



HAL
open science

Simple, strict, propre, happy : accessibilité et expressivité des graphes temporels

Arnaud Casteigts, Timothée Corsini, Writika Sarkar

► To cite this version:

Arnaud Casteigts, Timothée Corsini, Writika Sarkar. Simple, strict, propre, happy : accessibilité et expressivité des graphes temporels. AlgoTel 2023 - 25èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2023, Cargese, France. <hal-04085368>

HAL Id: hal-04085368

<https://hal.science/hal-04085368v1>

Submitted on 28 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Simple, strict, propre, happy : accessibilité et expressivité des graphes temporels[†]

Arnaud Casteigts¹, Timothée Corsini¹ et Writika Sarkar²

¹Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France

²Chennai Mathematical Institute, India

Les graphes temporels sont des objets qui modélisent les réseaux dynamiques. L'accessibilité dans ces graphes se base sur la notion de chemin temporel (ou *trajet*). Contrairement au contexte des graphes statiques, la relation correspondante n'est ni symétrique, ni transitive, ce qui rend inopérant de nombreuses techniques connues. Pour ajouter à la confusion, les graphes temporels s'avèrent sensibles à de subtiles variations dans leur définition, telles que le fait que les trajets soient *stricts* ou non (c.à.d. peut-on emprunter deux arêtes consécutives à la même date?), que le graphe soit *propre* ou non (les sommets peuvent-ils être incident à plus d'une arête à une même date?), ou encore que le graphe soit *simple* ou non (une même arête peut-elle être présente à plusieurs dates?).

Dans cet article, nous étudions l'impact de ces trois paramètres sur l'accessibilité. Nos résultats révèlent un paysage complexe, dans lequel différentes combinaisons de ces paramètres induisent différents niveaux d'expressivité en termes d'accessibilité. En particulier, nous montrons des *séparations* entre l'expressivité de certaines combinaisons (résultats négatifs) et des *transformations* préservant l'accessibilité entre d'autres combinaisons (résultats positifs). Ces résultats permettent de clarifier l'impact de ces paramètres et nous pensons qu'ils peuvent être utilisés pour comprendre la perte de généralité occasionnée (ou non) par différents choix d'hypothèses.

Mots-clefs : graphes temporels, expressivité, accessibilité, fermeture transitive

1 Définitions

Les graphes temporels se sont imposés ces dernières années pour modéliser de nombreux types de réseaux dynamiques, notamment dans le domaine des télécommunications. Ces objets sont connus sous différents noms, avec des nuances qui varient d'une communauté à l'autre. On parle ainsi de graphes dynamiques, de graphes évolutifs, de graphes variant dans le temps ou de flots de liens. Dans cet article, nous utilisons une version rudimentaire où un graphe temporel est défini par un graphe simple muni d'un étiquetage sur les arêtes, c'est à dire $\mathcal{G} = (V, E, \lambda)$, où V est un ensemble fini de sommets, $E \subseteq V \times V$ est un ensemble fini d'arêtes (non-orientées) et $\lambda : E \rightarrow 2^{\mathbb{N}}$ associe à chaque arête un ensemble (non vide) d'entiers qui correspond aux temps où cette arête est présente. Étant donné un tel graphe $\mathcal{G} = (V, E, \lambda)$, le graphe $G = (V, E)$ est appelé *empreinte* de \mathcal{G} . Les graphes $G_t = (V, E_t)$ où $E_t = \{e \in E \mid t \in \lambda(e)\}$ sont appelés *snapshots* de \mathcal{G} au temps t . L'empreinte et les snapshots sont des graphes *statiques*. Un couple (e, t) tel que $t \in \lambda(e)$ est appelé *contact*. Enfin, la *durée de vie* de \mathcal{G} correspond au temps maximum utilisé par \mathcal{G} . La durée de vie sera toujours finie dans cet article, bien qu'elle puisse en général être infinie.

Un *trajet* (ou chemin temporel) est une suite de contacts $\langle (e_i, t_i) \rangle$ telle que $\langle e_i \rangle$ est un chemin dans l'empreinte de \mathcal{G} et $\langle t_i \rangle$ est une suite croissante d'entiers (stricte ou non, nous distinguerons les deux plus tard). Les trajets définissent naturellement une relation d'*accessibilité* entre les sommets, que l'on peut représenter par un graphe statique orienté *closure*(\mathcal{G}) = (V, E_c) où $(u, v) \in E_c$ si et seulement si u peut rejoindre v par un trajet. \mathcal{G} est temporellement connexe si tous les sommets peuvent se rejoindre deux à deux.

