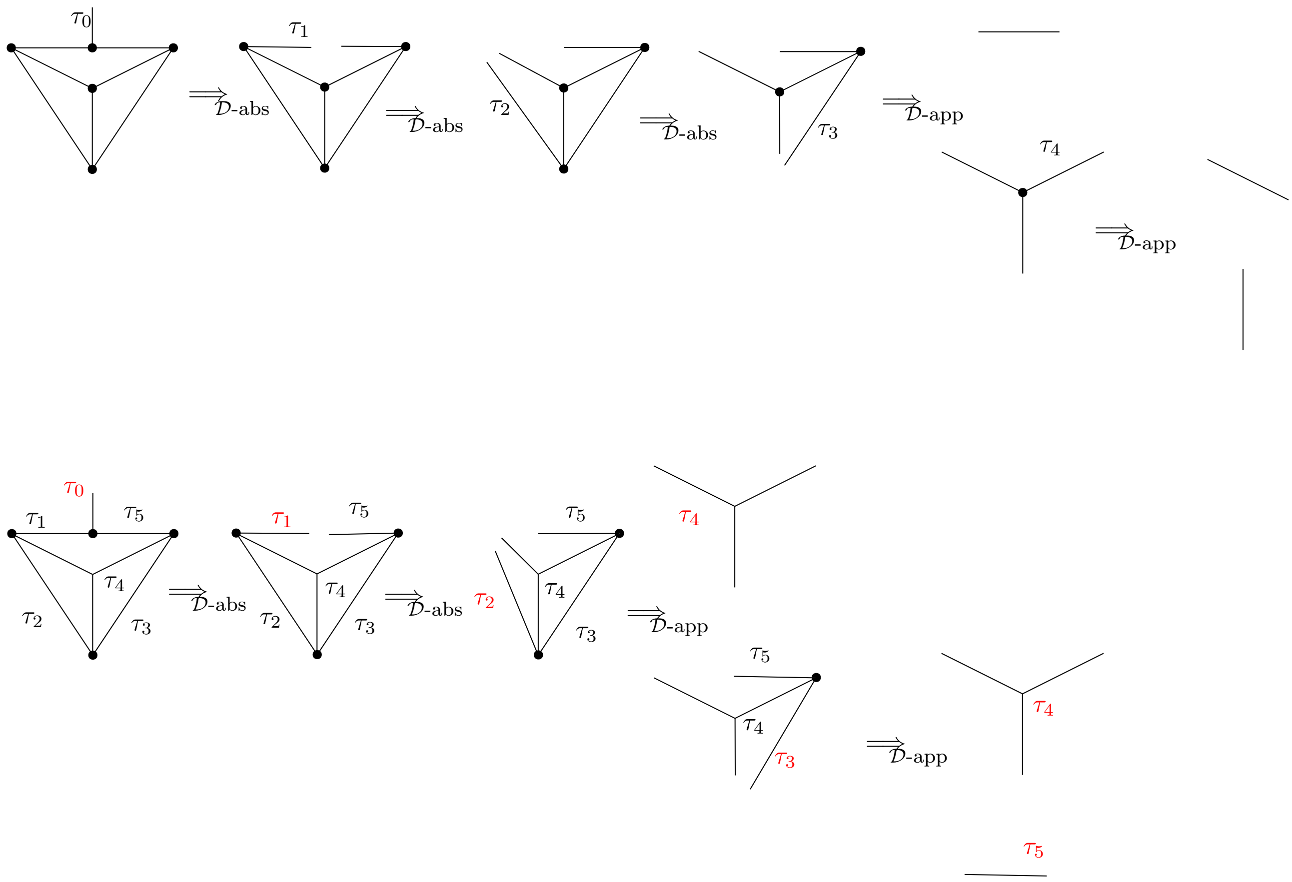


Fig. B.1. First CL^g-graph is a `BCI`-graph and the second is a proper ω-graph, i.e., with a `whip`

Example B.2 [Detailed Decomposition] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be the CL^g-graph in the Figure B.2, where $\mathcal{V} = \{v_0, v_1, v_2, v_3\}$ and $\mathcal{T} = \{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$.

By exhaustive application of the decomposition over $\mathcal{G}$ until all possible normal forms are obtained, it can be straightforward decided if $\mathcal{G}$ is an ω-graph. The schema of this decomposition is given in the Figure B.3.

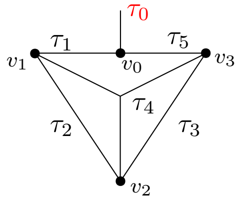


Fig. B.2. ω-graph that represents the elimination of redundant hypothesis in Implicational Intuitionist Logic

$$(\mathcal{T}, \mathcal{V}, \tau_0)$$
$$\Downarrow \text{D-abs}$$
$$(\mathcal{T}_1, \mathcal{V}_1, \tau_1)$$
$$\Downarrow \text{D-abs}$$
$$(\mathcal{T}_2, \mathcal{V}_2, \tau_2)$$
$$\Downarrow \text{D-app}$$
$$(\mathcal{T}_{21}, \mathcal{V}_{21}, \tau_3) \quad (\{\tau_4\}, \emptyset, \tau_4)$$
$$\Downarrow \text{D-app}$$
$$(\{\tau_5\}, \emptyset, \tau_5) \quad (\{\tau_4\}, \emptyset, \tau_4)$$

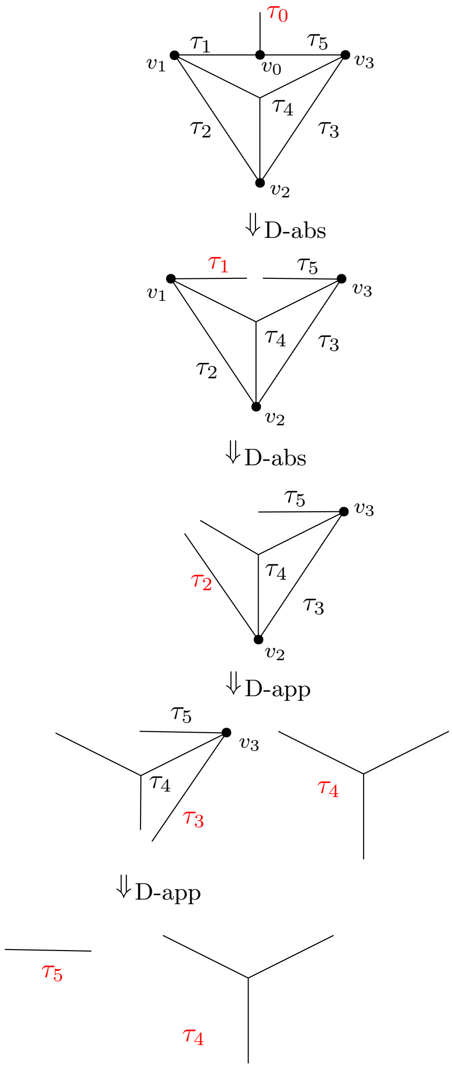


Fig. B.3. Schema of the Decomposition Process

C Proofs of Section 3

Theorem 3.9 (*D*-**consistency**) *Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a non degenerated CL^g-graph and $\tau \in \partial \mathcal{T}$ a free edge. If $\langle \mathcal{T}, \mathcal{V}, \tau \rangle \Longrightarrow_D \langle \mathcal{T}', \mathcal{V}', \tau' \rangle$ then $\tau' \in \mathcal{T}'$ and $\mathcal{G}' = \langle \mathcal{T}', \mathcal{V}' \rangle$ is a CL^g-graph or degenerated graph.*

Proof. Suppose that $\langle \mathcal{T}, \mathcal{V}, \tau \rangle \Longrightarrow_D \langle \mathcal{T}', \mathcal{V}', \tau' \rangle$ such that $\langle \mathcal{T}', \mathcal{V}' \rangle$ is a non-degenerated graph. We proceed by analysing the rule applied in this decomposition step.

(i) The rule is (`D-abs`):

(a) $\mathtt{deg}_{\mathcal{T}}(v) = 2$

Then $\mathcal{T}' = \mathcal{T} \setminus \{\tau\}, \mathcal{V}' = \mathcal{V} \setminus \Delta\tau$ and $L_v(1) = \tau'$, implies that, $\mathtt{Inc}(v) = \{\{\tau, \tau'\}\}$ and $v \in \Delta\tau'$.

- if $\mathcal{V}' = \emptyset$ then $\mathcal{G}'$ is a degenerated graph and the result follows.
- if $\mathcal{V}' \neq \emptyset$, then there exists $u \in \mathcal{V}'$ and $u \neq v$ such that $u \in \Delta\tau'$. Since $\mathcal{G}$ is a CL^g-graph, one has $\mathtt{deg}_{\mathcal{G}}(u) = 2$ or 3, therefore $|\top'| > 1$. Notice that in $\mathcal{G}'$ the vertices incident to τ' are $\{u\}$, i.e., $\Delta\tau' \cap \mathcal{V}' = \{u\}$, therefore, $\tau' \in \partial\mathcal{T}'$ is a free edge, so the result follows from Lemma 3.4.

