La complexité des procédures de démonstration de théorèmes

Stephen A. Cook

Université de Toronto

Résumé : On montre que tout problème de reconnaissance résolu par une machine de Turing déterministe en un temps polynomialement borné peut "être réduit" au problème de déterminer si une formule propositionnelle donnée est une tautologie. Ici "réduit" signifie, exprimé grossièrement, que le premier problème peut être résolu de façon déerministe en temps polynomial sous la condition qu'un oracle existe qui peut résoudre le second. À partir de cette notion de réductibilité, des degrés polynomials de difficulté sont définis, et on montre que le problème de déterminer le caractère tautologique a le même degré polynomial que le problème de déterminer si le premier de deux graphes est isomorphe à un sous-graphe du second graphe. On discute d'autres exemples. Une méthode pour mesurer la complexité des procédures de preuve pour le calcul des prédicats est introduite et discutée.

Tout au long de cet article, un *ensemble de chaînes de caractères*[1] signifie un ensemble de chaînes de caractères sur un alphabet fixé, grand et fini Σ. Cet alphabet est suffisamment grand pour inclure des symboles pour tous les ensembles décrits ici. Toutes les machines de Turing sont des dispositifs de reconnaissance déterministes, à moins que le contraire ne soit explicitement spécifié.

1. Tautologies et réductibilité polynomiale.

Fixons un formalisme pour le calcul propositionnel dans lequel les formules sont écrites comme des chaînes de caractères sur Σ. Puisque nous aurons besoin d'un nombre infini de symboles propositionnels (les atomes), chaque tel symbole consistera en un élément de Σ suivi d'un nombre en notation binaire permettant de le distinguer. Par conséquent, une formule de longueur n peut avoir seulement environ $n/\log n$ symboles distincts de fonctions et de prédicats. Les connecteurs logiques sont $\wedge$[2] (et), $\vee$ (ou), et $\neg$ (non).

L'ensemble des tautologies (noté {Tautologies}) est un certain ensemble récursif de chaînes de caractères sur cet alphabet, et nous nous intéressons au problème de trouver une borne inférieure de son temps de reconnaissance possible. Nous ne fournissons pas une telle borne ici, mais je démontrerai que {Tautologies} est un ensemble difficile à reconnaître, puisque de nombreux problèmes apparemment difficiles peuvent être réduits à une détermination du caractère tautologique de leur énoncé. Par *réduits*, nous voulons dire, grosso modo, que si le caractère tautologique pouvait être décidé instantanément (par un "oracle") alors ces problèmes pourraient être résolus en temps polynomial. Pour préciser cette notion, nous introduisons les machines à requêtes, qui sont comme des machines de Turing avec oracles dans [1].

Une *machine à requête* est une machine de Turing multi-rubans avec un ruban particulier appelé le *ruban à requête*, et trois états particuliers appelés *l'état de requête, l'état oui*, et *l'état non*, respectivement. Si M est une machine à requête et T est un ensemble de chaînes de caractères, alors

---

Transcription de l'article original de 1971 tapé à la machine par Tim Rohlfs (rev. 3). J'ai transcrit de façon basique exactement le texte que Cook a écrit ; fréquemment, j'ai même conservé la ponctuation incorrecte. À chaque fois que ma version diffère de celle de Cook, j'ai fourni une explication. Les coquilles typographiques ont été corrigées sans justification.

Traduction en français : Denise Vella-Chemla, novembre 2021.

1. Cook souligne les phrases sur lesquelles il souhaite mettre l'accent. J'utiliserai les lettres en italique dans ce but.
2. Cook utilise & ("et") à la place de $\wedge$. Pour une meilleure lisibilité, j'utiliserai $\wedge$, qui est l'usage commun.

un *T-calcul* de M est un calcul de M dans lequel initialement M est dans l'état initial et reçoit en entrée une chaîne de caractères w sur son ruban d'entrée, et à chaque fois que M est supposée être dans l'état de requête, il y a une chaîne de caractères u sur le ruban de requête, et le prochain état M est supposé être dans l'état oui si $u \in T$ et dans l'état non si $u \notin T$. On pense à un "oracle", qui connaît T, et qui place M dans l'état oui ou dans l'état non.

