$\boldsymbol{T}$-joins in infinite graphs

Attila Joó

MTA-ELTE Egerváry Research Group
Budapest, Hungary

joapaat@cs.elte.hu

Submitted: Jul 14, 2016; Accepted: Jun 25, 2017; Published: Jul 14, 2017
Mathematics Subject Classifications: 05C38; 05C38; 05C70

Abstract

We characterize the class of infinite connected graphs G for which there exists a T-join for any choice of an infinite $T \subseteq V(G)$. We also show that the following well-known fact remains true in the infinite case. If G is connected and does not contain a T-join, then it will if we either remove an arbitrary vertex from T or add any new vertex to T.

Keywords: T-join; infinite graph

1 Introduction

The graphs in this paper may have multiple edges although all of our results follow easily from their restrictions to simple graphs. Loops are irrelevant, hence we throw them away automatically if some arise during a construction. The 2-edge-connected components of a graph are its maximal 2-edge-connected subgraphs (a graph consisting of a single vertex is considered 2-edge-connected). A ***T*-join** in a graph G, where $T \subseteq V(G)$, is a system $\mathcal{P}$ of edge-disjoint paths in G such that the endvertices of the paths in $\mathcal{P}$ create a partition of T into two-element sets. In other words, we match by edge-disjoint paths the vertices in T. In the finite case the existence of an $F \subseteq E(G)$ for which $d_F(v)$ is odd if and only if $v \in T$ is equivalent with the existence of a T-join. Indeed, the united edge sets of the paths in $\mathcal{P}$ forms such an F, and such an F can be decomposed into a T-join and some cycles by the greedy method. Sometimes F itself is called a T-join. The two possible definitions are no more closely related in the infinite case. Take for example a one-way infinite path where T contains only its endvertex. Then there is no T-join according to the path-system based definition (which we will use during this paper) but the whole edge set forms a T-join with respect to the second definition.

T-join is a common tool in combinatorial optimization problems such as the well-known Chinese postman problem. For a detailed survey one can see [1]. For finite connected graphs the necessary and sufficient condition for the existence of a T-join is quite simple, $|T|$ must be even. Indeed, the necessity of the condition is trivial. For the sufficiency let $|T| = 2k$ and we apply induction on k. The case $k = 0$ is clear. If $|T| = 2k + 2$, then remove two vertices, u and v say, of T to obtain T'. By induction we have a T'-join. Take the symmetric difference of the edge set of a T'-join and the edges of an arbitrary path between u and v. By the greedy method we can partition the resulting edge set into a T-join and some cycles. If $|T|$ is even but G is infinite, then the same proof works. In this paper we investigate questions related to the existence of T-joins where T is infinite. For infinite T one can not guarantee in general the existence of a T-join in a connected graph. Consider for example an infinite star and subdivide all of its edges by a new vertex (see Figure 1) and let T be consist of the whole vertex set.

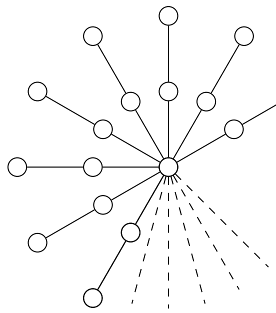

Figure 1: A subdivided infinite star. It has no T-join if T is the whole vertex set.

One of our main results is that essentially Figure 1 is the only counterexample, more precisely:

Theorem 1. *A connected infinite graph G contains a T-join for every infinite $T \subseteq V(G)$ if and only if there is no $U \subseteq V(G)$ for which $G - U$ has infinitely many connected components such that all of them are nontrivial (i.e. not consist of a single vertex) and are connected to U in G by a single edge.*

The "if" direction is straightforward since if such an U exists, then one can choose one vertex from U and two from each connected component of $G - U$ to obtain a T for which there is no T-join in G.