(b) $\mathtt{deg}_{\mathcal{T}}(v) = 3$

- $L_v(1), L_v(2) \notin \partial\mathcal{T}$ and $\mathcal{G}'$ is connected topological graph.
  Note that $\tau' = L_v(1) \notin \partial\mathcal{T}$ and because a decomposition step applies, τ' cannot be a bound `whip` $\mathcal{G}$, then τ' is an edge in $\mathcal{G}$, and $|\Delta\tau'| = 2$. Therefore, $|\Delta\tau' \cap \mathcal{V}'| = 1$ and $\tau' \in \partial\mathcal{T}'$. Let $w \in \mathcal{V}'$ be a vertex in $\mathcal{G}'$, since $w \neq v$ it follows that $w \notin \Delta\tau$ and $\mathcal{T}(w) \subseteq \mathcal{T}'$. Thus, $\mathcal{T}(w) = \mathcal{T}'(w)$ and $\mathtt{deg}_{\mathcal{T}'}(w) = \mathtt{deg}_{\mathcal{T}}(w) = 2$ or 3. The result follows.
- $L_v(1)$ is a loop.
  This case is trivial: $\mathcal{V}' = \mathcal{V} \setminus \Delta\tau = \emptyset$ and $\mathcal{G}'$ is a degenerated graph.

(ii) The rule is (`D-app`):

In this case $\mathtt{deg}_{\mathcal{T}}(v) = 3$ and there exist topological graphs $\mathcal{G}_i = \langle \mathcal{T}_i, \mathcal{V}_i \rangle$, for $i = 1, 2$. Notice that if $\mathcal{V}_1 = \mathcal{V}_2 = \emptyset$ then both graphs $\mathcal{G}_i$ are degenerated and the result follows.

Suppose $\mathcal{V}_i \neq \emptyset$ for some $i = 1, 2$. The remaining of the proof consists in showing that $\mathcal{G}_i$ is a CL^g-graph.

(a) $\mathcal{G}_i = \langle \mathcal{T}_i, \mathcal{V}_i \rangle$ is connected:

Suppose, without loss of generality, that $i = 1$ and let $w, w' \in \mathcal{V}_1$ be distinct vertices in $\mathcal{G}_1$. Since $\mathcal{G}$ is connected, there exists a sequence of pairwise distinct adjacent edges $\tau_1, \ldots, \tau_m \in \mathcal{T}$ connecting w and w' in $\mathcal{G}$. Suppose, by contradiction, that there exists an index j such that $\tau_j \notin \mathcal{T}_1$, take the smallest j with this property. Notice that $\tau_j \notin \mathcal{T}_1$ implies that $\tau_j \cap \mathcal{V}_1 = \emptyset$. Since τ_{j-1} and τ_j are adjacent, it follows that $\Delta\tau_{j-1} \cap \Delta\tau_j \neq \emptyset$, besides $j-1 < j$ implies, by minimality of j, that $\tau_{j-1} \in \mathcal{T}_1$. Since τ_{j-1} and τ_j are edges in $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ it follows that $\tau_{j-1}, \tau_j \neq \tau$ (for τ is a half edge). Besides, $\tau_{j-1} \cap \mathcal{V}_2 = \emptyset$, otherwise, there would be a connection between vertices from $\mathcal{V}_1$ and $\mathcal{V}_2$ contradicting the condition in rule (`app`). Therefore, $\{v\} = \Delta\tau_{j-1} \cap \Delta\tau_j$, then $\tau_{j-1} = L_v(1)$ and $\tau_j = L_v(2) \in \mathcal{T}_2$. However, in this case, there exists $v' \in \mathcal{V}_2 \cap \Delta\tau_j$ connected to $w' \in \mathcal{V}_1$ via path $\tau_{j+1}, \ldots, \tau_m$, which is a contradiction. Thus, $\mathcal{G}_1$ is connected.

(b) $\mathtt{deg}_{\mathcal{G}_i}(w) = 2$ or 3 for every $w \in \mathcal{V}_i$.

For all $w \in \mathcal{V}_i$, it follows that $w \neq v$ and $w \in \mathcal{V}$ and, by hypothesis, $\mathtt{deg}_{\mathcal{G}}(w) = 2$ or 3, and the result follows trivially.

(c) $L_v(i) \in \partial\mathcal{T}_i$, $i = 1, 2$.

Since $\mathcal{V}_i \neq \emptyset$ for some i, suppose, w.l.o.g., that $i = 1$ and let $\tau_1 := L_v(1)$ be the edge in $\mathcal{T}_1$ assigned via L_v. Notice that $|\Delta\tau_1| = 2$. It is straightforward to check that $|\Delta\tau_1 \cap \mathcal{V}_1| = 1$ since v was removed from $\mathcal{G}_1$ and $\tau_1 \in \partial\mathcal{T}_1$.

□

Lemma C.1 **($(\lambda, @)$-labelling consistency)** *Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a non degenerated CL^g-graph and $\tau \in \partial\mathcal{T}$ a free edge. If $\Delta\tau = \{v\}$ and v is an $(@)$-node then v is not a (λ)-node.*

Proof. Suppose that v is an $(@)$-node, then $\mathtt{deg}_{\mathcal{G}}(v) = 3$. Note that if $L_v(1) \in \partial\mathcal{T}$ or $L_v(2) \in \partial\mathcal{T}$ then $(\lambda)-$rule cannot be applied. Suppose that $L_v(1), L_v(2) \notin \partial\mathcal{T}$, then by the last requirement of $(@)$ it follows that $\mathcal{V}_1$ and $\mathcal{V}_2$ are non empty sets of variables.

Note that $L_v(1)$ cannot be a loop, otherwise, $\mathcal{V} = \{v\}$ and $\mathcal{T} = \{\tau, L_v(1)\}$, implying that $\mathcal{V}_1 = \emptyset$, which contradicts the previous step. Let $w_1 \in \mathcal{V}_1$ and $w_2 \in \mathcal{V}_2$ be vertices then there is no path in $\langle \mathcal{T} \setminus \{\tau\}, \mathcal{V} \setminus \Delta\tau \rangle$ connecting w_1 and w_2 therefore $\langle \mathcal{T} \setminus \{\tau\}, \mathcal{V} \setminus \Delta\tau \rangle$ is not a connected topological CL^g-graph. Therefore, the (`abs`)-rule cannot be applied and v is not a (λ)-node. □

C.1 Graph Skeleton

This subsection provides technical results regarding the notion of graph `Skeleton` that are used to prove completeness of the typing algorithm for graphs (`PTG`) introduced in Section 5.

Definition C.2 [Skeleton] Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{T} \rangle$ be a CL^g-graph and $\tau \in \partial\mathcal{T}$ a free edge, such that all normal forms of $\langle \mathcal{T}, \mathcal{V}, \tau \rangle$ with relation to $\Longrightarrow_{\mathcal{D}}$ are degenerated graphs. Then the set $\mathtt{Skel}_{\mathcal{T}}(\tau)$ is defined inductively as follows:

(i) if $\mathcal{G}$ is a degenerated graph then $\mathtt{Skel}_{\mathcal{T}}(\tau) := \emptyset$,

(ii) if $\langle \mathcal{T}, \mathcal{V}, \tau \rangle \Longrightarrow_{\mathcal{D}} \langle \mathcal{T}', \mathcal{V}', \tau' \rangle$ via rule $(\mathtt{D}-\mathtt{abs})$ then $\mathtt{Skel}_{\mathcal{T}}(\tau) := \{\tau\} \cup \mathtt{Skel}_{\mathcal{T}'}(\tau')$

(iii) if $\langle \mathcal{T}, \mathcal{V}, \tau \rangle \Longrightarrow_{\mathcal{D}} \langle \mathcal{T}_i, \mathcal{V}_i, \tau_i \rangle, i = 1, 2$ via $(\mathtt{D}-\mathtt{app})$ then $\mathtt{Skel}_{\mathcal{T}}(\tau) := \{\tau\} \cup \mathtt{Skel}_{\mathcal{T}_1}(\tau_1) \cup \mathtt{Skel}_{\mathcal{T}_2}(\tau_2)$

The $\texttt{Skel}_{\mathcal{T}}(\tau_0)$ of the graph in Figure 2 consists of the edges in red.

Lemma C.3 *If* $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow^*_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$ *then* $\texttt{Skel}_{\mathcal{T}'}(\tau') \subseteq \texttt{Skel}_{\mathcal{T}}(\tau')$

Proof. The proof is by induction on the number of steps in $\Longrightarrow^*_{\mathcal{D}}$.
Base Case: $\langle \mathcal{T}, \mathcal{V}, \tau\rangle \Longrightarrow_{\mathcal{D}} \langle \mathcal{T}', \mathcal{V}', \tau'\rangle$.
This case follows directly from the definition of $\texttt{Skeleton}$.
Induction Step: Suppose that $\langle \mathcal{T}, \mathcal{V}, \tau\rangle \Longrightarrow_{\mathcal{D}} \langle \mathcal{T}'', \mathcal{V}'', \tau''\rangle \Longrightarrow^*_{\mathcal{D}} \langle \mathcal{T}', \mathcal{V}', \tau'\rangle$. By the base case, $\texttt{Skel}_{\mathcal{T}''}(\tau'') \subseteq \texttt{Skel}_{\mathcal{T}}(\tau)$ and by (IH) it follows that $\texttt{Skel}_{\mathcal{T}'}(\tau') \subseteq \texttt{Skel}_{\mathcal{T}''}(\tau'')$ and the result follows. ☐

Lemma C.4 *Let* $\mathcal{G} = \langle \mathcal{V}, \mathcal{T}\rangle$ *be a* CL^g*-graph.*

(i) *If* $\tau' \in \texttt{Skel}_{\mathcal{T}}(\tau)$ *then there exists a* CL^g*-graph* $\mathcal{G}' = \langle \mathcal{T}', \mathcal{V}'\rangle$ *such that* $\tau' \in \partial\mathcal{T}'$ *and* $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow^*_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$.