Définition. *Un ensemble S de chaînes de caractères est* P-réductible *(*P *pour polynomial) à un ensemble T de chaînes de caractères si et seulement s'il y a une machine à requête M et un polynôme $Q(n)$ tel que pour chaque chaîne de caractères en entrée w, le T-calcul de M avec w en entrée s'arrête en $Q(|w|)$ étapes ($|w|$ est la longueur de w) et s'arrête dans un état acceptable si et seulement si $w \in S$.*

Il n'est pas difficile de voir que la P-réductibilité est une relation transitive. Par conséquent, la relation E sur les ensembles de chaînes de caractères, définie par $(S, T) \in E$ si et seulement si les ensembles de chaînes de caractères S et T sont P-réductibles l'un à l'autre et réciproquement, est une relation d'équivalence. La classe d'équivalence contenant un ensemble S sera notée $\deg(S)$ (le degré polynomial de difficulté de S).

Définition. *Nous noterons $\mathscr{L}_*$ le degré* $\deg(\{0\})$*, où 0 dénote la fonction nulle.*

Donc $\mathscr{L}_*$ est la classe des ensembles reconnaissables en temps polynomial. $\mathscr{L}_*$ est étudié dans [2], p. 5, et est l'analogue en termes de chaînes de caractères de la classe des fonctions $\mathscr{L}$ de Cobham [3] [3].

Nous définissons maintenant les ensembles particuliers suivants de chaînes de caractères.

1. *Le problème du sous-graphe* est le problème qui consiste à déterminer, étant donnés deux graphes non orientés finis, si le premier est isomorphe à un sous-graphe du second. Un graphe G peut être représenté par une chaîne de caractères $\overline{G}$ sur l'alphabet $\{0, 1, *\}$ en listant les lignes successives de sa matrice d'adjacence, séparées par des $*$. Nous notons {Paires de sous−graphes} l'ensemble des chaînes de caractères $\overline{G_1} * *\overline{G_2}$ tel que G_1 est isomorphe à un sous-graphe de G_2.
2. Le problème de l'*isomorphisme de graphes* sera représenté par l'ensemble, noté {Paires de graphes
isomorphes}, de toutes les chaînes de caractères $\overline{G_1} * *\overline{G_2}$ telles que G_1 est isomorphe à G_2.
3. L'ensemble {Nombres premiers} est l'ensemble de toutes les notations binaires des nombres premiers.
4. L'ensemble {Tautologies FND} est l'ensemble des chaînes de caractères représentant des tautologies en forme normale disjonctive.
5. L'ensemble D_3 est constitué de ces tautologies en forme normale disjonctive dans lesquelles chaque élément de la disjonction contient au plus trois éléments par conjonction (chacun de ces éléments est soit un atome, soit la négation d'un atome).

Théorème 1. *Si un ensemble S de chaînes de caractères est accepté par une certaine machine de*

3. Le papier mentionne de façon erronée le nom "Cabham".

Turing non-déterministe en temps polynomial, alors S est P*-réductible à* {*Tautologies FND*}.

Corollaire. *Chacun des ensembles dans les définitions 1)-5) est* P*-réductible à* {*Tautologies FND*}.

Cela est dû au fait que chaque ensemble, ou son complémentaire, est accepté en temps polynomial par une certaine machine de Turing non déterministe.

Preuve du théorème. Supposons qu'une machine de Turing non déterministe M reconnaisse un ensemble S de chaînes de caractères en un temps $Q(n)$, où $Q(n)$ est un polynôme. Étant donnée une chaîne de caractères en entrée w pour M, nous construirons une formule propositionnelle $A(w)$ en forme normale conjonctive telle que $A(w)$ est satisfiable si et seulement si M reconnaît w. Donc $\neg A(w)$ est facilement mis en forme normale disjonctive (en utilisant les lois de De Morgan), et $\neg A(w)$ est une tautologie si et seulement si $w \notin S$. Puisque la construction complète peut être effectuée en temps borné par un polynôme en $|w|$ (la longueur de w), le théorème est démontré.

