Un château... pas très fort !!

... une idée farfelue de Marie Duflot-Kremer
créée en juillet 2015

dernière modification le 24 août 2017

Le document que vous êtes en train de consulter n'est pas une référence très finalisée, ni un guide strict à suivre. Il regroupe des infos que vous pourriez trouver utiles si vous envisagez d'animer cette activité. Il contient une mise en situation, le déroulé de l'activité, l'explication côté informatique ainsi que des documents à imprimer pour réaliser le matériel de l'activité. La diffusion de ce document est libre, vous pouvez suggérer des améliorations/enrichissements à `marie.duflot-kremer@loria.fr`. Si ce document vous a été utile, vous pouvez également me le signaler car j'envisage de mentionner sur ma page médiation les écoles/associations/etc. qui ont testé et approuvé l'activité. La page `https://members.loria.fr/MDuflot/` permet de trouver (section médiation/activités) une liste d'autres activités pour faire découvrir différents aspects de l'informatique, réalisables en grande majorité sans ordinateur.

L'idée de cette activité est née quand j'ai réalisé que je faisais pas mal d'activités de médiation en informatique, mais qu'aucune n'avait de lien direct avec mon thème de recherche : la vérification formelle de systèmes, autrement dit la conception d'algorithmes et de programmes permettant de s'assurer qu'un système informatique (une machine à café, le protocole Bluetooth, un ascenseur, ...) fait bien ce qu'on attend de lui. J'ai donc conçu cette activité qui consiste à faire comprendre et expérimenter à des lycéens... mon cours de Master 2ème année. Et, même si je n'étais pas totalement sûre de moi, ça a super bien marché.

Étape 1 — Le public

Contrairement aux autres activités proposées sur ma page, cette activité est plutôt recommandée pour les élèves de lycée (testée en partie avec des élèves de troisième (14-15 ans) à la Fête de la Science, mais je pense qu'ils n'auraient pas forcément pu aller au bout de tous les aspects de l'activité). Elle a été testée plusieurs fois avec grand succès en classe de 1ère Scientifique (16-17 ans).

Étape 2 — Le contexte

Vous avez en fin de ce document le plan d'un chateau. Il n'est pas tout à fait classique et pas du tout à l'échelle, mais chaque cercle représente une salle du château, et chaque flèche entre deux salles signifie qu'on peut aller de la première salle à la deuxième.

Bon d'accord en général quand on peut aller d'une salle à une autre c'est qu'il y a une porte, et du coup on peut revenir dans l'autre sens. Seulement Eusebius, l'architecte de ce château a été farceur et a créé plein de portes et de passages secrets que pour la plupart on ne peut ouvrir ou activer que d'un côté, comme si la porte n'avait une poignée que d'un seul côté. Il y a même dans une salle en haut à gauche un passage secret qui part d'une pièce et nous ramène (après un tour sur nous-même ?) dans la même pièce.

Notre mission ? Ecrire des propriétés sur le château, par exemple "on peut atteindre une fenêtre" ou "on va forcément passer devant un tableau" dans un langage particulier appelé *langage logique* ou tout simplement *logique*, où les mots ont une signification très précise. Ensuite nous pourrons voir si le château (ou une salle particulière du château) satisfait chacune de ces propriétés.

Oui bon ça marche, on arrive presque sans se tromper à répondre à ces questions. Mais si la propriété est plus compliquée, ou si le château est plus grand ? On fait comment ? Et bien justement, “on” ne fait plus, c'est l'ordinateur qui le fait. Seulement pour cela il faut lui expliquer comment faire, et c'est le but de la dernière étape de l'activité.

Étape 3 — Préparer le matériel

Pour réaliser l'activité, il faut imprimer ou refaire soi-même le château (y compris les petits dessins près de chaque salle). Il faut également reproduire ou imprimer la liste des opérateurs (les petits dessins) et les découper (un dessin par petit bout de papier).