[†]La version complète de cet article a été présentée à SSS 2022 et est également disponible sur arXiv [CCS22]. Ce travail a été en partie financé par le projet ANR TEMPOGRAL (ANR-22-CE48-0001).

1.1 Simple, Strict, Propre, Happy ?

Plusieurs restrictions peuvent s'appliquer aux définitions précédentes. Dans la littérature, trois restrictions sont fréquemment utilisées :

- Les trajets doivent-ils être *stricts* ? Autrement dit, pour un trajet $\langle (e_i, t_i) \rangle$, faut-il que la suite $\langle t_i \rangle$ soit strictement croissante ou seulement croissante ?
- Le graphe est-il *propre* ? À savoir que si deux arêtes e et e' sont incidentes, on a $\lambda(e) \cap \lambda(e') = \emptyset$.
- Le graphe est-il *simple* ? À savoir que chaque arête de E n'est présente qu'un seul temps.

Certaines combinaisons de ces paramètres sont plus générales que d'autres. Cependant, elles ne sont pas toutes comparables, ce qui rend difficile la comparaison et le transfert de résultats dans le domaine des graphes temporels. Notre travail tente d'éclaircir ces différences. Pour commencer, toutes les combinaisons ne sont pas distinctes. Par exemple, si l'on considère les graphes propres, il n'y a aucune conséquence à imposer des trajets stricts (ils sont stricts de toute façon), contrairement graphes non-propres où cette distinction est cruciale. En fait, nous pouvons distinguer six configurations, illustrées à la Figure 1 :

- Non-propre, non-simple, strict (1)
- Non-propre, non-simple, non-strict (2)
- Non-propre, simple, strict (4)
- Non-propre, simple, non-strict (5)
- Propre, non-simple (3)
- Happy (= propre et simple) (6)

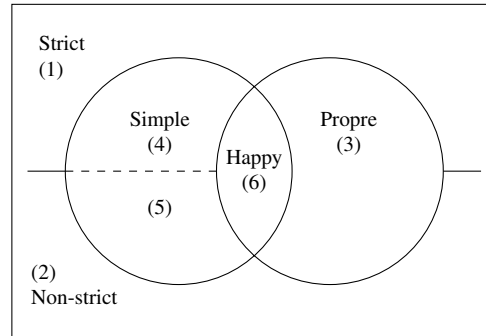


FIGURE 1 : Configurations obtenues en combinant les paramètres.

Une première subtilité apparaît : le fait d'être *propre* (ou non) et *simple* (ou non) est une propriété du graphe temporel, tandis que le fait d'être *strict* (ou non) est une propriété des trajets permis dans un tel graphe. Ainsi, les deux premiers choix peuvent être assimilés à des *familles* de graphes temporels, avec par exemple les graphes propres vus comme un cas particulier (sous-ensemble) des graphes non-propres, et similairement pour les graphes simples. Les options *strict* ou *non-strict*, quant à elles, sont plutôt vues comme une dimension supplémentaire venant compléter le choix d'une famille de graphes.

1.2 Expressivité en termes d'accessibilité

Nous nous intéressons à comparer l'expressivité des six configurations ci-dessus en fonction de l'accessibilité qu'elles permettent d'exprimer. Dans cette version courte du papier, nous dirons que deux graphes temporels ont la même accessibilité si la propriété suivante est satisfaite.

Définition 1 (Équivalence d'accessibilité). *Deux graphes temporels \mathcal{G}_1 et \mathcal{G}_2 sont équivalents en termes d'accessibilité si $\text{closure}(\mathcal{G}_1) \simeq \text{closure}(\mathcal{G}_2)$ (leurs graphes d'accessibilité sont isomorphes).*

Étant donné deux configurations \mathcal{C}_1 et \mathcal{C}_2 , on dira que $\mathcal{C}_1 \preceq \mathcal{C}_2$ si et seulement si pour tout graphe temporel \mathcal{G} interprété dans \mathcal{C}_1 , il existe un graphe temporel \mathcal{H} interprété dans \mathcal{C}_2 tel que \mathcal{G} et \mathcal{H} ont la même accessibilité. Si cette inclusion est stricte, on notera $\mathcal{C}_1 \prec \mathcal{C}_2$. Si $\mathcal{C}_1 \preceq \mathcal{C}_2$ et $\mathcal{C}_2 \preceq \mathcal{C}_1$, on dira que les deux configurations ont la même expressivité, noté $\mathcal{C}_1 \approx \mathcal{C}_2$. On pourrait déjà lister un certain nombre de relations entre configurations qui sont trivialement vraies, comme le fait que, tout autre paramètre étant égal, une configuration *non-propre* est au moins aussi expressive qu'une configuration *propre* (par inclusion). Mettant de côté ce type de relations, nous allons présenter quelques *séparations* entre configurations, ainsi que quelques *transformations* d'une configuration à une autre.