Nous pouvons également supposer que la machine de Turing M a seulement un ruban, qui est infini à droite mais a une case la plus à gauche. Numérotons les cases de la gauche vers la droite 1,2,.... Fixons une chaîne de caractères en entrée w pour M de longueur n, et supposons que $w \in S$. Alors il y a un calcul de M d'entrée w qui se termine dans un état de reconnaissance en $T = Q(n)$ étapes. La formule $A(w)$ sera constituée de nombreux symboles propositionnels différents, dont le sens intentionnel, listé ci-dessous, se réfère à un tel calcul.

Supposons que l'alphabet du ruban pour M est $\{\sigma_1, \ldots, \sigma_l\}$ et que l'ensemble des états est $\{q_1, \ldots, q_r\}$.[4] Noter que puisque le calcul a au plus $T = Q(n)$ étapes, aucune case de ruban au-delà de T n'est scannée.

Symboles propositionnels :

- $P^i_{s,t}$ pour $1 \leq i \leq l$, $1 \leq s, t \leq T$. $P^i_{s,t}$ est vrai ssi la case du ruban numérotée s à l'étape t contient le symbole σ_i.
- Q^i_t pour $1 \leq i \leq r, 1 \leq t \leq T$. Q^i_t est vrai ssi à l'étape t, la machine est dans l'état q_i.
- $S_{s,t}$ pour $1 \leq s, t \leq T$ est vrai ssi à l'instant t, la case numérotée s est scannée par la tête de lecture.

La formule $A(w)$ est une conjonction $B \wedge C \wedge D \wedge E \wedge F \wedge G \wedge H \wedge I$ formée comme suit. Noter que $A(w)$ est en forme normale conjonctive.

B affirme qu'à chaque étape t, une et une seule case est scannée. B est une conjonction $B_1 \wedge B_2 \wedge \ldots \wedge B_T$, où B_T affirme qu'à l'instant t, une et une seule case est scannée :

4. Ici, le papier original mentionne $\{q_1, \ldots, q_s\}$ à la place de $\{q_1, \ldots, q_r\}$. Cela est difficilement lisible, étant écrit à la main "r" sous le "s" et par la suite, Cook ne fait plus référence à s mais à r ; donc il est vraisemblable que q_r est correct.

$$B_t = (S_{1,t} \vee S_{2,t} \vee \ldots \vee S_{T,t}) \wedge \left[\bigwedge_{1 \leq i < j \leq T} (\neg S_{i,t} \vee \neg S_{j,t}) \right].$$

Pour $1 \leq s \leq T$ et $q \leq t \leq T_j$, $C_{s,t}$ affirme que dans la case s et à l'instant t, il y a un et seulement un symbole. C est la conjonction de tous les $C_{s,t}$.

D affirme que pour tout t, il y a un et un seul état.

E affirme que les conditions initiales sont satisfaites :

$$E = Q_1^0 \wedge S_{1,1} \wedge P_{1,1}^{i_1} \wedge P_{2,1}^{i_2} \wedge \ldots \wedge P_{n,1}^{i_n} \wedge P_{n+1,1}^1 \wedge \ldots \wedge P_{T,1}^1$$

où $w = \sigma_{i_1} \ldots \sigma_{i_n}$, q_0 est l'état initial et σ_1 est le symbole blanc.

F, G, et H affirment que pour tout instant t, les valeurs des P, Q et S sont mises à jour correctement. Par exemple, G est la conjonction sur tous les t, i, j de $G_{i,j}^t$, où $G_{i,j}^t$ affirme que si à l'instant t, la machine est dans l'état q_i en train de scanner le symbole σ_j, alors à l'instant $t+1$, la machine est dans l'état q_k, où q_k est l'état donné par la fonction de transition pour M.[5]

$$G_{i,j}^t = \bigwedge_{s=1}^{T} \left(\neg Q_t^i \vee \neg S_{s,t} \wedge \neg P_{s,t}^j \vee Q_{t+1}^k \right).$$

Finalement, la formule I affirme que la machine atteint un état de reconnaissance (acceptation) à un certain instant. La machine M devrait être modifiée de telle façon qu'elle continue de calculer de manière triviale après avoir atteint un état d'acceptation, de telle façon que $A(w)$ soit satisfait.

