contraire et il ne connaît pas au moins trois d'entre elles (*cas 2*).

Cas 1. Alain connaît au moins trois autres personnes. Examinons alors les relations que peuvent entretenir les trois connaissances d'Alain entre elles. Si deux d'entre elles se connaissent, alors ces deux personnes et Alain forment un ensemble de trois personnes qui se connaissent. On a obtenu ce qu'on attendait. On peut donc supposer que les trois connaissances d'Alain sont étrangères l'une à l'autre. Dans ce cas, elles constituent trois personnes de l'assemblée qui ne se connaissent pas, on a donc encore obtenu ce qu'on voulait.

Cas 2. Alain ne connaît pas au moins trois des autres personnes. On raisonne comme dans le premier cas, en inversant «connaître» et «ne pas connaître».

Pour voir que 5 personnes ne sont pas suffisantes pour que la propriété «3 se connaissent ou 3 ne se connaissent pas» soit toujours vérifiée, il suffit de considérer la situation suivante: A connaît B, qui connaît C, qui connaît D, qui connaît E, qui connaît A et rien d'autre.

ÉNIGME 4: RENOMMER LES NŒUDS D'UN GRAPHE

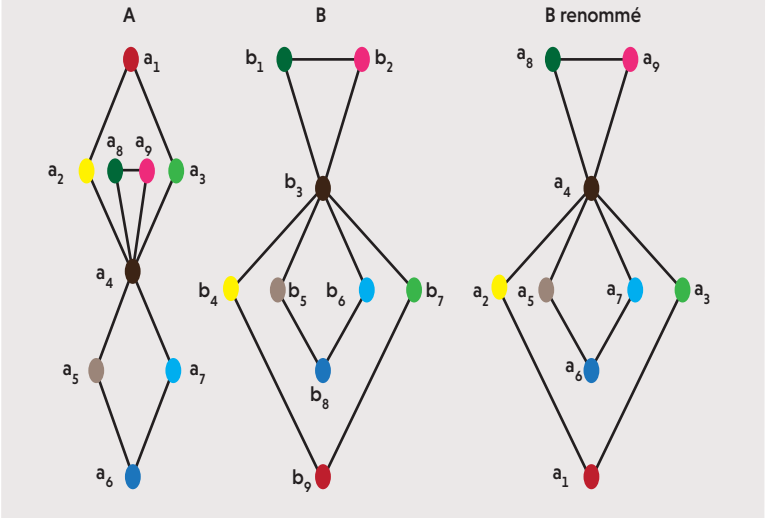
Je dispose de deux très grands graphes notés A et B. Je sais qu'ils sont identiques aux appellations des nœuds près, c'est-à-dire qu'en changeant les noms des nœuds du second par des noms de nœuds du premier et en les plaçant correctement, alors les arêtes des deux graphes sont les mêmes.

Dit autrement, si les arêtes étaient de parfaits élastiques et si je pouvais déplacer les nœuds du second graphe, je pourrais faire coïncider parfaitement les deux graphes (*voir la figure 4*). Lorsque les graphes sont très grands, ces mises en coïncidence sont difficiles à trouver, ce qui fait que l'on peut utiliser la connaissance d'une telle mise en coïncidence comme méthode d'authentification.

Supposons que j'aie eu beaucoup de mal à trouver cette mise en coïncidence par changement des noms, et que mon ami Jacques, qui trouve intéressant le changement des noms qui provoque la mise en coïncidence, souhaite me l'acheter. Il n'a pas confiance en moi car je l'ai escroqué dans une précédente affaire. Il veut que je lui prouve que je dispose de cette mise en coïncidence avant de me l'acheter. Je n'ai pas non plus confiance en lui car lui aussi m'a escroqué récemment (nous ne sommes d'ailleurs plus vraiment amis...). Je ne veux donc pas lui communiquer la moindre information sur la mise en coïncidence dont je dispose car cela pourrait l'aider à la trouver et il ne me l'achèterait pas. Je veux en revanche lui prouver que je connais ce changement des noms du graphe B. Comment mener une telle opération qui semble logiquement impossible?