Our other result describes the effect of finite modifications of T on the existence of a T-join. For finite T if G does not contain a T-join, then it contains a T'-join whenever $|(T \setminus T') \cup (T' \setminus T)| =: |T \triangle T'|$ is odd. (It is obvious, since in this case the existence of a T-join depends just on the parity of $|T|$.) Surprisingly this property remains true for infinite T as well. We have the following result about this.

Theorem 2. *The class $\{(G, T) :\ G$ is a connnected graph and $T \subseteq V(G)\}$ can be partitioned into three nonempty subclasses defined by the following three properties.*

(A) G contains a T'-join whenever $|T' \triangle T| < \infty$.

(B) G contains T'-join if $|T' \triangle T|$ is finite and even, but G has no T'-join if $|T' \triangle T|$ is odd.

(C) G contains T'-join if $|T' \triangle T|$ is finite and odd, but G has no T'-join if $|T' \triangle T|$ is even.

For a countable T our proofs based on purely combinatorial arguments. To handle uncountable T we apply the so called elementary submodel technique. We do not assume previous knowledge about this method. We build it up shortly and recommend [3] for a more detailed introduction.

To make the descriptions of the T-join-constructing processes more reader-friendly we introduce the following single player game terminology. There is an abstract set of tokens and every token is on some vertex of a graph G. (At the beginning typically we have exactly one token on each element of a prescribed vertex set T and none on the other vertices.) If two tokens are on the same vertex, then we may remove them (we say that we **match** them to each other). If we have a token t on the vertex u and uv is an edge of G, then we may move t from u to v, but then we have to delete uv from G. A **gameplay** is a transfinite sequence of the steps above in which we move every token just finitely many times. Limit steps are defined by the earlier steps in a natural way. Indeed, we just delete all the edges that have been deleted earlier, remove the tokens that have been removed before, and put all the remaining tokens to their stabilized positions. We call a gameplay winning if we remove all the tokens eventually. To make talking about the relevant part of the graph easier we also allow as a feasible step deleting vertices without tokens on them and deleting edges. Clearly there is a T-join in G if and only if there is a winning gameplay for the game on G where the initial token distribution is defined by T.

2 The 2-edge-connected case

A subgraph of G is called ***t*-infinite** if it contains infinitely many tokens. We define the notions ***t*-finite**, ***t*-empty**, ***t*-odd** and ***t*-even** similarly.

Lemma 3. *If G is 2-edge-connected and t-infinite, then there is a winning gameplay.*

Claim 4. *Assume that G is 2-edge-connected and contains even number of tokens, but at least four including $s \neq t$. Then there is a winning gameplay in which s and t are not matched with each other and t is not moved.*

Proof. We may assume that G is finite otherwise we may take a finite 2-edge-connected subgraph that contains all the tokens. Take two edge-disjoint paths between the vertices that contain s and t and let H be the Eulerian subgraph of G consisting of these paths. We can build up G from H by adding ears (as in the ear decomposition). We apply induction on the number of ears. If there is no ear i.e. $G = H$, then we take an Eulerian cycle O in G. Fix an Eulerian orientation of O. Either this orientation or the reverse of it induces a desired gameplay.

Otherwise let Q be the last ear. If the number of tokens on the new vertices given by Q is odd (even), then match inside Q all but one (two) of these tokens and move the exception(s) to the part of the G before the addition of Q. Delete the remaining part of Q. We are done by applying the induction hypothesis. $\square$

By contracting the 2-edge-connected components of a connected graph, we obtain a tree. If R is a 2-edge-connected component of a connected graph G, then we denote by $\mathsf{tree}(\boldsymbol{G}; \boldsymbol{R})$ the tree of the 2-edge-connected components of G rooted at R. We usually pick such a root R arbitrarily without mentioning it explicitly. For a component C other than the root R, we call the parent component of C the component which is the predecessor of C with respect to the tree-order. We write $\mathsf{tree}(G)$ if it is not rooted. We do not distinguish strictly the subtrees of $\mathsf{tree}(G; R)$ and the corresponding subgraphs of G.