(ii) *If* $\tau' \in \texttt{Skel}_{\mathcal{V}}(\mathcal{T}) \setminus \{\tau\}$ *then* τ' *is an edge in* $\mathcal{G}$.

(iii) *If* $\langle \mathcal{T}, \mathcal{V}, \tau\rangle \Longrightarrow_{\mathcal{D}} \langle \mathcal{T}_i, \mathcal{V}_i, \tau_i\rangle$ *via rule* $(\texttt{D}-\texttt{app})$ *then* $\texttt{Skel}_{\mathcal{T}_1}(\tau_1) \cap \texttt{Skel}_{\mathcal{T}_2}(\tau_2) = \emptyset$.

Proof.

(i) The proof is by induction on the length of derivation $\Longrightarrow^*_{\mathcal{D}}$.
Note that $\tau' \in \texttt{Skel}_{\mathcal{T}}(\tau)$ then it is not empty, implies that $\langle \mathcal{V}, \mathcal{T}\rangle$ it is not a degenerated topological graph and by hypothesis all normal forms are degenerated graphs, so $\Longrightarrow_{\mathcal{D}}$ can be applied on $(\mathcal{T}, \mathcal{V}, \tau)$ resulting in two possible cases, $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}_1, \mathcal{V}_1, \tau_1)$ by $\mathcal{D}$-abs or $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}_1, \mathcal{V}_1, \tau_1), (\mathcal{T}_2, \mathcal{V}_2, \tau_2)$ by $\mathcal{D}$-app. Therefore $\texttt{Skel}_{\mathcal{T}}(\tau) = \{\tau\} \cup \texttt{Skel}_{\mathcal{T}_1}(\tau_1)$ or $\{\tau\} \cup \texttt{Skel}_{\mathcal{T}_1}(\tau_1) \cup \texttt{Skel}_{\mathcal{T}_2}(\tau_2)$, following that $\tau' = \tau$ or $\tau' \in \texttt{Skel}_{\mathcal{T}_i}(\tau_i)$ for some $i \in \{1, 2\}$. If $\tau' = \tau$ then the result comes straight forward, suppose that $\tau' \neq \tau$, implies that $\tau' \in \texttt{Skel}_{\mathcal{T}_i}(\tau_i)$, so by induction on $\Longrightarrow_{\mathcal{D}}$ follows that exist $\langle \mathcal{T}', \mathcal{V}'\rangle$ a CL^g-graph such that $\tau' \in \partial\mathcal{T}'$ is a free edge and $(\mathcal{T}_i, \mathcal{V}_i, \tau_i) \Longrightarrow^*_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$, and the result follows because $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}_i, \mathcal{V}_i, \tau_i)$.

(ii) By hypothesis, $\tau' \in \texttt{Skel}_{\mathcal{T}}(\tau) \setminus \{\tau\}$ and by item (i) there exists a CL^g-graph $\langle \mathcal{T}', \mathcal{V}'\rangle$ such that $\tau' \in \mathcal{T}'$ and $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow^n_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$ for some $n \geq 1$ because $\tau' \neq \tau$. The proof is by induction on n.
Base Case: Suppose $n = 1$ then $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$ and follows directly by definition of $\Longrightarrow_{\mathcal{D}}$ because $\tau' \in \texttt{Skel}_{\mathcal{T}}(\tau) = \{\tau\} \cup \texttt{Skel}_{\mathcal{T}'}(\tau')$, implying that $\tau' \in \texttt{Skel}_{\mathcal{T}'}(\tau')$ and for this $\langle \mathcal{T}', \mathcal{V}'\rangle$ it is not a degenerated topological graph.
Induction Step: Suppose that $n > 1$, so $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}'', \mathcal{V}'', \tau'') \Longrightarrow^{n-1}_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$ and then by (IH) τ' is an edge in $\langle \mathcal{T}'', \mathcal{V}''\rangle$, but $\mathcal{T}'' \subseteq \mathcal{T}$ and $\mathcal{V}'' \subseteq \mathcal{V}$ so implies that τ' is also an edge in $\langle \mathcal{V}, \mathcal{T}\rangle$.

(iii) If $\langle \mathcal{T}_i, \mathcal{V}_i\rangle$ is a degenerated topological graph for some $i \in \{1, 2\}$ then $\texttt{Skel}_{\mathcal{T}_i}(\tau_i) = \emptyset$ and the result follows directly.
Suppose that for each $i = 1, 2$, $\langle \mathcal{T}_i, \mathcal{V}_i\rangle$ it is not a degenerated topological graph. Then $\texttt{Skel}_{\mathcal{T}_i}(\tau_i) = \{\tau_i\} \cup R_i$ for $i = 1, 2$, then supposing by contradiction that $R_1 \cap R_2 \neq \emptyset$, so exist a $\hat{\tau} \in R_1 \cap R_2 \subseteq \texttt{Skel}_{\mathcal{T}_i}(\tau_i)$ for $i = 1, 2$, so $\hat{\tau} \in \mathcal{T}_1 \cap \mathcal{T}_2$ and by item (ii) $\hat{\tau}$ is an edge in $\mathcal{T}_i$ for $i = 1, 2$, contradicts that $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$, therefore $R_1 \cap R_2 = \emptyset$ and because $\tau_1 \neq \tau_2$ and $\tau_{3-i} \notin \mathcal{T}_i \supseteq R_i$ for $i = 1, 2$ the result follows.

☐

D Proofs of Section 4

Definition 4.4 [isomorphism] Let $\langle \mathcal{T}, \mathcal{V}, r\rangle$ and $\langle \mathcal{T}', \mathcal{V}', r'\rangle$ be ω-graphs and $\varphi : \mathcal{T} \to \mathcal{T}'$ and $\psi : \mathcal{V} \to \mathcal{V}'$ mappings between edges and vertices, respectively. We say that the 2-map cell (φ, ψ) is an isomorphism iff the following properties hold:

(i) φ and ψ are bijections and $\psi(r) = r'$;

(ii) $\varphi(\tau)$ is homeomorphic to τ, for all $\tau \in \mathcal{T}$;

(iii) $\psi(\Delta\tau) = \Delta\varphi(\tau)$, for all $\tau \in \mathcal{T}$;

(iv) $\varphi(\texttt{Inc}(v)) = \texttt{Inc}(\psi(v))$, for all $v \in \mathcal{V}$. Besides, $\varphi(\texttt{Inc}(v))$ and $\texttt{Inc}(\psi(v))$ have the same clockwise orientation.

Theorem 4.5 (Correctness) *Let* $\mathcal{G}(r) = \langle \mathcal{T}, \mathcal{V}, r\rangle$ *be a rooted topological graph.*

(i) *(Soundness) If* $(\partial\mathcal{T}, \mathcal{V}, r) \vdash \mathcal{G}_{\mathcal{T}}$ *then* $\mathcal{G}(r)$ *is an* ω*-graph. Moreover, if only rules (ax), (app) and (abs) rules are then* $\mathcal{G}(r)$ *is a* BCI*-graph and, if the* ω*-rules are not used, then* $\mathcal{G}(r)$ *is a* BCK*-graph.*

(ii) *(Completeness) If* $\mathcal{G}(r)$ *is an* ω*-graph, then there exists an* ω*-graph* $\langle \mathcal{T}', \mathcal{V}', r'\rangle$ *isomorphic to* $\mathcal{G}(r)$ *such that* $(\partial\mathcal{T}', \mathcal{V}', r') \vdash \mathcal{G}_{\mathcal{T}'}$.

Proof.

(i) The proof is by induction on the derivation ∇ of $\langle \partial\mathcal{T}, \mathcal{V}, r\rangle \vdash \mathcal{G}_{\mathcal{T}}$.
Base Case. ∇ consists of an application of rule (ax).
Therefore, $\mathcal{T} = \partial\mathcal{T} = \{\tau\}$ where τ is a rooted free edge, then clearly by definition $(\mathcal{T}, \{r\}, r)$ is an ω-graph, and more, it is a `BCI`, `BCK`-graph.
Inductive Step: the proof is by analysing the last rule applied in ∇.