2 Séparations entre configurations

Pour séparer deux configurations, il suffit de trouver un graphe temporel dont l’accessibilité, interprétée dans la première configuration, ne peut être réalisée par aucun graphe dans la seconde. Voici les quatre séparations que nous avons obtenues dans [CCS22] :

1. “simple & strict” ne peut pas exprimer toutes les accessibilités de “non-simple & strict”;
2. “non-strict” ne peut pas exprimer toutes les accessibilités de “simple & strict”;
3. “simple & non-strict” ne peut pas exprimer toutes les accessibilités de “non-strict”;
4. “happy” ne peut pas exprimer toutes les accessibilités de “simple & non-strict”.

À titre d’exemple, la Figure 2 montre les deux graphes \mathcal{G} et \mathcal{H} utilisés dans [CCS22] pour obtenir les séparations 2 et 3, respectivement. Sans reproduire ici les arguments, le graphe \mathcal{G} dans “simple & strict” a pour accessibilité le graphe $closure(\mathcal{G})$, qui ne peut correspondre à l’accessibilité d’aucun graphe si les trajets non-stricts sont autorisés. De même, le graphe propre \mathcal{H} , qui peut être interprété indifféremment

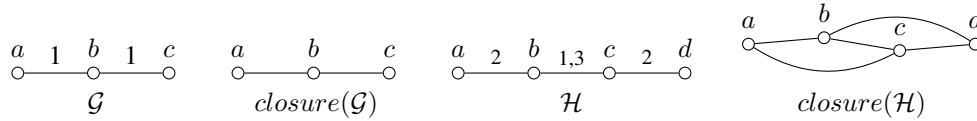


FIGURE 2 : \mathcal{G} un graphe simple et strict, \mathcal{H} un graphe non-strict et leurs fermetures.

dans la configuration “strict” ou “non-strict”, constitue donc un exemple de graphe de la configuration “non-strict” dont l’accessibilité $closure(\mathcal{H})$ ne peut être réalisée par aucun graphe dans “simple & non-strict”.

3 Transformations entre configurations

À l’inverse des séparations, il est parfois possible de concevoir des transformations canoniques entre configurations qui préservent l’accessibilité, montrant ainsi qu’une configuration est “au moins aussi” expressive qu’une autre. Nous proposons plusieurs transformations dans [CCS22]. La plus simple est la technique de *saturation*, qui prend en entrée un graphe quelconque d’une configuration “non-strict” et qui produit en sortie un graphe ayant la même accessibilité lorsque seuls les trajets stricts sont autorisés. Cette transformation démontre que “strict” est au moins aussi expressif que “non-strict”, ce qui combiné à l’une des séparations précédentes, implique que “strict” est *strictement* plus expressif que “non-strict”. Une autre transformation est la technique de *dilatation* (dilatation, en anglais), qui consiste à transformer n’importe quel graphe dans “non-strict” vers un graphe équivalent dans “propre”. Si l’on réalise que “propre” est un cas particulier de “non-strict” (de même que de “strict”, mais cela ne nous intéresse pas ici), il en résulte que les configurations “propre” et “non-strict” ont exactement la même expressivité, ce qui est inattendu. La troisième transformation, de portée différente, permet de reproduire *dans une certaine mesure* l’accessibilité de n’importe quel graphe temporel vers la configuration la moins expressive – celle des graphes “happy”. Étant donné un graphe temporel \mathcal{G} d’accessibilité $closure(\mathcal{G})$, l’idée consiste à construire un graphe \mathcal{H} *plus grand* que \mathcal{G} , dans lequel un sous-ensemble bien défini de sommets auront, *entre eux*, exactement la même accessibilité que les sommets de \mathcal{G} (on parle d’accessibilité *induite*). Techniquement, la transformation repose sur un gadget temporel appelé *sémaphore*, qui s’inspire d’une réduction de NP-difficulté issue de [BF03]. Cette troisième transformation montre que les graphes “happy”, bien qu’étant les moins expressifs, peuvent néanmoins capturer, de manière induite, l’expressivité de toutes les autres configurations.