Claim 5. *Let G be a connected graph with infinitely many tokens on it. Assume that G has only finitely many 2-edge-connected components. Then there is a finite gameplay after which all the components of the resulting G' are 2-edge-connected and contain infinitely many tokens.*

Proof. By the pigeon hole principle there is a t-infinite 2-edge-connected component R of G. We use induction on the number k of the 2-edge-connected components. For $k = 1$ we do nothing. If $k > 1$ we take a leaf C of $\mathsf{tree}(G; R)$. If C is t-infinite we remove the unique outgoing edge of $V(C)$ in G and we use induction to the arising component other than C. If C is t-even, then we match the tokens on C with each other inside C and we delete the remaining part of C and its unique outgoing edge and use induction. In the t-odd case we match all but one tokens of C inside C and move one to the parent component. We claim that it is doable. Indeed, put a new token t to the vertex of C incident with the cut edge to the parent component. Apply Claim 4 with this t and an arbitrary s. Finally put back the token t' which has matched with the fake token t and move t' to the parent component. We delete the remaining part of C again and use induction. $\square$

Now we turn to the proof of Lemma 3. If t is a token and H is a subgraph of G, then we use the abbreviation $\boldsymbol{t \in H}$ to express the fact that t is on some vertex of H. Assume first that T is countable. For technical reasons we assume just the following weaker condition instead of 2-edge-connectedness.

$$\text{All the connected components are 2-edge-connected and t-infinite.} \tag{1}$$

Let t_0 be an arbitrary token. It is enough to show, that there is a finite gameplay such that we remove t_0 and the resulting system still satisfies condition (1). Pick two edge-disjoint paths P_1, P_2 between t_0 and any other token t^* that lies in the same connected component K as t_0. For $i \in \{1, 2\}$ let K_i be the connected component of $K - E(P_i)$ that contains t^* and t_0. We claim that either K_1 or K_2 is t-infinite. Suppose that K_1 is not. Then there is a t-infinite component K_{inf} of $K - E(P_1)$ which does not contain t^* and t_0. Note that P_2 necessarily lies in K_1. Hence K_{inf} is a subgraph of $K - E(P_2)$ and P_1 ensures that $K_{\inf}$ and K_1 belongs to the same connected component of $K - E(P_2)$. Thus $K_{\inf}$ guarantees that this component is t-infinite. We will need the following basic observation.

Observation 6. *If each component of a graph G has finitely many 2-edge-connected components, then so does $G - f$ for every $f \in E(G)$.*

By symmetry we may assume that K_1 is t-infinite. Move t_0 along the edges of P_1 one by one. If the following edge e is a bridge, and moving t_0 along e would create a t-odd component K_{odd}, then $t^* \notin K_{odd}$ because the component which contains t^* is t-infinite. In this case we delete e without moving t_0 and obtain a t-infinite component and a t-even component K_{even} that contains t_0. We match the tokens on K_{even} and erase the remaining part of it and the first phase of the process is done. If this case does not occur, then we move t_0 to t^* along P_1 and remove both.

We need to fix the condition (1). Each component of the resulting graph is either t-even or t-infinite. We match the tokens on t-even components and erase the remaining part of these t-even components. Each t-infinite component has finitely many 2-edge-connected components by Observation 6 thus we are done by applying Claim 5.

Consider now the general case where T can be arbitrary large. Examples show that our approach for countable T may fail for uncountable T at limit steps. Add a new vertex z and draw all the edges zv ($v \in T$) to obtain H. Finding a T-join in G is obviously equivalent with covering in H all the edges incident with z by edge-disjoint cycles. To reduce the problem to the countable case it is enough to prove the following claim.