Il est évident de vérifier que $A(w)$ a toutes les propriétés énoncées dans le premier paragraphe de la démonstration. □

Théorème 2. *Les ensembles suivants sont* P*-réductibles les uns aux autres par paires (et par conséquent, ils ont le ;même ;degré polynomial de difficulté) :* {*Tautologies*}, {*Tautologies FND*}, *D3,* {*Paires de sous−graphes*}.

Remarque. Nous n'avons pas été capables d'ajouter soit {Nombres premiers} soit {Paires de graphes isomorphes} à la liste ci-dessus. Montrer que {Tautologies} est P-réductible à {Nombres premiers} semblerait nécessiter des résultats profonds de théorie des nombres, alors que montrer que {Tautologies} est P-réductible à {Paires de graphes isomorphes} bouleverserait probablement une conjecture de Corneil [4] dont il déduit que le problème de l'isomorphisme des graphes peut être résolu en temps polynomial.

Incidemment, il n'est pas[6] difficile de voir à partir de la procédure de Davis-Putnam [5] que l'ensemble D_2 constitué de toutes les tautologies FND avec au plus deux éléments conjugués par disjonction, est dans $\mathscr{L}_*$. Par conséquent, D_2 ne peut pas être ajouté à la liste dans le théorème 2

5. À la suite de cette phrase, l'article contient quelques annotations à la main que je n'ai pas pu déchiffrer.
6. L'article original contient une erreur typographique ici (“it” au lieu de “it is”).

(à moins que tous les ensembles dans la liste ne soient dans $\mathscr{L}_*$).

Preuve du théorème 2. Par le corollaire du théorème 1, chacun des ensembles est P-réductible à {Tautologies FND}. Puisque de façon évidente, {Tautologies FND} est P-réductible à {Tautologies}, il reste à montrer que {Tautologies FND} est P-réductible à D_3 et que D_3 est P-réductible à {Paires de sous−graphes}.

Pour montrer que {Tautologies FND} est P-réductible à D_3, appelons A une formule propositionnelle en forme normale disjonctive. Disons que $A = B_1 \vee B_2 \vee \ldots B_k$, où $B_1 = R_1 \wedge \ldots \wedge R_s$, et où chaque R_i est un atome ou la négation d'un atome, et $s > 3$. Alors A est une tautologie si et seulement si A' est une tautologie où

$$A' = P \wedge R_3 \wedge \ldots \wedge R_s \vee \neg P \wedge R_1 \wedge R_2 \vee B_2 \vee \ldots \vee B_k,$$

où P est un nouvel atome. Puisque nous avons réduit le nombre de termes de la conjonction dans B_1, le processus peut être répété jusqu'à ce qu'à la fin, une formule soit trouvée avec au plus trois éléments conjoints par disjonction. Clairement, le processus complet est borné en temps par un polynôme de la longueur de A.

Il reste à montrer que D_3 est P-réductible à {Paires de sous−graphes}. Supposons que A soit une formule en forme normale disjonctive avec trois conjoints par disjonction. Alors $A = C_1 \vee \ldots \vee C_k$, où $C_i = R_{i1} \wedge R_{i2} \wedge R_{i3}$, et où chaque R_{ij}, est un atome ou la négation d'un atome. Maintenant soit G_1 le graphe complet avec comme sommets $\{v_1, v_2, \ldots, v_k\}$, et soit G_2 le graphe de sommets $\{u_{ij}\}$, $1 \leq i \leq k, 1 \leq j \leq 3$, tel que u_{ij} est connecté par un arc à u_{rs} si et seulement si $i \neq r$ et les deux littérals (R_{ij}, R_{rs}) ne forment pas une paire opposée (c'est-à-dire qu'ils ne sont ni de la forme $(P, \neg P)$ ni de la forme $(\neg P, P)$). Donc il y a une assignation des valeurs de vérité qui rend fausse la formule A ssi il y a un homomorphisme de graphe $\phi : G_1 \to G_2$ tel que pour tout $i, \phi(i) = u_{ij}$ pour un certain j. (L'homomorphisme dit pour tout i quel élément parmi R_{i1}, R_{i2}, R_{i3} devrait être rendu faux, et le manque sélectif d'arcs dans G_2 garantit que l'assignation de valeurs de vérité qui en résulte est spécifiée de manière consistante.)