4 Synthèse et questions ouvertes

Si l’on combine les séparations mentionnées plus haut et les deux premières transformations, ainsi que plusieurs inclusions triviales, nous obtenons un paysage complexe qui démontre que les trois paramètres considérés dans ce travail ont une incidence réelle sur l’expressivité d’un graphe temporel. L’ensemble de nos résultats est présenté à la Figure 3.

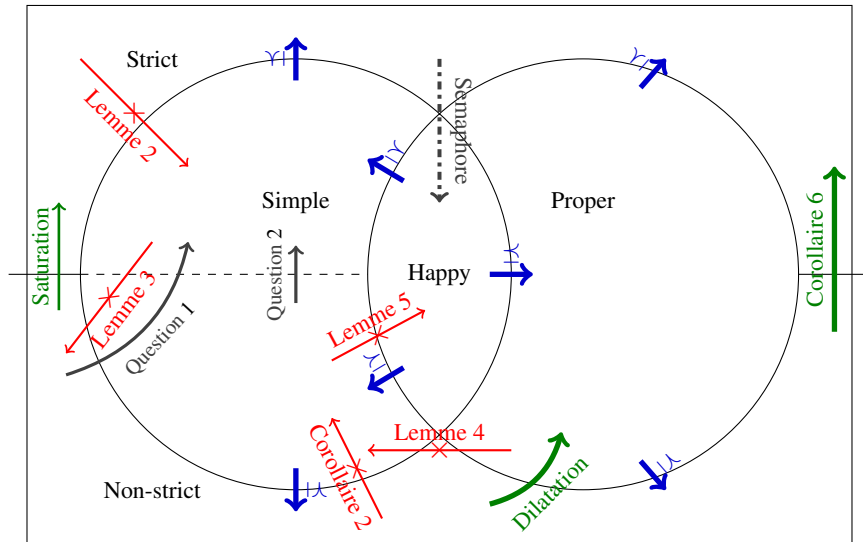


FIGURE 3 : Synthèse des résultats de la version complète [CCS22].

Dans ce paysage, le statut de certaines relations reste à déterminer. Notamment :

Question 1. Est-ce que “simple & strict” contient l’expressivité de “non-strict” ?

Question 2. Sinon, est-ce que “simple & strict” contient l’expressivité de “simple & non-strict” ?

Vus globalement, ces résultats définissent un ordre partiel sur l’expressivité des six configurations, illustré à la Figure 4. Comme déjà évoqué, l’une des surprises de cette étude est que les graphes propres sont aussi expressifs que n’importe quel graphe où les trajets non-stricts sont autorisés, ce qui légitimise leur étude.

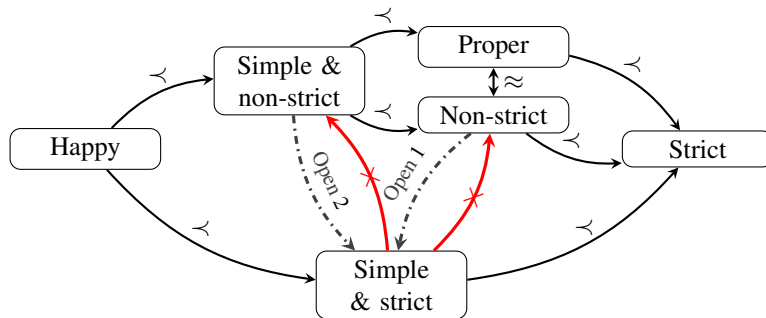


FIGURE 4 : Relation d’ordre entre les configurations de graphes temporels selon leur expressivité.

Au delà de cette classification, une question intéressante serait de caractériser les familles de graphes orientés qui correspondent à ces configurations.

Références

- [BF03] Sandeep Bhadra and Afonso Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *International Conference on Ad-Hoc Networks and Wireless*, pages 259–270. Springer, 2003.
- [CCS22] Arnaud Casteigts, Timothée Corsini, and Writika Sarkar. Simple, strict, proper, happy : A study of reachability in temporal graphs. In *Stabilization, Safety, and Security of Distributed Systems : 24th International Symposium, SSS 2022, Clermont-Ferrand, France, November 15–17, 2022, Proceedings*, pages 3–18. Springer, 2022. arXiv :2208.01720.