Comme on va avoir besoin d'écrire sur le château, il est fortement recommandé de le plastifier et d'utiliser des feutres pour tableau blanc/ardoise, sinon après avoir gommé 5 ou 6 fois on risque de passer au travers du papier.

Étape 4 — Apprivoiser les opérateurs

La première étape est d'utiliser les petits dessins pour écrire des propriétés de châteaux. On distribue à chaque groupe de deux participants un lot de dessins (qu'on appelle des opérateurs) et on les aide à se familiariser avec, en s'aidant de l'aide mémoire. Ces opérateurs se classent en plusieurs catégories :

1. Les opérateurs qui disent ce qui est vrai dans une salle. Il y a les torches, les trésors, les fenêtres et les tableaux. L'opérateur tableau par exemple signifie qu'il y a un tableau dans la pièce où on se trouve. On a aussi des opérateurs avec les mêmes objets barrés, qui signifient qu'il n'y a PAS de torche (respectivement de trésor, de fenêtre, de tableau) dans la pièce.
2. Les opérateurs logiques classiques : ET, OU, NON, $\Rightarrow$, dont les tables de vérité sont données en figure 1 qui ont le sens suivant :
   ET : la formule A ET B est vraie, quand A et B sont vrais tous les deux.
   OU : la formule A OU B est vraie quand au moins un des deux parmi A ou B est vrai.
   $\Rightarrow$: ce signe représente l'implication. A $\Rightarrow$ B pourrait se traduire par “si A est vrai alors B est vrai aussi”. Cette formule est fausse quand A est vrai mais pas B, et vraie dans tous les autres cas. En particulier elle est vraie quand A est faux, peu importe la valeur de B.
   NON : la formule NON A signifie : “il n'est pas vrai que A”. Elle est donc vraie quand A est faux, et fausse quand A est vrai.

A	B	A ET B	A OU B	A $\Rightarrow$ B	NON A	NON B
Vrai	Vrai	Vrai	Vrai	Vrai	Faux	Faux
Vrai	Faux	Faux	Vrai	Faux	Faux	Vrai
Faux	Vrai	Faux	Vrai	Vrai	Vrai	Faux
Faux	Faux	Faux	Faux	Vrai	Vrai	Vrai

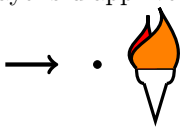
FIGURE 1 – Table de vérité des opérateurs logiques habituels

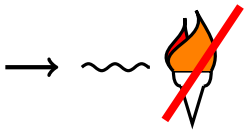
3. Les deux opérateurs comportant une ou plusieurs flèches permettent de parler des chemins qui partent d'une pièce. Celui avec une flèche dit qu'il existe un chemin à partir de cette pièce tel que .... (et il faut écrire la suite de la formule avec d'autres opérateurs). Celui avec plusieurs flèches dit que pour tout chemin qui part de cette pièce on a ... (et pareil la suite reste à écrire)
4. Les quatres derniers opérateurs servent à dire ce qu'il se passe le long d'un chemin.
   — L'opérateur “un jour” • dit qu'à un moment (= un jour) le long de ce chemin, quelque chose (qu'on va écrire avec nos opérateurs) va être vrai.
   — L'opérateur “toujours” ~~ dit qu'à toutes les étapes de ce chemin, quelque chose (qu'on va écrire avec nos opérateurs) va être vrai.
   — L'opérateur “dans la prochaine pièce” –• dit que le long de ce chemin, dans la prochaine pièce (celle qu'on va atteindre juste après) quelque chose va être vrai
   — enfin l'opérateur “jusqu'à” ~~• affirme qu'une chose est vraie jusqu'à ce qu'une autre devienne vraie.