Dans le but de garantir qu'un homomorphisme bijectif $\phi : G_1 \to G_2$ a la propriété que pour tout i, $\phi(i) = u_{ij}$ pour un certain j, on modifie G_1 et G_2 comme suit. On sélectionne les graphes $H_1, H_2, ...H_k$, qui sont suffisamment distincts l'un de l'autre, que si G_1' est fabriqué à partir de G_1 en attachant H_i à v_i, $1 \leq i \leq k$, et G_2' est fabriqué à partir de G_2 en attachant H_i à chacun des u_{i1} et u_{i2} et u_{i3}, $1 \leq i \leq k$, alors tout homomorphisme bijectif $\phi : G_1' \to G_2'$ a la propriété qui vient d'être énoncée. Il n'est pas difficile de voir qu'une telle construction peut être effectuée en temps polynomial. Alors G_1' peut être un sous-graphe de G_2' si et seulement si $A \notin D_3$. Cela complète la preuve du théorème 2. □

2. Discussion

Le théorème 1 et son corollaire montrent avec une grande évidence qu'il n'est pas facile de déterminer si une formule propositionnelle donnée est une tautologie, même si la formule est sous forme normale disjonctive. Les théorèmes 1 et 2 ensemble suggèrent qu'il est infructueux de chercher une procédure de décision polynomiale pour le problème des sous-graphes, puisque le succès amènerait

des procédures de décision polynomiales pour de nombreux autres problèmes apparemment non traitables. Bien sûr la même remarque s'applique à n'importe quel problème combinatoire auquel {Tautologies} est P-réductible.

De plus, les théorèmes suggèrent que {Tautologies} est un bon candidat pour un ensemble intéressant qui n'est pas dans $\mathscr{L}_*$, et j'ai le sentiment que cela vaut la peine de faire des efforts considérables pour essayer de prouver cette conjecture. Une telle preuve serait une percée majeure en théorie de la complexité.

Au vu de l'apparente complexité de {Tautologies FND}, il est intéressant d'examiner la procédure de Davis Putnam [5]. Cette procédure a été conçue pour déterminer si une formule donnée en forme normale conjonctive est satisfiable, mais bien sûr, la procédure "duale" détermine si une formule sous forme normale disjonctive est une tautologie. Je n'ai pas encore été capable de trouver une série d'exemples montrant que la procédure (convenablement traitée pour éviter certains pièges) doit nécessiter un temps plus que polynomial. Je n'ai pas non plus trouvé une borne supérieure intéressante pour le temps nécessaire.

Si on utilise les chaînes de caractères pour représenter des entiers naturels, (ou des k-uplets d'entiers naturels) en utilisant la notation m-adique ou toute autre notation appropriée, alors les notions des sections précédentes peuvent être modifiées pour s'appliquer à des ensembles de nombres (ou à des relations à k-places sur des nombres). Il n'est pas difficile de montrer que l'ensemble des relations acceptées en temps polynomial par une certaine machine de Turing non déterministe est précisément l'ensemble $\mathscr{L}^+$ des relations de la forme