- **The rule is** (app).
Then, by definition of the rule one has $\partial\mathcal{T} = \partial\mathcal{T}' \cup \partial\mathcal{T}^{''}$, $\mathcal{V} = \mathcal{V}' \cup (\mathcal{V}'' \setminus \{r''\} \cup \{r\})$ and there exist derivations ∇' and ∇'' of $\langle \partial\mathcal{T}', \mathcal{V}', r'\rangle \vdash \mathcal{G}_{\mathcal{T}'}(r')$ and $\langle \partial\mathcal{T}'', \mathcal{V}'', r''\rangle \vdash \mathcal{G}_{\mathcal{T}''}(r'')$, respectively. By (IH), $\mathcal{G}_{\mathcal{T}'}(r')$ and $\mathcal{G}_{\mathcal{T}''}(r'')$ are ω-graphs. Notice that $\mathcal{G}_{\mathcal{T}}(r)$ is obtained by colliding the roots of $\mathcal{G}_{\mathcal{T}'}(r')$ and $\mathcal{G}_{\mathcal{T}''}(r'')$ into vertex r' and maintaining all other connections and adding a new root vertex r and a new edge, say τ_r, connecting r and r'. So $\mathtt{deg}_{\mathcal{T}}(r') = 3$ and r' connects all other vertices in $\mathcal{V}'$ and $\mathcal{V}''$. Therefore, $\mathtt{deg}_{\mathcal{T}}(v) = 2$ or 3 for all $v \in \mathcal{V} \setminus \{r\}$ and $\mathcal{G}_{\mathcal{T}}$ is connected.
It remains to show that all normal forms of $\langle \mathcal{T}, \mathcal{V}, \tau_r\rangle$ w.r.t. $\Longrightarrow_D$ are degenerated graphs. First, notice that no bound $\tilde{\mathtt{whip}}$ is added. Also, since $\mathtt{deg}_{\mathcal{T}}(r') = 3$ one can apply rules $(\mathtt{D}-\mathtt{app})$ or $(\mathtt{D}-\mathtt{abs})$. However, $\langle \mathcal{T} \setminus \{\tau_r\}, \mathcal{V} \setminus r', L_v(1)\rangle$ is disconnected, therefore, the rule applied should be $(\mathtt{D}-\mathtt{app})$ and $\langle \mathcal{T}, \mathcal{V}, \tau_r\rangle \Longrightarrow_D \langle \mathcal{T}_i, \mathcal{V}_i, L_v(i)\rangle$ which corresponds to ω-graphs $\mathcal{G}_{\mathcal{T}'}$ and $\mathcal{G}_{\mathcal{T}''}$ for $i = 1, 2$, and their normal forms w.r.t. $\Longrightarrow_D$ are degenerated graphs.
- **The rule is** (abs).
So one has ∇' proof of $(\partial\mathcal{T}', \mathcal{V}', r') \vdash \mathcal{G}'_{\mathcal{T}}$ the premise of the last rule in ∇ such as in the `abs`-rules. Then, by (IH), $(\mathcal{T}', \mathcal{V}', r')$ is an ω-graph. Note that $\mathcal{G}_{\mathcal{T}}$ is obtained adding a new root r and a new edge connecting r and r'. Additionally, according to the `abs`-rule, either or not an edge τ' in the frontier is connected with r'; thus, all old vertices maintain their degrees, except r' that might reach degree either tree or two. `whips` in $\mathcal{T}$ are inherited by $\mathcal{T}'$.

The other cases are analogous. If ∇ does not use ω-rules then ∇' and ∇'' also do not use ω-rules and it follows, by (IH), that $\langle \mathcal{T}', \mathcal{V}', r'\rangle$ and $\langle \mathcal{T}'', \mathcal{V}'', r''\rangle$ are `BCK`-graphs and then $\langle \mathcal{T}, \mathcal{V}, r\rangle$ is also a `BCK`-graph, since last rule no adds a `whip`. In the case ∇ uses only rules (ax), (app) and (abs), then one can conclude that $\mathcal{G}_{\mathcal{T}}$ is a `BCI`-graph reasoning similarly.

(ii) The proof is by induction on the cardinality of $\mathcal{V}$. Let τ be the rooted edge in $\mathcal{G}(r)$, i.e., $r \in \Delta\tau$.
Induction Basis: $|\mathcal{V}| = 1$,
Then $\mathcal{V} = \{r\}$ and $\mathcal{T} = \{\tau\}$, that is, τ is a free edge. Therefore, $(\{\tau\}, \{r\}, r) \vdash \tau$ is obtained via rule (ax) and the result follows.
Inductive Step: $|\mathcal{V}| > 1$. Let $v \in \Delta\tau$ be the rooted vertex, There are two subcases:
(a) $\mathtt{deg}_{\mathcal{T}}(v) = 2$ and $\mathtt{Inc}(v) = \{\{\tau, \gamma\}\}$:
For $\mathcal{T}' = \mathcal{T} \setminus \{\tau\}$ and $\mathcal{V}' = \mathcal{V} \setminus \{r\}$, it follows that $\mathcal{G}'(v) = \langle \mathcal{T}', \mathcal{V}', v\rangle$ is an ω-graph with root v and γ is the only (free)-edge incident to v in $\mathcal{G}'$.
So, by (IH) there exists a rooted ω-graph $\langle \mathcal{T}'', \mathcal{V}'', r''\rangle$ isomorphic to $\langle \mathcal{T}', \mathcal{V}', v\rangle$ via isomorphism (ϕ, ψ) such that $(\partial\mathcal{T}'', \mathcal{V}'', r'') \vdash \mathcal{G}_{\mathcal{T}''}$. By applying rule (abs*) one obtains $(\partial\mathcal{T}'', \mathcal{V}'' \uplus \{r'\}, r') \vdash \mathcal{G}_{\mathcal{T}^*}$ with r' a new root vertex where $\mathcal{T}^* = \mathcal{T}'' \cup \{\tau_{r'}\}$ and $\tau_{r'}$ is a new edge such that $\Delta\tau_{r'} = \{r', r''\}$. Define an isomorphism $(\widehat{\varphi}, \widehat{\psi})$ as $\widehat{\psi}(r) := r'$, $\widehat{\psi}|_{\mathcal{V}''} := \psi$, $\widehat{\varphi}(\tau_{r'}) := \tau$ and $\widehat{\varphi}|_{\mathcal{T}''} := \phi$ and the result follows.
(b) $\mathtt{deg}_{\mathcal{T}}(v) = 3$ and $\mathtt{Inc}(v) = \{\{\tau, \gamma_1, \gamma_2\}\}$: this case is similar to the previous one.
There are additional subcases to be considered.
  - **case** $\gamma_1 = \gamma_2$:
Thus, γ_1 is a *loop* and $\mathcal{T} = \{\tau, \gamma_1\}, \mathcal{V} = \{r, v\}$ and $\partial\mathcal{T} = \emptyset$. So, $(\{\tau\}, \{v\}, v) \vdash \tau$ is an axiom and applying abs over τ one obtains a proof ∇ of $(\{\tau', \gamma'_1\}, \{r', v\}, r')$, where γ'_1 is a loop with v as only vertex and τ' is the rooted edge with vertices r', v, thus, the isomorphism with $\mathcal{T}$ is straightforward.
  - **case** γ_2 is a bound $n+1$-`whip`:
Let $p \in \gamma_2$ be the only point in γ_2 such that $\gamma_2 \setminus (\Delta\gamma_2 \cup \{p\})$ have $(n+1)$ connected components, we say $\mathcal{C}_{\gamma_2}$ is the set of such connected components. Let $t \in \mathcal{C}_{\gamma_2}$ be the connected component such that $t \cup \{p, v\}$ is homeomorphic to $[0, 1]$, and t' a topological space homeomorphic to $[0, 1]$ such that p is an extremity of t', t' that does not cross any topological edge in $\mathcal{T} \setminus \{\gamma_2\}$ and $\gamma''_2 := \gamma_2 \setminus (t \cup \{v, p\}) \cup t'$ is a free `whip`. Then, $\mathcal{G}_1(v) := \langle \mathcal{V} \setminus \{r\}, \mathcal{T} \setminus \{(\tau_r, \gamma_2\}) \cup \{\gamma''_2\}, v\rangle$ is an ω-graph with v as root, because removing v has the same normal forms than $\mathcal{G}(r)$.
By applying the (IH) to $\mathcal{G}_1(v)$ there is an ω-graph $\mathcal{G}_2(v_2) = \langle \mathcal{V}_2, \mathcal{T}_2, v_2\rangle$ isomorphic to $\mathcal{G}_1(v)$ such that $(\partial\mathcal{T}_2, \mathcal{V}_2, v_2) \vdash \mathcal{G}_{\mathcal{T}_2}$, with the isomorphism (φ, ψ). So, applying the rule ω-abs in $\mathcal{G}_2(v_2)$ over $\varphi(\gamma''_2)$ results in the ω-graph $\mathcal{G}'(r') = \langle \mathcal{V}_2 \cup \{r'\}, (\mathcal{T}_2 \setminus \{\varphi(\gamma''_2)\}) \cup \gamma'_2, r'\rangle$, where γ'_2 is a bounded $(n+1)$-`whip`with $\Delta\gamma'_2 = \Delta\varphi(\gamma''_2) \cup \{\psi(v)\}$.
With that, one can build the functions $\varphi' : \mathcal{T} \mapsto \mathcal{T}'$, $\psi' : \mathcal{V} \mapsto \mathcal{V}'$ such that $\varphi'(\tau_r) = \tau_{r'}$, $\varphi'(\gamma_2) = \gamma'_2$

and $\varphi'|_{\mathcal{T}\setminus\{\tau_r,\gamma_2\}} = \varphi|_{\mathcal{T}\setminus\{\tau_r,\gamma_2\}}$, $\psi'(r) = r'$ and $\psi'|_{\mathcal{V}\setminus\{r\}} = \psi|_{\mathcal{V}\setminus\{r\}}$, and then it follows that (φ', ψ') is an isomorphism.

- **case** γ_2 is an edge:
  There are two possibilities, v is an `abs`-node or an `app`-node. The case of being an `abs`-node is analogous to the previous case, only considering γ_2 an edge and using the `abs` rule. The case of being an `app`-node is also analogous to the previous case, but considering the `app` rule and ω-`app` rule.