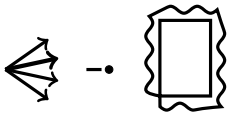
5. Il y a aussi des parenthèses pour dire dans quel ordre appliquer les opérateurs. Par exemple : Un jour (fenêtre ET torche) est différent de (Un jour fenêtre) et torche. Dans le deuxième cas, la torche ce n'est pas un jour qu'on doit l'avoir mais tout de suite.

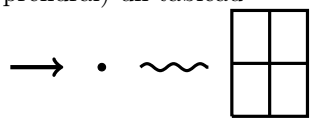
Remarques : Attention, si on ne met aucun opérateur de chemin (ceux décrits dans le point 4. ci dessus) alors on parle de ce qu'il se passe en ce moment. De plus on ne s'intéresse avec cette logique qu'à des chemins qui ne s'arrêtent jamais. Donc quand je dis "toujours" je n'ai pas le droit de m'arrêter dans la pièce qui me plait et dire c'est bon j'ai toujours une fenêtre, je dois continuer à franchir des portes (mais si la porte me ramène dans la même pièce là j'ai le droit de la prendre encore et encore tout en restant dans la même pièce).

Essayons d'apprivoiser ces formules par quelques exemples :

—  : il existe un chemin qui me mène devant une torche (un jour)

—  : il existe un chemin le long duquel je suis toujours dans une pièce où il n'y a pas de torche, en d'autres mots il existe un chemin qui évite les torches (qui n'en croise aucune)

—  : dans la prochaine pièce j'aurais forcément (quelle que soit la porte que je prendrai) un tableau

—  : il existe un chemin le long duquel à partir d'un moment (= un jour) j'ai toujours des fenêtres. En clair sur ce chemin peut être qu'au début je n'ai pas de fenêtre, mais à partir d'un moment j'en verrai dans toutes les pièces que je croiserai.

Étape 5 ____ A la recherche des formules magiques, euh logiques.

Maintenant qu'on a un peu manipulé les opérateurs, on peut donner des phrases aux participants et leur demander de trouver les formules qui vont avec. Voici une liste de propriétés possibles. Bien évidemment je ne les fais pas toutes, je pioche selon mes envies.

1. Je peux accéder à un trésor
2. Je peux atteindre une pièce ou je peux me remplir les poches (de trésor) tout en admirant un tableau.
3. Je peux aller dans une pièce où je pourrai correctement admirer un tableau (pour l'admirer il faut une source de lumière, il y en a deux possibles).
4. Toute ballade dans le château permet de voir au moins une fenêtre.
5. Je peux me ballader dans uniquement des pièces avec des torches.
6. Je vais forcément me ballader dans uniquement des pièces avec des torches.
7. Je peux arriver devant un tableau en exactement 3 déplacements.
8. Je peux arriver devant un tableau en 3 déplacements maximum.
9. Quoi qu'on fasse, on verra toujours une torche, un portraît ou une fenêtre dans sa pièce.
10. Je ne peux pas me retrouver devant un tableau sans torche.
11. Si je me promène dans le château, à partir d'un moment je ne verrai plus jamais de tableau.
12. Il existe un chemin (infini) qui ne passe jamais par le trésor
13. Je peux me ballader et ne plus voir de torches après un certain moment

14. Quoi qu'on fasse, après deux déplacements on est forcément dans une pièce avec une torche.
15. Il y a moyen de se perdre dans le château dans un endroit à partir duquel je ne peux plus atteindre le trésor.
16. Quelqu'un qui a peur du noir peut atteindre le trésor (les gens qui ont peur du noir ne peuvent aller dans les pièces où il n'y a ni torche ni fenêtre).
17. Si j'arrive dans une pièce avec un tableau alors deux pièces plus tard je verrai forcément une fenêtre.
18. Je peux atteindre le trésor en passant d'abord par des salles ayant toutes un tableau, puis des salles ayant toutes une torche, puis le trésor.