Claim 7. *There is a partition of $E(H)$ into countable sets E_i ($i \in I$) in such a way, that for all $i \in I$ the graphs $H_i := (V(H), E_i)$ have the following property. The connected components of $H_i - z$ are 2-edge-connected and connect to z in H_i by either zero or infinitely many edges.*

Our proof of Claim 7 is a basic application of the elementary submodel technique. One can find a detailed survey about this method with many combinatorial applications in [3]. We give here just the fundamental definitions and cite the results that we need. Let $\Sigma = \{\varphi_1, \ldots, \varphi_n\}$ be a finite set of formulas in the language of set theory where the free variables of φ_i are $x_{i,1}, \ldots, x_{i,n_i}$. A set M is a **Σ-elementary submodel** if the formulas in Σ are absolute between $(M, \in)$ and the universe i.e.

$$[(M, \in) \models \varphi_i(a_1, \ldots, a_{n_i})] \iff \varphi_i(a_1, \ldots, a_{n_i})$$

holds whenever $1 \leqslant i \leqslant n$ and $a_1, \ldots, a_{n_i} \in M$. By using Lévy's Reflection Principle and the Downward Lowenheim Skolem Theorem (see in [2] or any other set theory or logic textbook) one can derive the following fact.

Claim 8. *For any finite set Σ of formulas, set x and infinite cardinal κ there exists a Σ-elementary submodel $M \ni x$ with $\kappa = |M| \subseteq M$.*

Now we use some methods developed by L. Soukup in [3]. Call a class $\mathfrak{C}$ of graphs **well-reflecting** if for all large enough finite set Σ of formulas, infinite cardinal κ, set x

and $G \in \mathfrak{C}$ there is a Σ-elementary submodel M with $x, G \in M$ for which $\kappa = |M| \subseteq M$ and $(V(G), E(G) \cap M)$, $(V(G), E(G) \setminus M) \in \mathfrak{C}$. ("For all large enough finite Σ" means here that there is some finite Σ_0 such that for all finite $\Sigma \supseteq \Sigma_0$.)

Theorem 9 (L. Soukup, Theorem 5.4 of [3])**.** *Assume that the graph-class $\mathfrak{C}$ is well-reflecting and $G \in \mathfrak{C}$. Then there is a partition of $E(G)$ into countable sets E_i $(i \in I)$ in such a way, that for all $i \in I$ we have $(V(G), E_i) \in \mathfrak{C}$.*

Remark 10. L. Soukup used originally a stricter notion of well-reflectingness but his proof still works with our weaker notion as well.

We apply the Theorem above to prove Claim 7. Let $\mathfrak{C}$ be the class of graphs G for which $z \in V(G)$, the connected components of $G - z$ are 2-edge-connected and connect to z in G either by infinitely many edges or send no edge to z at all. We need to show that $\mathfrak{C}$ is well-reflecting. Assume that Σ is a finite set of formulas that contains all the formulas of length at most 10^{10} with variables at most $x_1, \dots x_{10^{10}}$. (From the proof one can get an exact list of formulas need to be in Σ. The usual terminology says just to fix a large enough Σ. We decided that a more explicit definition is beneficial for readers who first met with this method.)

Let κ, x and $G \in \mathfrak{C}$ be given. By Claim 8 we can find a Σ-elementary submodel $M \ni x, G, z$ with $\kappa = |M| \subseteq M$. We know that $(G - z) \in M$ by using the absoluteness of the formula "x_1 graph obtained by the deletion of vertex x_2 of graph x_3"$\in \Sigma$ (see Claim 2.7 and 2.8 of [3] for basic facts about absoluteness). The proof of $(V(G), E(G) \cap M) \in \mathfrak{C}$ is easy. We just need the absoluteness of formulas such that "the local edge-connectivity between the vertices x_1 and x_2 in the graph x_3 is x_4" $\in \Sigma$. The hard part is to show $(V(G), E(G) \setminus M) \in \mathfrak{C}$. We use the following proposition.