4

MISE EN COÏNCIDENCE DE GRAPHES



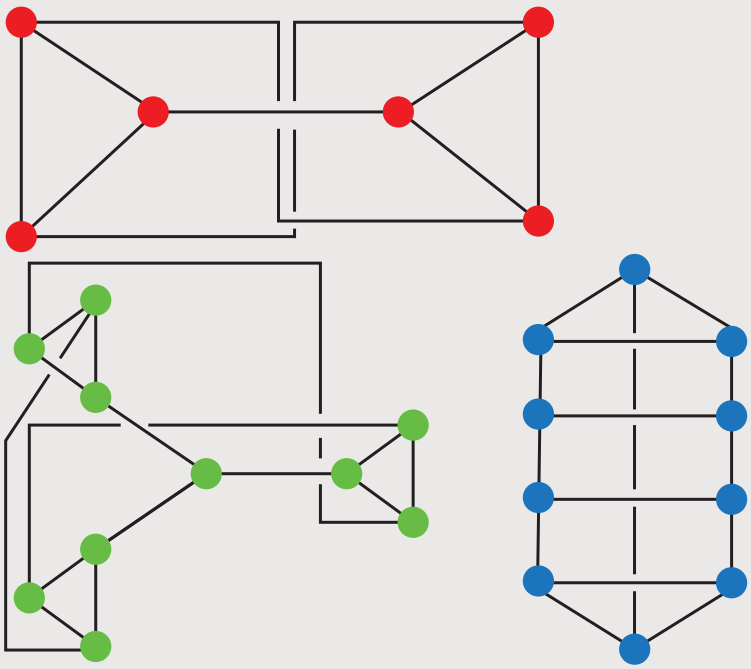
Solution: Le problème est classique et sert souvent comme exemple de «preuve à divulgation nulle de connaissance», un type de protocole utile en cryptographie. Quand deux graphes sont identiques aux noms près des nœuds, on dit qu'ils sont isomorphes, mais nous continuerons à parler simplement de «mise en coïncidence». Voici comment je persuade Jacques.

(a) Je dessine un autre graphe C en changeant les noms des nœuds de A, avec des noms différents de ceux utilisés pour A et B. Je fais bien attention de refaire le dessin en déplaçant les nœuds pour que Jacques ne puisse pas tirer d'informations de la disposition de C. Je le présente à Jacques.

(b) Jacques m'indique alors librement A ou B (éventuellement, il tire à pile ou face pour choisir).

(c) S'il m'indique A, je lui fournis les changements des noms qui transforment C en A. Cela m'est facile, puisque c'est moi qui ai construit C à partir de A. Jacques vérifie que ce changement des noms transforme bien C en A.

(d) S'il m'indique B, alors je lui indique un changement des noms des nœuds de C qui le transforme en B. J'obtiens ce changement en changeant d'abord les noms de C en ceux de A (par exemple c_1 devient a_8, etc.), puis en utilisant la mise en coïncidence que j'ai calculée et que je veux lui vendre pour changer les noms de A en ceux de C (a_8 devient par exemple b_7, etc.). Je lui indique le résultat final de ce double changement des noms (c_1 devient b_7, etc.) qui >



GRAPHES LABYRINTHIQUES

5

> bien sûr définit une mise en coïncidence parfaite entre C et B. Jacques vérifie que c'est bien une mise en coïncidence.

(e) En faisant cela une seule fois, quelqu'un ayant de la chance, mais qui ne connaît pas le bon changement des noms entre A et B, peut répondre de façon satisfaisante: il choisit C en décidant de le mettre en coïncidence avec A et Jacques par chance demande A, ou il choisit C en le mettant en coïncidence avec B et par chance Jacques demande B. Avec une probabilité de 1/2, quelqu'un ne connaissant pas le bon changement de noms entre A et B trompera Jacques.

(f) En revanche, si on opère deux fois ce jeu et que je ne connais pas le bon changement des noms entre A et B, je ne tromperai Jacques qu'une fois sur quatre (il faut que deux fois de suite je gagne à un jeu où j'ai une chance sur deux de gagner). Si l'on recommence le jeu 10 fois, je tromperai Jacques avec une probabilité 1/1024 ($1024=2^{10}$). Etc.

(g) En opérant le jeu autant que Jacques le souhaitera et en réussissant à chaque fois, Jacques finira par être certain que je dispose bien d'un bon changement de noms entre A et B.

(h) Pourtant, même en faisant le jeu 1000 fois, il n'aura rien appris, car les graphes C que j'utilise à chaque fois et pour lesquels je réponds ne lui apprennent qu'un rapport entre A et C ou entre B et C, ce qui ne dit rien sur les rapports entre A et B.

Jacques se décidera à m'acheter ma mise en coïncidence de A et B.