Quand on fait cet exercice il faut savoir deux choses :
— d'une part il existe plusieurs formules correctes pour une même question, on dit qu'elles sont équivalentes
— d'autre part la façon classique de prouver qu'une formule est fausse c'est de trouver un château qui satisfait notre propriété (écrite en français) mais pas la formule logique, ou inversement, la formule mais pas la phrase en français. Cela demande un peu d'imagination, et aussi de d'abord tomber d'accord sur ce que veut dire la phrase en français. Et si l'on vous dit que les phrases françaises sont ambigües, c'est super, ça fait un argument de poids pour justifier d'utiliser des langages logiques précis.

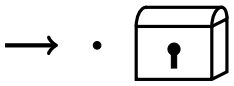
Étape 6 — Comment vérifier les formules ?

Les formules ci-dessus ne sont pas toutes vraies dans mon château, du moins pas à partir de la pièce de départ. Un réflexe des participants est, dès qu'on leur donne une propriété, de regarder à la main si elle est vraie sur notre château. Cela fait une excellente transition vers cette partie.

Tout d'abord on va justement vérifier à la main : est-ce que la pièce de départ satisfait notre propriété ? et si non y a-t-il une pièce du château qui la satisfait ? et si encore non peut-on imaginer un château pas complètement trivial qui satisfait la propriété ?

Ensuite il faut s'imaginer que les propriétés peuvent être un peu plus compiquées que celles que l'on a ici, mais surtout que les systèmes à vérifier sont **beaucoup** plus grands que notre petit château et sa dizaine de pièces. En pratique on peut avoir des millions d'états (nos pièces), voire plus. Du coup le faire à la main ça n'est plus crédible du tout : on n'a pas le temps, même pour une quarantaine de pièces, et on a toutes les chances de se tromper.

La formule que j'utilise pour rechercher un algorithme de vérification est la suivante :



Il existe un chemin le long duquel un jour je rencontre un trésor. On commence par se demander s'il n'y aurait pas par hasard des pièces qui satisfont cette propriété et pour lesquelles on peut le voir facilement.

Et effectivement il y en a : les pièces qui contiennent un trésor satisfont cette propriété car il existe bien un chemin (je prends celui que je veux, peu importe) le long duquel un jour (et plus précisément maintenant là tout de suite) j'ai un trésor. On va donc, sur notre feuille plastifiée, mettre un point dans ces pièces là.

Maintenant on va réfléchir à comment continuer. N'y a-t-il pas d'autres pièces pour lesquelles on peut dire simplement qu'à partir d'elles on peut aller dans les pièces à trésor ? Et oui c'est le cas : toutes les pièces qui ont une flèche qui amène dans une pièce à trésor ont bien un chemin qui un jour amène au trésor. On ve donc pouvoir leur mettre un point aussi. Et on continue, maintenant toutes les pièces qui en une étape nous amènent dans une pièce à point peuvent atteindre le trésor, et en continuant on va toutes les trouver.

Pour formaliser un peu, on va en fait appliquer l'algorithme suivant (à faire trouver, avec aide si besoin, par les participants) :

Tant que *j'ai des pièces avec un point dedans* **Faire**
    Choisir une pièce **p** avec un point dedans
    Remplacer le point dans **p** par une croix
    `# pour dire qu'on s'en est occupé`
    Pour *chaque pièce qui a une porte qui arrive dans* **p Faire**
        Si *il n'y a ni point ni croix dans cette pièce* **Alors**
            Y mettre un point
            `# S'il y avait déjà quelque chose on sait que la pièce ne sera pas oubliée`
        Finsi
    Finpour
Fintantque

Déjà cet algorithme se termine car grâce à nos points et nos croix on met maximum une fois un point dans une pièce, et une fois qu'on traîte la pièce on enlève le point (pour ne pas la reprendre une autre fois). De plus quand il se termine on n'a plus de point (sinon on ne sortirait pas de la boucle Tant que), mais juste des pièces avec des croix (celles qui satisfont la propriété), et d'autres sans rien (celles qui ne satisfont pas la propriété).