$$(\exists y \leq g_k(\overline{x}))R(\overline{x}, y) \tag{1}$$

où $g_k(\overline{x}) = 2^{(l(\max \overline{x}))^k}$, $l(z)$ est la longueur dyadique de z, et $R(\overline{x}, y)$ est une relation $\mathscr{L}_*$, ($\mathscr{L}^+$ est la classe des relations rudimentaires positives étendues de Bennett [6]). Si on enlevait la contrainte sur le quantificateur dans la formule (1), la classe $\mathscr{L}^+$ deviendrait la classe des ensembles récursivement énumérables. Donc si $\mathscr{L}^+$ est l'analogue de la classe des ensembles r.e., alors déterminer le caractère tautologique est l'analogue du problème de l'arrêt ; puisque, selon le théorème 1, {Tautologies} a le degré complet $\mathscr{L}^+$ juste comme le problème de l'arrêt a le degré complet r.e.. Malheureusement, l'argument de la diagonale qui montre que le problème de l'arrêt n'est apparemment pas récursif ne peut pas être adapté pour montrer que {Tautologies} n'est pas dans $\mathscr{L}_*$.

3. Le calcul des prédicats

Les formules dans le calcul des prédicats sont représentées par des chaînes de caractères de façon similaire à ce qui se fait en calcul propositionnel. En plus des symboles de ce dernier, on a besoin des symboles de quantification $\forall$ et $\exists$, et des symboles pour former une liste infinie de variables individuelles, et des listes infinies de symboles de fonctions et de symboles de prédicats de chaque ordre (bien sûr, l'alphabet sous-jacent Σ est toujours fini).

Supposons que Q est une procédure qui agit sur les formules ci-dessus et qui se termine étant donnée une certaine formule en entrée A si et seulement si A est insatisfiable. Puisqu'il n'y a pas de procédure de décision pour la satisfiabilité pour le calcul des prédicats, il en découle qu'il n'y a

aucune fonction récursive T telle que si A est insatisfiable, alors Q terminera en $T(n)$ étapes, où n est la longueur de A. Comment évalue-t-on alors l'efficacité de la procédure ?

Nous allons prendre l'approche suivante. La plupart des solveurs automatiques de théorèmes dépendent du théorème de Herbrand, qui établit qu'une formule A est insatisfiable si et seulement si une certaine conjonction d'instances de substitution de la forme fonctionnelle $fn(A)$ de A est de valeur de vérité fonctionnellement inconsistante. Supposons que nous ordonnons les termes dans l'univers de Herbrand de $fn(A)$ selon leur rang, et que nous ordonnions alors de façon naturelle les instances de substitution de $fn(A)$ de l'univers de Herbrand. L'ordonnancement devrait être tel qu'en général, les instances de substitution qui utilisent des termes de rang plus grand suivent celles qui utilisent des termes de rang moindre. Soient $A_1, A_2, \ldots$ ces instances de substitution dans l'ordre.

Définition. *Si A est insatisfiable, alors $\phi(A)$ est le plus petit k tel que $A_1 \wedge A_2 \wedge \ldots \wedge A_k$ est à valeur de vérité fonctionnellement inconsistante. Si A est satisfiable, alors $\phi(A)$ est indéfini.*

Maintenant soit Q la procédure qui, étant donné A, calcule la séquence $A_1, A_2, \ldots$ et pour tout i, teste si $A_1 \wedge \ldots \wedge A_i$ est à valeur de vérité fonctionnellement inconsistante. Si la réponse est non, la procédure s'arrête avec succès. Alors il y a de façon évidente une fonction récursive $T(k)$ telle que pour tout k et toutes les formules A, si la longueur de $A \leq k$ et $\phi(A) \leq k$, alors Q terminera en $T(k)$ étapes. Nous suggérons que la fonction $T(k)$ est une mesure de l'efficacité de Q.

Pour des raisons pratiques, toutes les procédures dans cette section seront réalisées sur des machines de Turing à ruban unique, que nous appellerons simplement *machines*.