□

Theorem 4.6 (ω-graph to λ-term) *For each ω-graph exists only one λ-term M without variable-clashes, modulo α-conversion and free variables renaming and a surjective function $f : \mathcal{T} \mapsto Subterms(M)$ such that for each $v \in \mathcal{V}\setminus\{r\}$ where $\tau := L_v(0), \tau' := L_v(1)$ and (when $\mathtt{deg}_{\mathcal{T}}(v) = 3$) $\tau'' := L_v(2)$, the following properties hold for f:*

(i) *For each $\gamma, \gamma' \in \partial\mathcal{T}$ distinct, $f(\gamma), f(\gamma')$ are distinct variables.*
(ii) *if v is an `app`-node then, $f(\tau) = (f(\tau')f(\tau''))$*
(iii) *if v is an `abs`-node and $\mathtt{deg}_{\mathcal{T}}(v) = 3$ then, $f(\tau) = \lambda f(\tau'') \cdot f(\tau')$ where $f(\tau'')$ is a variable.*
(iv) *if v is an `abs`-node and $\mathtt{deg}_{\mathcal{T}}(v) = 2$ then, $f(\tau) = \lambda x_v \cdot f(\tau')$, where x_v is a variable.*

Proof. Will be proved by induction over the $\mathbb{G}(w)$ calculus and using the theorem 4.5, so can be assumed that each ω-graph is a derivation in the $\mathbb{G}(\omega)$ calculus.
Induction Basis: Suppose that $(\partial\mathcal{T}, \mathcal{V}, r) \vdash \mathcal{G}_{\mathcal{T}}$ by the rule (ax), then $\mathcal{T} = \{\tau\}, \mathcal{V} = \{r\}$ and $\partial\mathcal{T} = \mathcal{T}$ so τ is a free-edge. Let x be a variable and make the function $f : \tau \mapsto \{x\}$ such that $f(\tau) = x$, then the result follows by vacuity because $\mathcal{V} \setminus \{r\} = \emptyset$.
Induction Step: Suppose that $(\partial\mathcal{T}, \mathcal{V}, r) \vdash \mathcal{G}_{\mathcal{T}}$ does not end with the rule (ax).

- case `app`-rule: there are two premises $(\partial\mathcal{T}', \mathcal{V}', r') \vdash \mathcal{G}'_{\mathcal{T}}$ and $(\partial\mathcal{T}'', \mathcal{V}'', r'') \vdash \mathcal{G}''_{\mathcal{T}}$ and by (IH) there are two λ-terms without variable-clash M' and M'' (with disjoint variables) and two surjective functions f' and f'' corresponding to $\langle\mathcal{T}', \mathcal{V}', r'\rangle$ and $\langle\mathcal{T}'', \mathcal{V}'', r''\rangle$, respectively, such that $FV(M') \cap FV(M'') = \emptyset$. Then $\mathcal{V} = \mathcal{V}'\dot{\cup}(\mathcal{V}'' \setminus \{r''\})\dot{\cup}\{r\}$, $\mathcal{T} = \mathcal{T}'\dot{\cup}(\mathcal{T}'' \setminus \{\tau_{r''}\})\dot{\cup}\{\gamma, \tau_r\}$, where $\Delta\tau_{r''} = \{v_{r''}, r''\}, \tau_r$ is the rooted edge connecting r' to r and γ is an edge connecting v''_r to r' So, let $f : \mathcal{T} \mapsto Subterms(M'M'')$ be a function such that $f(\tau_r) = (M'M'')$, $f(\gamma) := f''(\tau_{r''})$, for each $\tau \in \mathcal{T}', f(\tau) := f'(\tau)$ and for each $\tau \in \mathcal{T}'' \setminus \{\tau_{r''}\}, f(\tau) = f''(\tau)$. Is clear that f is a surjective function, because the union of f' with f'' except for replacing $\tau_{r''}$ by γ in the domain, but maintaining the images and including τ_r is different from the others.
  Thus, by (IH) the functions f' and f'' have the properties described before. Then, it is only necessary to consider the properties of the new vertex r', and the result follows straightforwardly for because r' is an `app`-node in the ω-graph $\langle\mathcal{V}, \mathcal{T}, r\rangle$ with $L_{r'}(0) = \tau_r, L_{r'}(1) = \tau_{r'}$ and $L_{r'}(2) = \gamma$.
- case (ω-*app*)-rule: exactly as for the `app`-rule, but the free variables of M', M'' will be renamed until each pair $(\tau', \tau'') \in \partial\mathcal{T}' \times \partial\mathcal{T}''$ such that $\tau'\tau'' \in \mathcal{T}$ has $f'(\tau') = f''(\tau'')$. Then $FV(M') \cap FV(M'') \neq \emptyset$ and for each $x \in FV(M') \cap FV(M'')$, $\tau' := f'^{-1}(x)$, $\tau'' := f''^{-1}(x)$ are replaced by $\tau'\tau''$ in $\mathcal{T}$ and $f(\tau'\tau'') := x$.

The other rules have similar proof but easier than those described above. □

Theorem 4.7 (λ-term to ω-graph) *For each λ-term M without variable clash there exist an ω-graph $\mathcal{G}(r) = \langle\mathcal{V}, \mathcal{T}, r\rangle$ and a surjective function $f : \mathcal{T} \mapsto Subterms(M)$ such that f has the properties listed in the theorem 4.6 and $f(\tau_r) = M$.*

Proof. Let M be a λ-term without variable-clash, will be proved by induction over the λ-term construction.
Induction Basis: M is a variable x. So just apply the rule (ax) in the $\mathbb{G}(\omega)$ calculus, then there is $\langle\mathcal{V}, \mathcal{T}, r\rangle$ a ω-graph by theorem 4.5 such that $\mathcal{T} = \{\tau\}, \mathcal{V} = \{r\}$ and τ is a free edge. Let $f : \mathcal{T} \mapsto \{x\}$ a function such that $f(\tau) = x$, then f is a surjective function and every property listed in the theorem 4.6 is true by vacuity.
Induction Step: Suppose that M is a non-variable λ-term, then theres is two possible cases.

1) Case $M = \lambda x \cdot N$: So by (IH) there is a ω-graph $\mathcal{G}_N = \langle\mathcal{V}', \mathcal{T}', r'\rangle$ and a function $f : \mathcal{T}' \mapsto Subterms(N)$ like is described in the statement.
   (a) Theres is exactly $n > 1$ occurrences of x in N: All occurrences of x in N are free because N does not have variable-clash: Note that f is a surjective function so $f^{-1}(x) \neq \emptyset$.
   Let $\tau' \in f^{-1}(x)$ the τ' could be in $\partial\mathcal{T}'$ or $\mathcal{T}' \setminus \partial\mathcal{T}'$. Suppose by contradiction that τ' belongs to $\mathcal{T}' \setminus \partial\mathcal{T}'$. τ' could be an edge, a loop or a bound m-`whip`.
   - Case τ' is a loop: there is $v \in \mathcal{V}'$ such that $L_v(1) = L_v(2) = \tau'$ and v is an `abs`-node, so x occurs bound in N, which is a contradiction.
   - Case τ' is a bound m-`whip`: there is $v \in \mathcal{V}'$ such that $L_v(2) = \tau'$ and v is an `abs`-node, so x occurs bound in N, which is a contradiction.

- Case τ' is an edge: $v_1, v_2 \in \mathcal{V}'$ distinct such that $\tau' = L_{v_1}(i_1)$ and $\tau' = L_{v_2}(i_2)$, note that i_1, i_2 cannot be 0 otherwise $x = f(\tau') = \lambda y \cdot t$ or $x = f(\tau') = (tt')$, where t, t' are λ-terms, so $i_1, i_2 \geq 1$.
  supposing that $i_1 = 1$: There is a topological edge τ'' such that $f(\tau'') = \lambda y \cdot x$ or $f(\tau'') = (xt)$ or $f(\tau'') = (tx)$, implying that v_2 is not a `app`-node otherwise $i_2 = 0$ because the decomposition rule statement assures that i_2=0, so a contradiction, implying v_2 is a (λ)-node and by the same reason of the `app`-node, $i_2 \neq 1$. following that $i_2 = 2$ so x occurs bound in N, again a contradiction.
  Supposing $i_1 = 2$: v_2 could not be a `abs`-node, because x is a free variable and N has no variable-clash, implying that v_1 is `app`-node and the rest is similar to the previous argument.

  Therefore $\tau' \in \partial\mathcal{T}'$, and it is proved that $f^{-1}(x) \subseteq \partial\mathcal{T}'$, for this reason cannot exist $\tau'' \in f^{-1}(x)$ distinct of τ', the property (i) assures that for each $\tau', \tau'' \in \partial\mathcal{T}'$ distinct, $f(\tau'), f(\tau'')$ are distinct variables. So it is proved that $f^{-1}(x) = \{\tau'\}$, and because $n > 1$ exist at least 2 nodes v, v'(they could be equal, but in this case v is an `app`-node and $L_v(1) = L_v(2) = \tau'$) having τ' as incident topological edge and for this reason τ' must be a free $(n-1)$-`whip`.

  Applying the ω-abs rule over τ', there is new ω-graph $\mathcal{G} := \langle \mathcal{V}, \mathcal{T}, r\rangle = \langle \mathcal{V}'\dot\cup\{r\}, (\mathcal{T}' \setminus \{\tau'\})\dot\cup\{\tau, \tau_r\}, r\rangle$, where r is the new root, τ_r is the rooted edge, r' the rooted vertex, τ is a bound n-`whip` connected to r' and all vertices from $\Delta\tau'$. Now defining the function $F : \mathcal{T} \mapsto \mathit{Subterms}(\lambda x \cdot N)$ such that $F(\tau_r) = \lambda x \cdot N$, $F(\tau) = f(\tau') = x$ and $F|_{\mathcal{T}\setminus\{\tau,\tau_r\}} = f|_{\mathcal{T}\setminus\{\tau,\tau_r\}}$, and the results follows because r' is an `abs`-node in $\mathcal{G}$ with the function L over $\mathcal{G}$, $L_{r'}(0) = \tau_r$ the rooted edge in $\mathcal{G}$, $L_{r'}(1) = \tau_{r'}$ the rooted edge in $\mathcal{G}_N$ and $L_{r'}(2) = \tau$.

