Exploration de la conjecture PSW

Nicolas Reyland

22 décembre 2022

Résumé

Ce document est la deuxième partie du concours d'entrée au LRE (Laboratoire de Recherche de l'EPITA). Ce n'est certainement pas une preuve de la conjecture nommée ici PSW, mais plutôt une énumération des pistes qui ont été empreintées dans la recherche d'une affirmation ou d'une infirmation de la conjecture PSW.

Table des matières

1 Introduction

La conjecture PSW n'est pas son nom original. Il a été donné par manque de nom officiel. C'est en tout cas une conjecture donnée par Pomerance, Selfridge et Wagstaff. Voici son énoncé :

> Si n est un entier $n \in \mathbb{N}$ tel que les propositions suivantes sont vraies
> - n est impair ($\exists k \in \mathbb{N}$ tel que $n = 2k + 1$)
> - $n \equiv \pm 2[5]$
> - $2^{n-1} \equiv 1[n]$
> - $F_{n+1} \equiv 0[n]$ ((F_n) la suite de Fibonacci)
>
> Alors $n \in \mathbb{P}$ (n est premier).

2 Infirmation

L'information d'une telle hypothèse peut se faire de plusieurs manières, mais la manière la plus simple est de trouver un contrexemple. On peut aussi passer par un raisonnement par l'absurde, mais cela demande déjà plus de réflexion.

2.1 Recherche d'un contrexemple

Pour trouver un bon contrexemple à la conjecture PSW, il faut trouver un nombre composé n qui soit à la fois un nombre pseudo-premier de Fermat ET un nombre pseudo-premier de Fibonacci[3, §3.41]. Ce nombre doit donc vérifier les propositions suivantes, tout en n'étant pas premier :

- $n \equiv \pm 2\,[5]$
- $2^{n-1} \equiv 1\,[n]$
- $F_{n+1} \equiv 0\,[n]$

Voici le code source d'un programme en Haskell qui cherche tous les contrexemples en se basant sur une liste de l'OEIS[2] des nombres pseudo-premiers de Fermat allant jusqu'à 10^{12} :

```haskell
{-
This program finds all the counter-examples to the PSW conjeture
using a list of all the Fermat pseudo-primes to base 2 that are
less than 10^12.

(see haskell sources n°1).
-}

-- Nth Fibonacci number
fibo :: Integer -> Integer
fibo = fst . fib_
    where
        fib_ 0 = (1, 1)
        fib_ 1 = (1, 2)
        fib_ n
            | even n    = (a*a + b*b, c*c - a*a)
            | otherwise = (c*c - a*a, b*b + c*c)
            where
                (a,b) = fib_ (n `div` 2 - 1)
                c     = a + b

-- Check if the input is a counter example to the PSW conjecture
-- The input should already be a strong pseudoprime to base 2
isFiboCounterExample :: Integer -> Bool
isFiboCounterExample n = (fibo (n + 1)) `mod` n == 0

-- Must have the right remainder
isRightMod5 :: Integer -> Bool
isRightMod5 = (\n -> n == 2 || n == 3) . (flip rem 5)
```

```haskell
-- List of Fermat pseudo-prime numbers to base 2 (see haskell sources n°1).
strong_psp_b2_list :: IO [Integer]
strong_psp_b2_list = do
    content <- readFile "pseudo-primes-b2.txt"
    return $ map (read . tail . (dropWhile (/=' '))) (lines content)

-- List of numbers that are -2 (or 3) or 2 mod 5
selected_strong_psp_b2_list :: IO [Integer]
selected_strong_psp_b2_list = (filter isRightMod5) <$> strong_psp_b2_list

-- List of counter examples to the PSW conjecture
counter_examples :: IO [Integer]
counter_examples = (filter isFiboCounterExample) <$> selected_strong_psp_b2_list

main :: IO ()
main = do
    pure_counter_examples <- counter_examples
    putStrLn $
        "Counter Examples to the PSW conjecture: " ++
        show pure_counter_examples ++
        " (" ++ show (length pure_counter_examples) ++ " found)"
```

Les nombres sont stockés dans le fichier *pseudo-primes-b2.txt*. Comme on peut l'imaginer, ce programme nous confirme qu'il n'y a pas de contrexemples pour un n inférieur à 10^{12}. D'ailleurs, il y a aussi une liste de l'OEIS notant les dix-mille premiers nombres pseudo-premiers de Fibonacci[1]. On pourrait penser qu'il suffit de prendre l'intersection de ces deux listes. Même si cela aurait été plus rapide, la liste des nombres pseudo-premiers de Fibonacci de l'OEIS ne couvre pas un aussi grand intervalle ($n \leq 8 * 10^8$) que celle des nombres pseudo-premiers de Fermat ($n \leq 10^{12}$). Pour avoir plus de chances de trouver un contrexemple, il faut se baser sur la liste des nombres pseudo-premiers de Fermat et vérifier s'ils sont aussi des nombres pseudo-premiers de Fibonacci.