Proposition 11 (Lemma 5.3 of [3])**.** *If M is a Σ-elementary submodel (for some large enough finite Σ) for which $G \in M$, $|M| \subseteq M$ and $x \neq y \in V(G)$ are in the same connected component of $(V(G), E(G) \setminus M)$ and $F \subseteq E(G) \setminus M$ separates them where $|F| \leqslant |M|$; then F separates x and y in G as well.*

If the local edge-connectivity between vertices $x \neq y$ would be one in the graph $(V(G - z), E(G - z) \setminus M)$, then we can separate them by the deletion of a single edge e. But then by applying Proposition 11 with $F = \{e\}$ we may conclude that the same separation is possible in $G - z$, which contradicts the assumption $G \in \mathfrak{C}$.

Suppose, to the contrary, that z sends finitely many, but at least one, edges, say $e_1, \dots, e_k$, to a connected component of $(V(G - z), E(G - z) \setminus M)$ in $(V(G), E(G) \setminus M)$. Let p be the endvertex of e_1 other than z. Then $F := \{e_i\}_{i=1}^k$ separates p and z in $(V(G), E(G) \setminus M)$ and $|F| < \infty$ holds. Hence F separates them in G as well which is a contradiction. Now the proof of Lemma 3 is complete.

3 The simplification process

Lemma 3 makes it possible to decide the existence of a T-join by just investigating the structure of the 2-edge-connected components and the quantity of tokens on them. Let G

be a connected graph, $T \subseteq V(G)$ and let R be a 2-edge-connected component of it. We define a graph $H = H(G, R, T)$ with a token-distribution on it. To obtain H we apply the following gameplay that we call **simplification process**. We denote by $\mathsf{subt}(\boldsymbol{C}; \boldsymbol{G}, \boldsymbol{R})$ the subtree of the descendants of the 2-edge-connected component C rooted at C in $\mathsf{tree}(G; R)$. Delete all those 2-edge-connected componets C for which $\mathsf{subt}(C; G, R)$ does not contain any token. Then consider the t-finite leafs of the reminder of $\mathsf{tree}(G; R)$. (We do not consider the root as a leaf.) Match the tokens on any t-even leaf C inside C and for t-odd leafs C move one token to the parent and match the others inside C. In both cases delete the remaining part of C. Iterate the steps above as long as possible and denote by H the resulting graph. Clearly either $H = R$ with some tokens on it or if $\mathsf{tree}(H)$ is nontrivial, then $\mathsf{subt}(C; H, R)$ must be t-infinite for all 2-edge-connected components C of H.

Claim 12. *There is a winning gameplay for the original system if and only if H is t-even or t-infinite.*

Proof. We show the "if" part here and the "only if part" later in Claim 13. In the t-even case it is obvious since H is connected. Assume that H is t-infinite. We may suppose that H is not 2-edge-connected because otherwise we are done by applying Lemma 3. Recall $\mathsf{subt}(C; H, R)$ is t-infinite for all 2-edge-connected components C of H. For each 2 -edge-connected component of H we fix a path P_C in the tree $\mathsf{subt}(C; H, R)$ that starts at C and terminates at some component of H other than C which is not t-empty.

After these preparations we do the following. If the root R is t-even or t-infinite, then we match all the tokens of it inside R (use Lemma 3 in the t-infinite case) and delete the remaining part of R. If it is t-odd we move one of its tokens, say it will turn to be t^*, to some child of R determined by the path P_R and we define $P_{t^*} := P_R$. We match the other tokens inside R and then delete the remaining part of R. At the next step we deal with the $\mathsf{subt}(C; H, R)$ trees where C is a child of R. In the cases where C is t-infinite or t-even we do the same as earlier. Assume that C is t-odd. If there is no token on C that comes from R, then we do the same as earlier. Suppose that there is i.e. t^* came to here. If there is a token on C other than t^*, then we match here t^* and send forward some other token t_1 in the direction defined by P_C (apply Claim 4 and a phantom-token). In this case we also need to fix a path for token t_1, let $P_{t_1} := P_C$. If t^* is the only token of C_0, then we move t^* in the direction P_{t^*}. We iterate the process recursively. The only not entirely trivial thing that we need to justify is that we do not move a token infinitely many times. If we moved some token t at the previous step, then we match it at the current step unless it is the only token at the corresponding 2-edge-connected component. On the other hand, when we move t for the first time we define the path P_t. The later movements of t are leaded by P_t which ensure that eventually t will meet some other token. $\square$