(b) There is exactly one occurrence of the variable x in N: Analogous to the previous case, except that τ' will be a free edge and the `abs`-rule will be used.

(c) x does not occur in N: This case is trivial, applying the induction over N getting the function f and a ω-graph $\mathcal{G}_N = \langle \mathcal{V}', \mathcal{T}', r'\rangle$, exactly as the first case, then applying the (abs^*)-rule obtaining the new ω-graph $\mathcal{G} := \langle \mathcal{V}, \mathcal{T}', r'\rangle = \langle \mathcal{V}\dot\cup\{r\}, \mathcal{T}\dot\cup\{\tau_r\}, r\rangle$, where, r is the root, τ_r is the rooted edge connected r'. So Make $F\mathcal{T} \mapsto \mathit{Subterms}(\lambda x \cdot N)$ where $F(\tau_r) = \lambda x \cdot N$ and $F|_{\mathcal{T}'} = f$, and the result follows because $\mathtt{deg}_{\mathcal{T}}(r') = 2$, so r' is an `abs`-node

2) Case $M = (N_1 N_2)$: By (IH) over M_1, M_2 exists ω-graphs $\mathcal{G}_i = \langle \mathcal{V}_i, \mathcal{T}_i, r_i\rangle$ and surjective functions $f_i\mathcal{T}_i \mapsto \mathit{Subterms}(M_i)$ for $i \in \{1, 2\}$ with the properties described in the statement, and there is two possibilities for this case:

   (a) Case $FV(M_1) \cap FV(M_2) = \emptyset$: This case is trivial, just apply the `app`-rule resulting in a new ω-graph $\mathcal{G} := \langle \mathcal{V}, \mathcal{T}, r\rangle = \langle \mathcal{V}_1\dot\cup(\mathcal{V}_2\setminus\{r_2\})\dot\cup\{r\}, \mathcal{T}_1\dot\cup(\mathcal{T}_2\setminus\{\tau_{r_2}\})\dot\cup\{\tau_r, \gamma\}, r\rangle$ where r is the root, τ_r is the rooted edge connected to r_1, γ is an edge who connects the rooted vertex of $\mathcal{G}_2$ to the right side of r_1 in $\mathcal{G}$. Then the function $F : \mathcal{T} \mapsto \mathit{Subterms}(M_1 M_2)$ is defined by $F(\gamma) = f_2(\tau_{r_2}), F(\tau_r) = (M_1 M_2), F|_{\mathcal{T}_1} := f_1$ and $F|_{\mathcal{T}_2\setminus\{\tau_{r_2}\}} := f_2|_{\mathcal{T}_2\setminus\{\tau_{r_2}\}}$. And the result follows because r_1 is an `app`-node in $\mathcal{G}$ with $L_{r_1}(0) = \tau_r, L_{r_1}(1) = \tau_{r_1}$ and $L_{r_1}(2) = \gamma$.

   (b) Case $FV(M_1) \cap FV(M_2) = \{x_1, \ldots, x_m\}$: Exactly in the previous case except that Will be applied the $\omega - app$ rule with the following list of joined free edges or free `whips` $\tau_{11}\tau_{21}, \ldots, \tau_{1m}\tau_{2m}$ where $\tau_{ij} = f_i^{-1}(x_j)$, and $F(\tau_{1j}\tau_{2j}) := x_j$. Them the result follows.

□

E Proofs of Section 5

For the proof of the following lemma, recall the notion of graph `Skeleton` introduced in Subsection C.1.

Lemma E.1 (Partial Typing-Function Lemma) *Let $\langle \mathcal{V}, \mathcal{T}\rangle$ and $\langle \mathcal{V}', \mathcal{T}'\rangle$ be CL^g-graphs, $\tau \in \partial\mathcal{T}$, $\tau' \in \partial\mathcal{T}'$ free edges, $E : \mathcal{T} \longrightarrow$ `Type` a partial function and $T' = \mathtt{PTG}(\mathcal{T}', \mathcal{V}', \tau', E)$ such that $\langle \mathcal{V}, \mathcal{T}\rangle$ it is not a degenerated topological graph, $(\mathcal{T}, \mathcal{V}, \tau) \Longrightarrow^*_{\mathcal{D}} (\mathcal{T}', \mathcal{V}', \tau')$ and $T' \neq \emptyset$,* `FAIL`.

(i) *If $\mathcal{D}om(E) \cap \mathtt{Skel}_{\mathcal{T}'}(\tau') = \emptyset$ then there exists a type substitution $\mathbf{t}$ such that $\mathbf{t}(E) \subseteq T$.*

(ii) *If $\mathcal{D}om(E) \cap \mathtt{Skel}_{\mathcal{T}'}(\tau') = \emptyset$ then $T' = \mathtt{PTG}(\mathcal{T}', \mathcal{V}', \tau', E|_{\mathcal{T}'}) \cup \mathbf{t}(E|_{\mathcal{D}om(E)\setminus\{\mathcal{T}'\}})$, for some type substitution $\mathbf{t}$.*

Proof.

(i) The proof is by by induction on $|\mathcal{T}'|$.
   Induction Basis: Suppose $|\mathcal{T}'| = 1$, so $\mathcal{V}' = \emptyset$ because $\langle \mathcal{T}', \mathcal{V}'\rangle$ is a CL^g-graph. Then $\langle \mathcal{T}', \mathcal{V}'\rangle$ is a degenerated topological graph. Since $T' = PTG(\mathcal{T}', \mathcal{V}', \tau', E)$, by the algorithm $T = E$ or $T = E\dot\cup\{\tau' \mapsto a\}$ where a is an atom. Then, it results straightforward that $E \subseteq T$, and by defining $\mathbf{t}$ as the identity type substitution, one obtains that $\mathbf{t}(E) = E \subseteq T$.
   Induction Step: Suppose that $|\mathcal{T}'| > 1$.
   Then $\tau' \cap \mathcal{V}' = \{v'\}$ and $\langle \mathcal{T}', \mathcal{V}'\rangle$ is not a degenerated graph. Thus, there are two possible cases:

(a) v' **is an abs-node:**
Then $(\mathcal{T}', \mathcal{V}', \tau') \Longrightarrow_{\mathcal{D}} (\mathcal{T}'', \mathcal{V}'', \tau'')$, and if $\langle \mathcal{T}'', \mathcal{V}'' \rangle$ is a degenerated topological graph then for $T'' := PTG(\mathcal{T}'', \mathcal{V}'', \tau'', E)$ the induction basis proved that $E \subseteq T''$, but by the algorithm $(\mathcal{T}', \mathcal{V}', \tau', E) \Longrightarrow_{\mathcal{PTG}} (\mathcal{T}'', \mathcal{V}'', \tau'', E)$ with the $\mathcal{D}$-abs rule, so $T' = T'' \setminus \{\tau' \mapsto T''(\tau')\}\{\tau' \mapsto \rho \to \sigma\}$ implying that $E \subseteq T'$ because $\tau' \in \mathtt{Skel}_{\mathcal{T}'}(\tau')$. Then, $\tau' \notin \mathcal{D}om(E)$ which implies that $E \subseteq T'' \setminus \{\tau' \mapsto T''(\tau')\}$. Thus, taking $\mathbf{t}$ as the identity type substitution will be enough.
Suppose that $\langle \mathcal{T}'', \mathcal{V}'' \rangle$ is not a degenerated topological graph. Then $\tau'' \in \mathtt{Skel}_{\mathcal{T}''}(\tau'')$ by definition of `Skeleton` and by the Lemma C.3, $\tau'' \in \mathtt{Skel}_{\mathcal{T}''}(\tau'') \subseteq \mathtt{Skel}_{\mathcal{T}'}(\tau') \subseteq \mathtt{Skel}_{\mathcal{T}}(\tau)$, implying that $\mathcal{D}om(E) \cap \mathtt{Skel}_{\mathcal{T}''}(\tau'') = \emptyset$ and by (IH) there exists a type substitution $\mathbf{t}$ such that $\mathbf{t}(E) \subseteq T''$, but $T' = T'' \setminus \{\tau' \mapsto T''(\tau')\} \cup \{\tau' \mapsto \rho \to \sigma\}$. Note that $\tau' \in \mathtt{Skel}_{\mathcal{T}'}(\tau')$, then $\tau' \notin \mathcal{D}om(E)$ implies that $\mathbf{t}(E) \subseteq T'' \setminus \{\tau' \mapsto T''(\tau')\}$. Therefore, $\mathbf{t}(E) \subseteq T'$.