Et il existe des algorithmes similaires, comme pour "Pour tout chemin toujours fenêtre", et d'autres un peu plus difficiles pour des formules comme "Pour tout chemin un jour trésor" ou "Il existe un chemins sur lequel j'ai une torche jusqu'à ce que je rencontre une fenêtre" que l'on peut aborder si on a vraiment du temps en plus et des gens intéressés.

Étape 7 — Et hop, c'est de l'informatique

Oui bon en cherchant un algorithme qui nous permette de trouver toutes les salles qui satisfont une propriété on se doute bien qu'on est un peu en train de faire de l'informatique. Seulement cet algorithme je ne l'ai pas inventé pour vous. Les formules non plus.

Tout d'abord le dessin du château est une façon formelle de représenter un système. On peut donner un sens à ces cercles et ces flèches et une fois le dessin réalisé, on peut dire précisément les déplacements possibles dans le château. Notre combinaison de cercles pour les salles, de flèches pour les passages entre deux salles et de petits dessins disant ce qu'on peut trouver dans la salle existent en informatique. Le type de modèle est une *structure de Kripke* (oui ça vous fait une belle jambe), les cercles sont des *états*, les flèches sont des *transitions* et les dessins des *propositions atomiques* (en gros des formules logiques les plus simples possibles pour lesquelles on peut immédiatement dire si elles sont vraies dans un état ou pas).

Ensuite les propriétés de ces systèmes s'expriment à l'aide d'opérateurs logiques, exactement comme on l'a fait avec les petites étiquettes. Les opérateurs s'appellent E (there Exists a path/il existe un chemin) A (for All paths/pour tout chemin) F (Finally/un jour) X (neXt/après une transition) G (Globally/toujours) et U (Until/jusqu'à ce que) au lieu d'être des petits dessins comme on l'a vu pour le château, mais vous avez découvert tous les opérateurs qui permettent de définir la logique CTL (Computation Tree Logic), et utilisé les véritables algorithmes qui permettent de s'assurer qu'une formule (simple) de CTL est vraie.

Comme on l'a vu dans l'activité, il existe des algorithmes qui permettent de trouver tous les états qui satisfont une formule de CTL. Il suffit alors de regarder si l'état de départ en fait partie pour dire avec certitude si notre système satisfait une propriété ou non. Et cette méthode qui consiste à construire un modèle formel d'un système, à exprimer formellement également des propriétés attendues ou redoutées pour le système puis à lancer un programme qui nous dit automatiquement et avec certitude si le modèle satisfait la propriété s'appelle le Model Checking (vérification de modèles).