4 Proof of the theorems

Proof of Theorem 1. Let G be an infinite connected graph such that there is no $U \subseteq V(G)$ for which $G - U$ has infinitely many connected components, all of them are nontrivial and are connected to U in G by a single edge. Let $T \subseteq V(G)$ be infinite. Consider the vertices v of degree one that are in T i.e. there is a token on them. Move these tokens to the only possible direction and then delete all the vertices of degree one or zero. We denote the resulting graph by G' and we fix a 2-edge-connected component R of it. If G' has a t-infinite 2-edge-connected component, then it cannot vanish during the simplification process, thus the resulting H will be t-infinite and we are done by Claim 12.

Assume there is no such component. The degree of $C_0 := R$ in $\mathsf{tree}(G')$ must be finite otherwise $U := V(C_0)$ would violate the assumption about G. Since C_0 is t-finite by the pigeonhole principle there is a child C_1 of C_0 such that $\mathsf{subt}(C_1; G', R)$ contains infinitely many tokens. By recursion we obtain a one-way infinite path of $\mathsf{tree}(G')$ with vertices C_n ($n \in \mathbb{N}$) such that for every n the tree $\mathsf{subt}(C_n; G', R)$ contains infinitely many tokens. The set $U := \bigcup_{n=0}^{\infty} V(C_n)$ may not have infinitely many outgoing edges in G' otherwise U violates the condition about G. Thus for a large enough n the tree $\mathsf{subt}(C_n; G', R)$ is just a terminal segment of the one-way infinite path we constructed. This implies that infinitely many of the C_n's contain at least one token. Since such a path cannot vanish during the simplification process, it terminates with a t-infinite H again. $\square$

Proof of Theorem 2. Let a connected G and a 2-edge-connected component R of it be fixed. Let $T \subseteq V(G)$. We will show that case (A) of Theorem 2 occurs if and only if the simplification process terminates with infinitely many tokens and (B)/(C) occurs if and only if it terminates with an even/odd number of tokens respectively.

Assume first that the result H of the simplification process initialized by T (T-process from now on) is t-infinite. Let $T' \subseteq V(G)$ such that $|T' \triangle T| < \infty$. Call T'-process the simplification process with the initial tokens given by T' and denote by H' the result of it. If G has a 2-edge-connected, t-infinite component C with respect to T, then C is t-infinite with respect to T' as well. Observe that such a C remains untouched during the simplification process. Thus H' is t-infinite and therefore there is a T'-join in G by Claim 12.

We may suppose that there is no 2-edge-connected, t-infinite component in G. If a t-infinite component C arises during the T-process, then C receives a token from infinitely many children of it. Since $|T \triangle T'| < \infty$ we have $|V(D) \cap T| = |V(D) \cap T'|$ for all but finitely many 2-edge-connected component D. Hence the token-structure of $\mathsf{subt}(D; G, R)$ is the same for all but finitely many child D of C at the case of T'. Thus C will receive infinitely many tokens during the T'-process as well.