(b) v' **is an app-node:**
Then $(\mathcal{T}', \mathcal{V}', \tau') \Longrightarrow_{\mathcal{D}} (\mathcal{T}_i, \mathcal{V}_i, \tau_i)$ for $i = 1, 2$, where $L_{v'}(i) = \tau_i$, $T_1 := PTG(\mathcal{T}_1, \mathcal{V}_1, \tau_i, E)$ and $T_2 := PTG(\mathcal{T}_2, \mathcal{V}_2, \tau_2, T_1)$. Since $T' \neq \emptyset$, `FAIL`, $T_i \neq \emptyset$, `FAIL` for $i = 1, 2$, by analysing the TPG algorithm. Note that by Lemma C.3 $\mathtt{Skel}_{\mathcal{T}_i}(\tau_i) \subseteq \mathtt{Skel}_{\mathcal{T}'}(\tau')$ then $\mathcal{D}om(E) \cap \mathtt{Skel}_{\mathcal{T}_1}(\tau_1) = \emptyset$, and by (IH), it follows that there exists a type substitution $\mathbf{t}_1$ such that $\mathbf{t}_1(E) \subseteq T_1$.
Suppose by contradiction that there exists a $\hat{\tau} \in \mathcal{D}om(T_1) \cap \mathtt{Skel}_{\mathcal{T}_2}(\tau_2)$, so by Lemma C.4 $\hat{\tau}$ is an edge of $\langle \mathcal{T}_2, \mathcal{V}_2 \rangle$ or $\hat{\tau} = \tau_2$, but if $\hat{\tau} = \tau_2$ then $\hat{\tau} \notin \mathcal{T}_1$. But by the secondary hypothesis $\hat{\tau} \in \mathcal{D}om(T_1)$ one has that $\tau_2 \in \mathcal{D}om(E)$, which is a contradiction of the primary hypothesis. So, $\hat{\tau}$ is an edge of $\langle \mathcal{T}_2, \mathcal{V}_2 \rangle$. Note that $\hat{\tau} \in \mathtt{Skel}_{\mathcal{T}_2}(\tau_2)$ implies $\hat{\tau} \notin \mathcal{D}om(E)$ and $\hat{\tau} \in \mathcal{D}om(T_1)$ with $\hat{\tau} \notin \mathcal{D}om(E)$, and this implies that $\hat{\tau} \in \mathcal{T}_1$. So there is a contradiction in the assertion that $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ because $\hat{\tau} \in \tau_i$ implies $\hat{\tau} \cap \mathcal{V}_i \neq \emptyset$ for $i = 1, 2$. Therefore, $\mathcal{D}om(T_1) \cap \mathtt{Skel}_{\mathcal{T}_2}(\tau_2) = \emptyset$. Then by (IH) there should exist a type substitution $\mathbf{t}_2$ such that $\mathbf{t}(T_1) \subseteq E_2$ and therefore $\mathbf{t}_2(\mathbf{t}_1(E)) \subseteq T_2$.
Suppose that $T_1(\tau_1)$ is an atom, then by the algorithm $\mathbf{t}_3 = \mathtt{Unify}(a, T_2(\tau_2) \to b) \neq \mathtt{FAIL}$ because $T' \neq \emptyset$. On the other side, if $T_1(\tau_1) = \rho \to \sigma$ then $\mathbf{t}_3 = \mathtt{Unify}(\rho, T_2(\tau_2)) \neq \mathtt{FAIL}$ by the same reason. So there exists $T' = \mathbf{t}_3(T_2 \setminus \{\tau' \mapsto T_2(\tau')\} \cup \{\tau' \mapsto \mathbf{t}_3(\hat{\sigma})\}$ where $\hat{\sigma} = \sigma$ or the fresh atom b.
Note that $\mathbf{t}_3 \circ \mathbf{t}_2 \circ \mathbf{t}_1(E) \subseteq \mathbf{t}_3(T_2 \setminus \{\tau' \mapsto T_2(\tau')\})$ because $\tau' \in \mathtt{Skel}_{\mathcal{T}'}(\tau')$ then the result follows.

(ii) The proof is a direct adaptation of item (i).

□

Lemma E.2 *Let $\langle \mathcal{V}, \mathcal{T} \rangle$ be a CL^g-graph, $\tau \in \partial\mathcal{T}$ a free edge, $E : \mathcal{T} \mapsto$ `Type` a partial function such that $E(\hat{\tau})$ is an atom for each $\hat{\tau} \in \mathcal{D}om(E)$. Then,*

$$T_1 := \mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, \emptyset) \neq \emptyset, \mathtt{FAIL} \text{ iff } T_2 := \mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E) \neq \emptyset, \mathtt{FAIL},$$

and also $T_1 =_\alpha T_2$, i,e, T_1 differs from T_2 only by a swap of atoms in their image.

Proof. It is straightforward from the algorithm, because every $\tau' \in \mathcal{D}om(E) \cap \mathtt{Skel}_{\mathcal{T}}(\tau)$, $E(\tau')$ will be replaced by a proper type, and if $\tau' \in \mathcal{D}om(E) \cap \mathcal{T} \setminus \mathtt{Skel}_{\mathcal{T}}(\tau)$, τ' will be a normal form of $(\mathcal{T}, \mathcal{V}, \tau)$. So, the atom $E(\tau')$ will be preserved and used in the algorithm. □

Lemma E.3 *Let $\langle \mathcal{V}, \mathcal{T} \rangle$ be a CL^g-graph, $\tau \in \partial\mathcal{T}$ a free edge, $E, E' : \mathcal{T} \mapsto$ Types partial functions such that $\mathcal{D}om(E') = \mathcal{D}om(E)$ and $E'(\hat{\tau})$ is a fresh atom for each $\hat{\tau} \in \mathcal{D}om(E')$. If $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E) \neq \emptyset, \mathtt{FAIL}$ then $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E') \neq \emptyset, \mathtt{FAIL}$*

Proof. It is enough to prove the counter positive of this lemma. Suppose that $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E') \neq \mathtt{FAIL}$ and $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E') = \emptyset$, so take $\mathbf{t}$ a type substitution such that $\mathbf{t}(E') = E$. It exists because $\mathcal{D}om(E') = \mathcal{D}om(E)$ and $E'(\tau')$ is fresh and atomic for all $\tau' \in \mathcal{D}om(E')$. Note that $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E) \neq \mathtt{FAIL}$ because this only depends on $\Longrightarrow_{\mathcal{D}}$. So, take the first instance of the execution tree of $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E')$ such that it returns $\emptyset$. By the algorithm such that instance represents an `app`-node and a failed unification. Then, apply $\mathbf{t}$ to all successor instances and now there are instances of $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E)$ and the unification fails too. This happens because if $\rho =^? \sigma$ is not unifiable then for all type substitution π, $\pi(\rho) =^? \pi(\sigma)$ it is not unifiable, in particular for $\mathbf{t}$. □

Lemma E.4 *Let $\langle \mathcal{V}, \mathcal{T} \rangle$ be a CL^g-graph, $\tau \in \partial\mathcal{T}$ a free edge, $E : \mathcal{T} \mapsto$ `Type` a partial function and $T = \mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, \emptyset) \neq \mathtt{FAIL}$. If $(\langle \mathcal{V}, \mathcal{T} \rangle, E)$ is a typed graph with respect to τ then $E = \mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, E)$.*

Proof. Note that in the execution tree of the algorithm for $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, \emptyset)$ no new atom will be created because $E(\tau')$ is defined for each $\tau' \in \mathcal{T}$ and for each $v \in \mathcal{V}$ such that v is an `app`-node, $E(L_v(1))$ is a composite type. On the other side, all unifications are trivial, i.e., for each instance $\mathbf{t}$ is the identity. □

Lemma E.5 *Let $\langle \mathcal{V}, \mathcal{T} \rangle$ be a CL^g-graph, $\tau \in \partial\mathcal{T}$ a free edge, $E : \mathcal{T} \mapsto$ `Type` a partial function and $T = \mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, \emptyset) \neq \mathtt{FAIL}$. If $(\langle \mathcal{V}, \mathcal{T} \rangle, E)$ is a typed graph w.r.t τ then $\mathtt{PTG}(\mathcal{T}, \mathcal{V}, \tau, \emptyset) \neq \emptyset$*

Proof. The proof follows from the previous lemmas. □

Theorem 5.5 (Soundness and Completeness of PTG) *Let* $\langle\mathcal{V},\mathcal{T}\rangle$ *be a* CL^g*-graph,* $\tau \in \partial\mathcal{T}$ *a free-edge,* $T : \mathcal{T} \mapsto \mathtt{Type}$ *a partial function*

(i) *(Soundness) If* $E = \mathtt{PTG}(\mathcal{T},\mathcal{V},\tau,T) \neq \emptyset$ *and* $E \neq \mathtt{FAIL}$ *then* $(\langle\mathcal{V},\mathcal{T}\rangle, E)$ *is a typed* CL^g*-graph w.r.t.* τ*.*

(ii) *(Completeness) If* $E = \mathtt{PTG}(\mathcal{T},\mathcal{V},\tau,\emptyset) \neq \mathtt{FAIL}$ *and* $(\langle\mathcal{V},\mathcal{T}\rangle, T)$ *is a typed graph w.r.t.* τ *then there exists a type substitution* $\mathbf{t}$ *such that* $T(\tau) = \mathbf{t}(E(\tau))$*.*

Proof.

(i) Considering $E = \mathtt{PTG}(\mathcal{T},\mathcal{V},\tau,T)$ the current instance of PTG, the proof is by induction on $|\mathcal{T}|$.
Induction Basis: Suppose $|\mathcal{T}| = 1$, then $\langle\mathcal{V},\mathcal{T},\tau\rangle$ is a degenerated graph and if $\tau \notin \mathcal{T}T$ then PTG returns τ into a fresh atom (lines 4)
if it is undefined in the current T, otherwise it is preserved (lines 4 to 6).
Inductive Step: Suppose that $\mathcal{V} \neq \emptyset$ and $\Delta\tau = \{v\}$. The proof follows by analysing the vertex v.