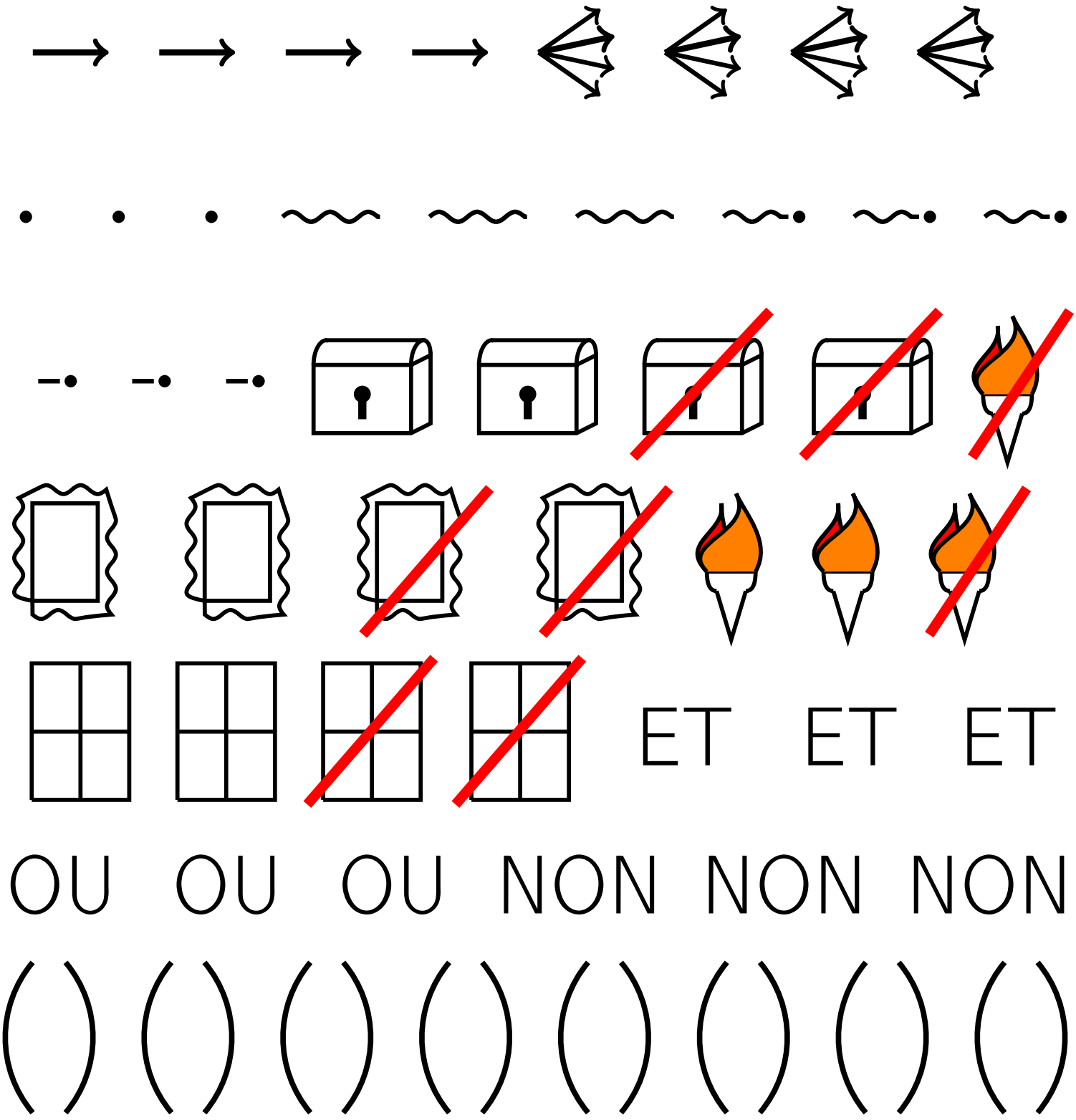
Et si on y réfléchit bien notre méthode sur le château est vraiment efficace, car en regardant peu de fois chaque état et maximum une fois chaque transition du système, on peut en un temps fini s'assurer de propriétés d'un système qui a un nombre infini de chemins (on peut boucler sur les états qu'on veut dans l'ordre qu'on veut et on ne s'arrête pas).

Par contre cette méthode nécessite de construire un modèle formel de notre système (ici le château), et cela veut aussi dire que si on se trompe en créant le modèle, on aura vérifié un modèle qui ne sert à