Finally we suppose that such a component does not arise, i.e. H has no t-infinite component. Then $\mathsf{tree}(H)$ must be an infinite tree since H is t-infinite. Furthermore, $\mathsf{subt}(C; H, R)$ must contain at least one token for all 2-edge-connected component C of H otherwise we may erase $\mathsf{subt}(C; H, R)$ to continue the simplification process. Fix a one-way infinite path P in $\mathsf{tree}(H)$ with vertices $C_n (n \in \mathbb{N})$, where $C_0 = R$ and infinitely many C_n contain at least one token. For a large enough n_0 the token-distribution of $\mathsf{subt}(C_{n_0}; G, R)$ is the same at the T and at the T' cases. Hence the T-process and T'-process runs identical on the subgraph of G corresponding to $\mathsf{subt}(C_{n_0}; G, R)$. Thus

$\mathsf{subt}(C_{n_0}; H, R) = \mathsf{subt}(C_{n_0}; H', R)$ holds and the tokens on them are the same. But then the terminal segment of the path P in $\mathsf{tree}(H')$ shows that H' is t-infinite as well.

Claim 13. *There is no T-join in G if the simplification process terminates with an odd number of tokens.*

Proof. Remember that no t-infinite 2-edge-connected component may arise during the simplification process in this case. Assume, to the contrary, that there is a T-join $\mathcal{P}$ in G. We play a winning gameplay induced by $\mathcal{P}$ i.e. every step we move some token along the appropriate $P \in \mathcal{P}$ towards its partner. If for a 2-edge-connected component C the subgraph $\mathsf{subt}(C; G, R)$ does not contain any vertex from T, then clearly no $P \in \mathcal{P}$ comes here. Hence we may delete these parts of the graph. If C is a leaf of (the remaining part of) $\mathsf{tree}(G; R)$ with $|T \cap V(C)|$ even, then the corresponding paths are inside C. In the odd case exactly one $P \in \mathcal{P}$ uses the unique outgoing edge of $\mathsf{subt}(C; G, R)$ and some other paths match inside C the other T-vertices of C. Thus along P we may move one token to the parent component and match the others along the other paths in C. Therefore after we do these steps the quantity of the tokens on the 2-edge-connected components will be the same as after the first step of the simplification process. Similar arguments show that the remaining graph and the quantity of the tokens on the 2-edge-connected components of this remaining graph of a successor step of the simplification process is the same as in some step of the game induced by $\mathcal{P}$. Since the first difference may not arise at a limit step for all steps of the simplification process we have a corresponding position of the gameplay induced by $\mathcal{P}$ where the remaining graph and the token quantities on the 2-edge-connected components are the same. Since $\mathcal{P}$ is a T-join, it leads to a winning gameplay, thus it may not arise a position with odd number of tokens in total. But in the final position of the simplification process we have odd number of tokens in total which is a contradiction. $\square$

Claim 14. *If the simplification process for T terminates with an even (odd) number of tokens and $|T \triangle T'| = 1$, then the simplification process for T' terminates with an odd (even) number of tokens.*

Proof. By symmetry we may let $T' = T \cup \{v\}$. If $v \in V(R)$, then the simplification process for T' runs in the same way as for T except that at the end we have the extra token on v which changes the parity of the remaining tokens as we claimed. If v is not in the root R, then it is in $\mathsf{subt}(C; G, R)$ for some child C of R. This C is closer in $\mathsf{tree}(G)$ to the 2-edge-connected component that contains v than R. On the one hand we know by induction that the parity of the number of tokens on C will be different when C will become a leaf in the case of T'. On the other hand for the other children of R this parity will be clearly the same, thus the parity of the number of the remaining tokens changed again. $\square$

The remaining part of Theorem 2 follows from Claim 13 and from the repeated application of Claim 14. $\square$

References

[1] Frank, A. A survey on T-joins, T-cuts, and conservative weightings, Combinatorics, Paul Erdős is eighty, Vol. 2 (Keszthely, 1993), 213–252, Bolyai Soc. Math. Stud., 2, Jáos Bolyai Math. Soc., Budapest, 1996.

[2] Kunen, K. *Set theory, volume 34 of Studies in Logic.* College Publications, London, 2011.

[3] Soukup, L. Elementary submodels in infinite combinatorics. *Discrete Math. 311*, 15 (2011), 1585–1598.