Ce type de protocoles sert à organiser des systèmes d'authentification. Si je choisis moi-même les deux grands graphes A et B en les prenant isomorphes et en les rendant publics, je pourrai m'authentifier vis-à-vis de tout interlocuteur sans jamais affaiblir la méthode, puisque les échanges envisagés dans le protocole ne laissent échapper aucune information sur l'isomorphisme que j'ai choisi, et que je suis le seul à connaître (*voir la bibliographie*).

L'isomorphisme de graphes est un problème délicat dont on a du mal à connaître la classe de complexité. Récemment en 2015, puis en 2017 après s'être rétracté, László Babai, un spécialiste réputé de ces questions, a annoncé disposer d'une démonstration que le problème de l'isomorphisme de graphes était à peu de chose près un problème qu'on peut résoudre en temps polynomial, c'est-à-dire assez rapidement. Si c'était confirmé, alors mon ancien ami Jacques risque de ne plus vouloir m'acheter la mise en coïncidence que nous étions en train de négocier.

ÉNIGME 5: NE PAS SE PERDRE DANS UN LABYRINTHE

On se donne un graphe dont chaque nœud possède exactement 3 arcs (ou arêtes) incidents. On pourrait parler de labyrinthe où 3 chemins exactement se rencontrent à chaque carrefour. Le graphe n'est pas nécessairement planaire (représentable sur un plan sans que les arcs se croisent), mais en chaque nœud, les trois arcs incidents sont placés dans un même plan horizontal, de façon que lorsqu'on arrive à un nœud par l'un des arcs, il y a un arc à droite et un arc à gauche clairement identifiables.

Quelques exemples de tels graphes sont donnés par la figure 5. On peut les disposer sur un plan en acceptant quelques ponts ou tunnels. Cela permet, quand on parcourt de tels graphes et qu'on arrive à un carrefour, de désigner sans ambiguïté l'arc de droite et l'arc de gauche. Les stratégies de parcours «prendre toujours à droite» ou «prendre alternativement à droite et à gauche» sont des stratégies qui vous ramènent à votre point de départ. Il faut le démontrer!

Premier problème. Vous êtes posé sur un arc du graphe et vous avancez en utilisant la méthode suivante: à chaque fois que vous arrivez à un nouveau nœud, vous choisissez l'arc de droite. Prouvez qu'en avançant ainsi, sans jamais revenir en arrière, vous retournerez nécessairement à votre arc de départ.

Deuxième problème. Vous êtes posé sur un arc du graphe et vous avancez en utilisant une méthode un peu plus compliquée: à chaque fois que vous arrivez à un nouveau nœud, vous choisissez alternativement l'arc de droite ou l'arc de gauche. Prouvez à nouveau que vous serez inévitablement ramené à votre arc de départ.

Notons que lorsque nous avançons en suivant une méthode déterminée, nous pouvons ne jamais revenir au point de départ et par exemple parcourir des arcs qui, à partir d'un certain moment, forment un cycle a-b-c-d-e-c-d-e-c-d-e-c-d-e... Les questions posées ne sont donc pas évidentes.

Solution: Quand vous suivez la méthode 1 et que vous êtes sur un arc dans une direction donnée, vous savez d'où vous venez: en vous retournant et en allant au nœud dont vous venez, l'arc qui vous y a amené est celui de gauche.

Il n'y a qu'un nombre fini de positions possibles sur un arc du graphe et seulement deux directions possibles à partir d'un nœud, donc il n'y a qu'un nombre fini de parcours d'un arc. Quand vous êtes sur un arc dans une direction donnée, la suite de ce que vous allez faire est déterminée. Il en résulte qu'en appliquant la méthode «toujours à droite», vous finirez par tomber sur un cycle. On est donc dans une situation du type a-b-c-a-b-c-a-b-c... ou du type a-b-c-d-e-c-d-e-c-d-e... Dans ces suites, les lettres minuscules désignent des positions sur un arc, donc un arc avec l'indication d'une direction de parcours.

Deux cas sont possibles: soit on entre tout de suite dans un cycle, soit on y entre après quelques parcours d'arcs où l'on ne repassera plus. Il faut montrer que le second cas est impossible. Si on avait une situation du type a-b-c-d-e-c-d-e-c-d-e-..., cela signifierait qu'il y a deux antécédents à c : de l'arc orienté b on passe à c, et de l'arc orienté e on passe à c, cela en suivant la règle «toujours à droite». C'est impossible puisque, comme on l'a expliqué, on peut savoir d'où l'on vient: ou bien, quand on est en position c, on vient de la position b, ou bien quand on est en position c on vient de la position e. Un seul cas est possible. Puisque la situation où l'on entre dans un cycle après le parcours de quelques arcs par lesquels on ne repassera pas est impossible, cela signifie qu'on entre dès le départ dans un cycle, et donc qu'on revient à son arc de départ avec le même sens de déplacement sur l'arc.