Définition. *Étant donnée une machine M_Q et une fonction récursive $T_Q(k)$, nous dirons que M_Q est de type Q et effectue un calcul en temps $T_Q(k)$ lorsque quand M_Q commence avec une formule du calcul des prédicats A écrite sur son ruban, alors M_Q s'arrête si et seulement si A est insatisfiable, et pour tout k, si $\phi(A) \leq k$ et $|A| \leq \log_2 k$, alors M_Q s'arrête en $T_Q(k)$ étapes. Dans ce cas, nous dirons également que $T_Q(k)$ est de type Q. Ici $|A|$ est la longueur de A.*

La raison de la condition $|A| \leq \log_2 k$ plutôt que $|A| \leq k$, est qu'avec cette dernière, trouver une borne inférieure pour $T_Q(k)$ serait presque équivalent à trouver une borne inférieure pour le problème de décision pour le calcul propositionnel. En particulier, le théorème 3A deviendrait évident et trivial.

Théorème 3. *A) Pour tout $T_Q(k)$ de type Q,*

$$\frac{T_Q(k)}{\sqrt{k}/(\log k)^2} \tag{2}$$

est non borné.

B) Il existe un $T_Q(k)$ de type Q tel que

$$T_Q(k) \leq k\, 2^{k(\log k)^2}.$$

Esquisse de la preuve. A) Étant donnée une machine M, on peut construire une formule du calcul des prédicats $A(M)$ qui est satisfiable si et seulement si M ne s'arrête jamais quand elle commence

avec un ruban blanc. C'est ce qui est fait tout au long des lignes décrites dans l'article de Wang [7] dans la démonstration qui réduit le problème de l'arrêt au problème de la décision en calcul des prédicats. De plus, si M s'arrête en s étapes, alors $\phi(A(M)) \leq s^2$. Ainsi, si, au contraire de (2), $T_Q(k) = O(\sqrt{k}/\log^2 k)$, alors une modification de M_Q devrait pouvoir vérifier en seulement

$$O(\sqrt{s^2}/\log^2 s^2) = O(s/\log^2 s)$$

étapes que M s'est arrêtée en s étapes (en supposant que $m \leq \log s^2$, où m est la longueur de $A(M)$). Un argument diagonal (voir [8] p. 153) montre que c'est en général impossible.

B) La machine M_Q procède en un temps T_Q en suivant la procédure décrite au début de cette section. Noter que la formule $A_1 \wedge A_2 \wedge \ldots \wedge A_k$ a pour longueur $O(k \log^2 k)$, puisque nous pouvons supposer $|A| \leq \log k$. □

Théorème 4. *Si l'ensemble S de chaînes de caractères est reconnu par une machine non déterministe dans un temps $T(n) = 2^n$, et si $T_Q(k)$ est une fonction "honnête" (i.e. calculable en temps-réel) de type Q, alors, il existe une constante K telle que S peut être reconnu par une machine déterministe dans un temps $T_Q(K\ 8^n)$.*

Preuve. Supposons que M_1 est une machine non-déterministe qui reconnaît S en temps 2^n. Soit M_2 une machine non-déterminisste qui simule M_1 pour exactement 2^n étapes et alors s'arrête, à moins que M_1 ne reconnaisse l'entrée, auquel cas M_2 calcule indéfiniment. Donc pour toutes les chaînes de caractères w, si $w \in S$ alors il existe un calcul pour lequel M_2 avec l'entrée w ne parvient pas à s'arrêter, et si $w \notin S$, alors M_2 avec l'entrée w s'arrête en 4^n étapes pour *tous* les calculs. Maintenant étant donné w de longueur n, on peut construire une formule $A(w)$ de longueur $O(n)$ telle que $A(w)$ est satisfiable si et seulement si M_1 reconnaît w. ($A(w)$ est construite d'une manière similaire à $A(M)$ dans la preuve de 3A.) [7] De plus, si M_2 s'arrête en 4^n étapes pour tous les calculs possibles, alors $\phi(A(w)) \leq K(4^n)^2 = K\ 8^n$. Donc, une machine déterministe M peut être construite pour déterminer si $w \in S$ en présentant à M_Q l'entrée $A(w)$. Si aucun résultat n'apparaît en $T_Q(K\ 8^n)$ étapes, alors $w \in S$, et sinon $w \notin S$. □