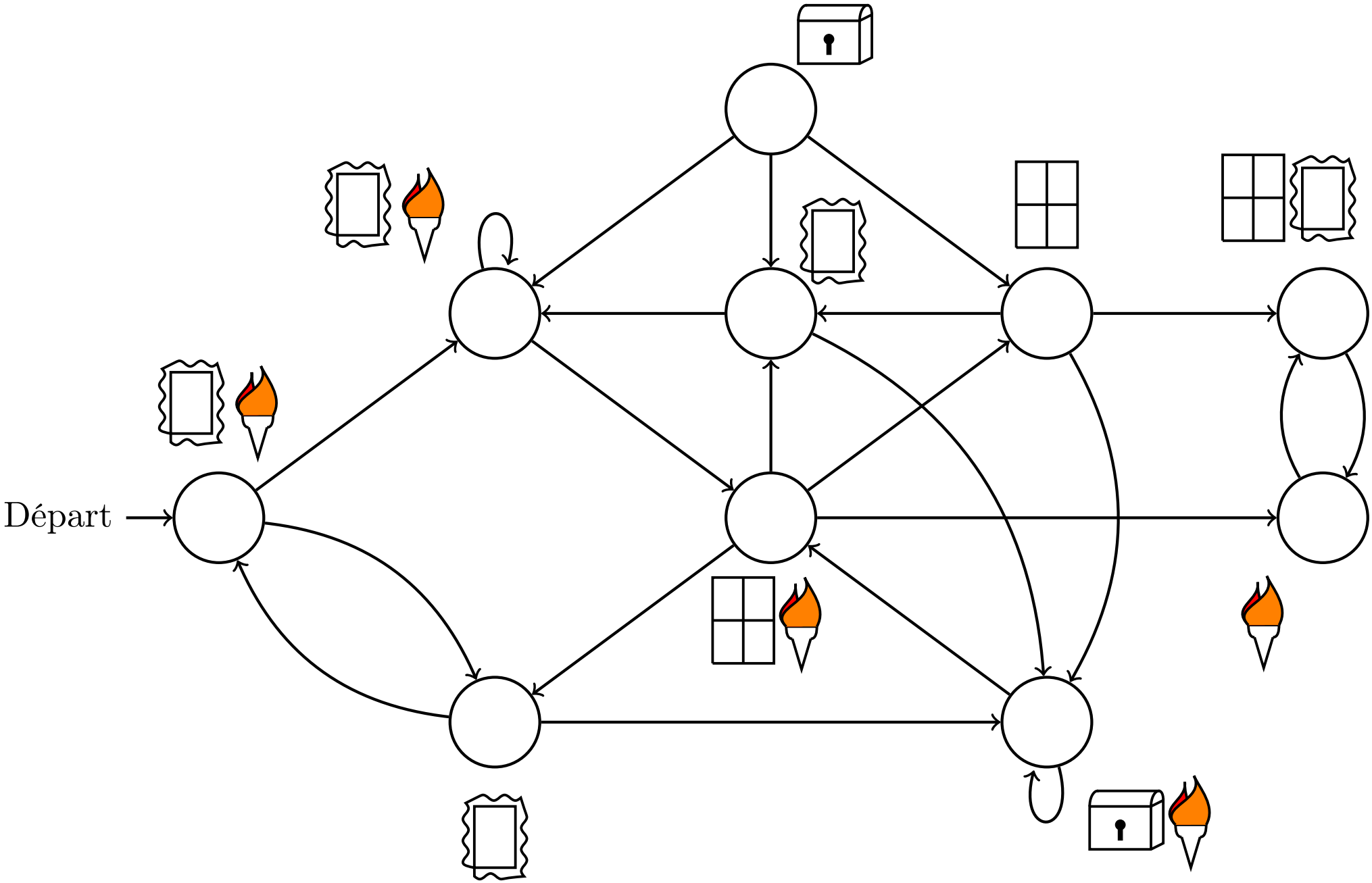
rien. On a donc encore besoin d'humains pour créer ce modèle, ainsi que pour écrire la propriété comme une formule logique. Le reste par contre est fait par des logiciels spécialisés.

Étape 8 — Annexes

La suite contient le plan du château à imprimer et plastifier, ainsi que des opérateurs à découper et un pense bête pour se souvenir du rôle des opérateurs un peu exotiques



→	: il existe un chemin tel que...
⇜	: pour tout chemin on a...
.	: un jour le long de ce chemin...
~	: dans toutes les pièces de ce chemin...
-.	: dans la pièce suivante...
~.	: le long du chemin on a... jusqu'à ce qu'on aie...


Départ