Pour le second algorithme, le raisonnement est assez proche, sauf que cette fois on va considérer les parcours en prenant en compte la liste des arcs et sens de déplacement de l'arc par lesquels on passe, auxquels on ajoute l'information indiquant si l'on doit tourner à droite ou à gauche au prochain nœud. Un parcours sera donc la donnée d'une suite d'arcs orientés (désignés comme précédemment par des minuscules) avec pour chacun l'indication droite ou gauche. Par exemple a/droite—b/gauche—c/droite—d/gauche—...

Quand on est dans une position du parcours conçu de cette façon, on peut revenir en arrière, car l'information supplémentaire droite ou gauche nous indique aussi ce qui s'est passé au nœud précédent et donc quel arc nous y a amenés. Comme dans le raisonnement précédent, il est clair qu'un parcours revient à un moment ou un autre en une position déjà occupée (car il n'y a qu'un nombre fini de positions x/droite y/gauche). Il faut prouver que cela ne peut pas se faire ailleurs qu'en la position de départ. Le raisonnement est le même que précédemment, avec la nouvelle notion de parcours: si on ne revenait pas au point de départ, l'une des positions aurait deux antécédents, ce qui n'est pas possible. Notons que le raisonnement assure qu'on revient à l'arc de départ en le parcourant dans le même sens et avec l'indication droite. Il n'est donc pas exclu qu'on y soit passé avant avec l'indication gauche ou en le parcourant dans l'autre sens, mais, quel que soit le cas, on revient à l'arc de départ.

Le résultat se généralise. Si, toujours avec un graphe dont tous les nœuds possèdent 3 arcs incidents, on applique une méthode de choix entre droite et gauche périodique, par exemple droite/droite/gauche répété, on sera certain de revenir à son point de départ. Pour le démontrer, on ajoute autant d'informations qu'il faut aux éléments de ce qu'on considérera comme des positions dans un parcours de façon à pouvoir revenir en arrière, et on raisonne comme précédemment en montrant qu'une position ne peut pas avoir deux antécédents.

Pour les graphes quelconques, on peut encore généraliser le résultat en procédant comme suit. On suppose que le plus grand nombre possible d'arcs incidents pour un nœud est N. On suppose qu'à chaque nœud, les arcs sont numérotés de 1 à M avec $M \leq N$. On peut d'ailleurs imaginer que cette numérotation est fixée par celui qui suit le parcours quand il arrive à un nœud pour la première fois. On définit la méthode de parcours en se donnant une suite périodique d'entiers entre 1 et N (par exemple 2331423314234314...). On décide de suivre le chemin qui, à chaque nouvel arc rencontré, choisira l'arc correspondant à l'entier K de la suite périodique auquel on est arrivé, cela en respectant la convention suivante: on compte les chemins à partir de celui par lequel on est arrivé, éventuellement en tournant plusieurs fois si l'entier K est supérieur au nombre d'arcs incidents au nœud où l'on se trouve. Par exemple, si on arrive par l'arc 2, qu'il y a 5 arcs arrivant sur le nœud où l'on se trouve et que $K=7$, on repartira par le chemin 2+7=9 qui est le même que l'arc 4, puisqu'on compte les K arcs en tournant: 3, 4, 5, 1, 2, 3, 4.

Une telle méthode, par un argument d'antécédent unique (qui sera applicable si l'on associe suffisamment d'entiers à chaque arc quand on définit la notion de position), vous ramènera toujours à votre arc de départ. De tels algorithmes de parcours engendrent des promenades complexes tout en préservant l'assurance de ne jamais se perdre. ■

BIBLIOGRAPHIE

ÉNIGME 1

P. Winkler, **Mathematical Mind-Benders**, chapitre « The adventures of ant Alice », A K Peters/CRC Press, 2007.

ÉNIGME 2

J.-P. Delahaye, **Le désordre total n'existe pas**, *Pour la Science n° 376*, pp. 86-91, février 2009.