4. Discussion supplémentaire

Il y a un grand écart entre la borne inférieure de $\sqrt{k}/(\log k)^2$ pour les fonctions de temps $T_Q(k)$ données dans le théorème 3A et un possible

$$T_Q(k) = k\ 2^{k(\log k)^2}$$

donné en 3B. Pourtant, il y a des raisons à cet écart. Par exemple, si on pouvait améliorer le résultat en 3B et trouver un $T_Q(k)$ borné par un polynôme en k, alors par le théorème 4, on pourrait simuler une machine non déterministe à temps borné par 2^n de façon déterminée en temps $p(2^n)$ pour un certain polynôme p. C'est contraire à l'expérience que nous donnent des simulations déterministes d'une machine non déterministe bornée par un temps $T(n)$ qui indiquent qu'elles nécessitent un temps $k^{T(n)}$ en général.

7. L'article fait référence à "1A" ici. Puisque ce théorème n'existe pas, et que seul existe $A(M)$ dans 3A, il semble certain que ce soit 3A qui soit correct.

D'un autre côté, si nous pouvions élever la borne inférieure donnée dans le théorème 3A et montrer que

$$\frac{T_Q(k)}{2^k}$$

n'est pas bornée, alors nous pourrions conclure que {Tautologies} $\notin \mathscr{L}_*$, puisque sinon, la procédure de preuve générale de Herbrand fournirait un $T_Q(k)$ plus petit que 2^k. Donc une telle amélioration dans 3A devrait nécessiter une percée majeure en théorie de la complexité.

Le domaine de la mécanisation de théorèmes a foncièrement besoin d'une base pour comparer et évaluer les douzaines de procédures qui apparaissent dans la littérature. La performance d'une procédure sur des exemples sur ordinateurs est un critère valable, mais pas suffisant (à moins que la procédure s'avère utile d'une certaine façon pratique). Un critère de complexité théorique est nécessaire qui autorisera des limitations fondamentales et suggèrera de nouveaux buts à poursuivre. Le critère suggéré ici (la fonction $T_Q(k)$) est probablement trop grossière. Par exemple, il serait peut-être mieux de faire de $T_Q(k)$ une fonction à plusieurs variables, dont l'une serait $\phi(A)$, et une autre pourrait être le nombre minimum d'instances de substitution de $fn(A)$ qui sont nécessaires pour former une contradiction (noter qu'en général, tous les $A_1, A_2, ..., A_{\phi(A)}$ ne sont pas nécessaires).

$T_Q(k)$ peut être une mesure grossière, mais elle fournit une base de discussion et, j'espère, elle stimulera le progrès pour trouver de meilleures mesures de complexité pour les prouveurs de théorèmes.

Références

[1] D. L. Kreider and R. W. Ritchie : Predictably Computable Functionals and Definitions by Recursion. *Zeitschrift für math. Logik und Grundlagen der Math.*, Vol. 10, 65-80 (1964).

[2] S. A. Cook : Characterizations of Pushdown Machines in terms of Time-Bounded Computers. J. Assoc. *Computing Machinery*, Vol. 18, No. 1, Jan. 1971, pp 4-18.

[3] Cobham, Alan : The intrinsic computational difficulty of functions. *Proc. of the 1964 International Congress for Logic, Methodology, and the Philosophy of Science*, North Holland Publishing Co., Amsterdam, pp. 24-30.

[4] D. G. Corneil and C. C. Gotlieb : An Efficient Algorithm for Graph Isomorphism. *J. Assoc. Computing Machinery*, Vol. 17, No. 1, Jan. 1970, pp 51-64.

[5] M. Davis and H. Putnam : A Computing Procedure for Quantification Theory. *J. Assoc. Computing Machinery*, 1960, pp. 201-215.

[6] J. H. Bennett : *On Spectra*. Doctoral Dissertation, Princeton University, 1962.

[7] Hao Wang : Dominoes and the AEA case of the decision problems. *Proc. of the Symposium on Mathematical Theory of Automata*, at Polytechnic Institute of Brooklyn, 1962. pp. 23-55.

[8] John Hopcroft and Jeffrey Ullman : *Formal Languages and their Relation to Automata*. Addison Wesley, 1969.