(a) v **is an abs-node** (line 11, algorithm PTG):
In this case, $(\mathcal{T},\mathcal{V},\tau,T) \Longrightarrow_{\mathtt{PTG}} (\mathcal{T}',\mathcal{V}',\tau',T)$ where $(\mathcal{T},\mathcal{V},\tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}',\mathcal{V}',\tau')$. By Theorem 3.9, $\langle\mathcal{V}',\mathcal{T}'\rangle$ is a CL^g-graph and $\tau' \in \partial\mathcal{T}'$. By (IH) $(\langle\mathcal{V}',\mathcal{T}'\rangle, E')$ is a typed graph w.r.t. τ' and $E' = \mathtt{PTG}(\mathcal{T}',\mathcal{V}',\tau',T)$. Notice that $E = (E' \setminus \{\tau \mapsto T(\tau)\}) \cup \{\tau \mapsto \rho \to \sigma\}$, where $\sigma = E'(L_v(1))$. Notice also that ρ is a fresh atom, if $\mathtt{deg}_{\mathcal{T}}(v) = 2$ (line 17) and, $\rho = E'(L_v(2))$, when $\mathtt{deg}_{\mathcal{T}}(v) = 3$ (line 19).

(b) v **is an app-node** (line 20):
This case is analogous to the previous one, but PTG splits into two instances (lines 22 and 26), resulting in $E_1 = \mathtt{PTG}(\mathcal{T}_1,\mathcal{V}_1,\tau_1,T)$ and $E_2 = \mathtt{PTG}(\mathcal{T}_2,\mathcal{V}_2,\tau_2,E_1)$ where $(\mathcal{T},\mathcal{V},\tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}_i,\mathcal{V}_i,\tau_i), i = 1,2$. By (IH) one has that $(\langle\mathcal{V}_i,\mathcal{T}_i\rangle, E_i), i = 1,2$ are typed graphs w.r.t τ_i. Since $E \neq \emptyset$, the unification does not fail hence the result follows.

(ii) The proof is by induction on $|\mathcal{T}|$ in a more general result that includes the statement of theorem.
Induction Hypothesis: Suppose $(\langle\mathcal{V},\mathcal{T}\rangle, T)$ is a typed graph w.r.t. τ then there exists a type substitution $\mathbf{t}$ such that $T(\tau) = \mathbf{t}(E(\tau))$ and $T(\gamma) = \mathbf{t}(E(\gamma))$, for each $\gamma \in \partial\mathcal{T}$.
Induction Basis: if $|\mathcal{T}| = 1$ then $\langle\mathcal{V},\mathcal{T}\rangle$ is a degenerated topological graph, then $E = \{\tau \mapsto a\}$ and a is an atom. The result is straightforward: just use the type substitution $\mathbf{t} := \{a \mapsto T(\tau)\}$.
Inductive Step: if $|\mathcal{T}| > 1$ then $\Delta\tau = \{v\}$ and since $E \neq \mathtt{FAIL}$, by Lemma C.1, v is an app-node or an abs-node.
By hypothesis, $(\langle\mathcal{V},\mathcal{T}\rangle, T)$ is a typed graph, then by Lemma E.5, $E \neq \emptyset$. Therefore, by Theorem 5.5(i), E is a function from $\mathcal{T}$ to *Types* and $(\langle\mathcal{V},\mathcal{T}\rangle, E)$ is a typed graph.

- **Case 1:** v is an app-node.
  There are triplets $(\mathcal{T}_i,\mathcal{V}_i,\tau_i)$ with $i = 1,2$ such that $(\mathcal{T},\mathcal{V},\tau) \Longrightarrow_{\mathcal{D}} (\mathcal{T}_i,\mathcal{V}_i,\tau_i)$. Let $E_1 = \mathtt{PTG}(\mathcal{T}_1,\mathcal{V}_1,\tau_1,\emptyset)$ and $E_2 := \mathtt{PTG}(\mathcal{T}_2,\mathcal{V}_2,\tau_2,E_1)$ since $E \neq \emptyset$ and $E \neq$FAIL, by the PTG algorithm it follows that $E_i \neq \emptyset$ and $E_i \neq \mathtt{FAIL}$ for $i = 1,2$.
  By Lemma E.1 there exists a type substitution $\mathbf{t}_{12}$ such that $\mathbf{t}_{12}(E_1) \subseteq E_2$. Suppose w.l.o.g. that $E_1(\tau_1) = \rho' \to \sigma'$ is a composite type, then $E_2(\tau_1) = \mathbf{t}_{12}(E_1(\tau_1)) = \rho \to \sigma$ is a composite type. Note that $(\langle\mathcal{V}_1,\mathcal{T}_1\rangle, T|_{\mathcal{T}_1})$ is a typed graph, so by (IH) there exists a type substitution $\mathbf{t}_1$ such that $T|_{\mathcal{T}_1}(\tau_1) = \mathbf{t}_1(E_1(\tau_1))$ and for each $\gamma \in \partial\mathcal{T}_1, T(\gamma) = \mathbf{t}_1(E(\gamma))$.
  Let be $E_2' = PTG(\mathcal{T}_2,\mathcal{V}_2,\tau_2,\emptyset)$, since $E_2 \neq \emptyset$ and $E_2 \neq \mathtt{FAIL}$, $E_2' \neq \mathtt{FAIL}$ because in both running they have the same decomposition, and applying the Lemma E.5 $E_2' \neq \emptyset$. Hence it is possible to apply the (IH) since $(\langle\mathcal{V}_2,\mathcal{T}_2\rangle, T|_{\mathcal{T}_2}), (\langle\mathcal{V}_2,\mathcal{T}_2\rangle, E_2|_{\mathcal{T}_2})$ are typed graphs, implying that exists type substitutions $\mathbf{t}_2, \mathbf{t}_2'$ such that $T(\tau_2) = \mathbf{t}_2(E_2'(\tau_2)), E_2(\tau_2) = \mathbf{t}_2'(E_2'(\tau_2))$ and for all $\gamma \in \mathcal{T}_2, T(\gamma) = \mathbf{t}_2(E_2'(\gamma)), E_2(\gamma) = \mathbf{t}_2'(E_2'(\gamma))$.
  By the algorithm PTG, $\pi := \mathtt{Unify}(\rho, E_2(\tau_2)) \neq \mathtt{FAIL}$, i.e, π is the *mgu* of ρ and $E_2(\tau_2)$ and $T = \pi(E_2 \setminus \{\tau \mapsto E_2(\tau)\}) \cup \{\tau \mapsto \pi(\sigma)\}$ and E_2' has only fresh atoms in the image, so $\mathbf{t}_2'(E_2) = E_2$, implying that $\mathbf{t}_2' \circ \mathbf{t}_{12}(E_1) = \mathbf{t}_{12}(E_1)$ and $\mathbf{t}_{12}(E_2') = E_2'$.
  For the main result it is enough to prove that $\mathbf{t}_1 = \mathbf{t} \circ \pi \circ \mathbf{t}_{12}$, for some type substitution $\mathbf{t}$. Defining $H := \{(E_1(\gamma), E_2'(\gamma)) \mid \gamma \in \partial\mathcal{T}_1 \cap \partial\mathcal{T}_2\}$, is the set of all pairs of types attributions to the same whip belonging to both $\mathcal{T}_1,\mathcal{T}_2$. In the one hand, for all $\gamma \in \mathcal{T}_1 \cap \mathcal{T}_2$ then $\mathbf{t}_2' \circ \mathbf{t}_{12}(E_1(\gamma) = \mathbf{t}_{12}(E_1(\gamma) = E_2(\gamma) = \mathbf{t}_2'(E_2'(\gamma)) = \mathbf{t}_2' \circ \mathbf{t}_{12}(E_2'(\gamma))$, then $\mathbf{t}_2' \circ \mathbf{t}_{12}$ unifies H and more over, it is a *mgu* of H. In the other hand, $\rho'' \to \sigma'' = T(\tau_1) = \mathbf{t}_1(E(\tau_1) = \mathbf{t}_1(\rho' \to \sigma')$ and $\rho'' = T(\tau_2) = \mathbf{t}_2(E_2'(\tau_2))$ then $\mathbf{t}_2 \circ t_1$ unifies $\rho', E_2'(\tau_2)$ and by the same argument $\mathbf{t}_2 \circ \mathbf{t}_1$ also unifies H.
  Remember that $\mathbf{t}_2' \circ \mathbf{t}_{12}(E_1(\tau_1)) = \mathbf{t}_2'(E_2(\tau_1)) = E_2(\tau_1) = \rho \to \sigma$ and $\mathbf{t}_2' \circ \mathbf{t}_{12}(E_2'(\tau_2)) = \mathbf{t}_2'(E_2'(\tau_2)) = E_2(\tau_2)$, this implies that π is an *mgu* of $\mathbf{t}_2 \circ \mathbf{t}_{12}(E_2'(\tau_2))$ and $\mathbf{t}_2 \circ \mathbf{t}_{12}(\rho')$, hence $\pi \circ \mathbf{t}_2' \circ \mathbf{t}_{12}$ is a *mgu* for $(\rho', E_2'(\tau_2))$ and H.
  Because $\pi \circ \mathbf{t}_2' \circ \mathbf{t}_{12}$ is an *mgu* of $(\rho', E_2'(\tau_2))$ and H, there exists π' such that $\pi' \circ \pi \circ \mathbf{t}_2' \circ \mathbf{t}_{12} = \mathbf{t}_2 \circ \mathbf{t}_1$ and considering only atoms occurring in E_1, E_2, E and T, there is $\pi \circ \mathbf{t}_2' \circ \mathbf{t}_{12} = \pi \circ \mathbf{t}_{12}$ and removing all unnecessary atoms from the domain of π' obtaining $\mathbf{t}$ such that $\mathbf{t} \circ \pi \circ \mathbf{t}_{12} = \mathbf{t}_1$.

- v is an `abs`-node: This is straightforward from (IH).

□