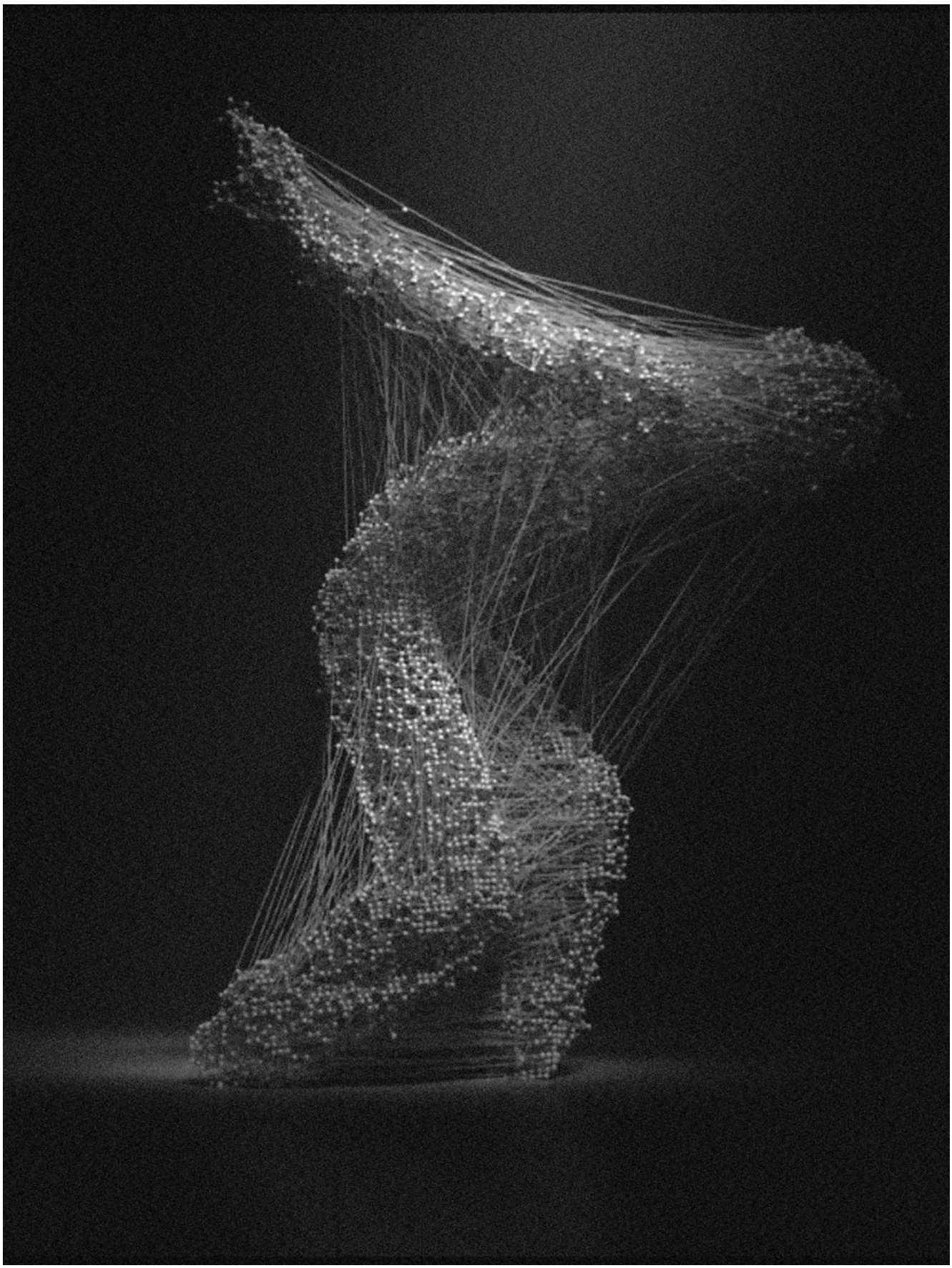
R. Graham *et al.*, **Ramsey Theory**, John Wiley & Sons, 1990.

S. Radziszowski, **Small Ramsey numbers**, 2017 : http://bit.ly/2h26G2o

ÉNIGME 4

J.-P. Delahaye, **Persuader de son savoir sans le transmettre**, *Pour la Science* n° 399, pp. 88-93, janvier 2011.

O. Goldreich, **Probabilistic proof systems : A primer**, *Foundations and Trends in Theoretical Computer Science*, vol. 3(1), pp. 1-91, 2008.



© Pulsar

Danser pour mettre en échec les systèmes de reconnaissance automatique de mouvements suspects.