Notons d'ailleurs qu'une implémentation $O(1)$ pour obtenir F_n ne fonctionne pas, car à partir de $n \approx 76$, la précision du type *Double* en Haskell produisait des erreurs. Les implémentations très optimisées utilisant des matrices ou des listes ne fonctionnent pas non plus, parce qu'elles prennent trop d'espace mémoire (24G disponibles ont été remplies très rapidement), ou parce que pour avoir un élément avec un indice d'une liste en Haskell, on ne pouvait utiliser le *BigNum* d'Haskell (type *Integer*). Il fallait impérativement utiliser un *Int*, or les nombres dans la liste de l'OEIS sont bien plus grands que *MAX INT*.

2.2 Existence d'un contrexemple

Soient E l'ensemble des nombres μ pseudo-premiers de Fermat tels que $\mu \equiv \pm 2\,[5]$ et I l'ensemble des nombres ν pseudo-premiers de Fibonacci tels que $\nu \equiv \pm 2\,[5]$. Pour montrer qu'il existe un contrexemple à la conjecture PSW (si c'était fait avec succès, cela rapporterait $620 à l'auteur de la preuve), il faut montrer que $E \cap I \neq \varnothing$. D'ailleurs, on montre que la conjecture est vraie si on montre l'inverse ($E \cap I = \varnothing$).

Puisque nous essayons ici d'infirmer le théorème, montrons que $E \cap I \neq \varnothing$.

Soient $(a, b, n) \in \mathbb{N}^3$ tels que :

- $n = a * b$
- $n \neq a$
- $n \neq b$ (n est vraiment un nombre composé)

Ici, deux pistes sont valables. On peut dire que $n \in E$, puis montrer que $n \in I$, ou alors on dit que $n \in I$, puis il faut montrer que $n \in E$.

2.2.1 Hypothèse 1

Supposons que $n \in E$. On a donc $2^{n-1} \equiv 1\,[n]$. Soit (F_n) la suite de Fibonacci et (L_n) la suite de Lucas ($L_0 = 2$, $L_1 = 1$). Il y a des nombres qui valident toutes les conditions énumérées

jusqu'ici, comme 1387, 2047 ou 3277.

$$F_{n+1} = F_n + F_{n-1} = \frac{\varphi^{n+1} - \psi^{n+1}}{\varphi - \psi}$$

avec $\varphi = \frac{1+\sqrt{5}}{2}$ et $\psi = \frac{1-\sqrt{5}}{2}$

On a donc : ...

2.2.2 Hypothèse 2

Supposons que $n \in I$. On a donc $F_{n+1} \equiv 0\,[n]$ avec (F_n) la suite de Fibonacci. Il y a des nombres qui valident toutes les conditions énumérées jusqu'ici, comme 144. Nous savons que $n \equiv \pm 2\,[5]$. C'est équivalent à dire que $n \equiv 2\,[5]$ ou $n \equiv 3[5]$, car $-2 \equiv 3[5]$.

Montrons d'abord que tout entier n qui vérifie $n \equiv \pm 2\,[5]$ peut s'écrire $n = 10 * \lambda + r$ avec $\lambda \in \mathbb{N}$ et $r \in \{3, 7\}$:

Si $n \equiv \pm 2\,[5]$, alors $n \equiv 2\,[5]$ ou $n \equiv 3[5]$.

On peut écire $n = 10 * \lambda + r$ avec $\lambda \in \mathbb{N}$ et $k \in [\![0; 9]\!]$.

$10 * \lambda \equiv 0\,[5]$, donc $n \equiv r\,[5]$. Si $n \equiv 2\,[5]$, on peut restreindre k à l'ensemble $\{2, 7\}$. Si $n \equiv 3\,[5]$, on peut restreindre k à l'ensemble $\{3, 8\}$.

On peut alors restreindre r à $\{2, 3, 7, 8\}$. Enfin, on sait que n est impair. La parité de n est décidée par la parité de r, car c'est le dernier chiffre de la représentation décimale de n. Finalement, $r \in \{3, 7\}$

On en conclut que $n = 10 * \lambda + r$ avec $\lambda \in \mathbb{N}$ et $r \in \{3, 7\}$.

Enfin, analysons 2^{n-1} et son reste par n. Tout d'abord, si $n = 10 * \lambda + r$ avec $(\lambda, r) \in \mathbb{N} \times \{3, 7\}$, on peut dire que $n - 1 = 10 * \lambda + r$ avec $(\lambda, k) \in \mathbb{N} \times \{2, 6\}$.

On peut écrire $2^{n-1} = 2^{10*\lambda+k} = (2^{10})^{\lambda} * 2^k$.

Aussi, $2^k \in \{4, 64\} = \{4, 10 * 6 + 4\}$.

Propriété sur les puissances sur 6 (PP6)
Montrons par récurrence que toutes les puissances entières strictement positives d'un nombre d qui s'écrit $d = 10 * \lambda + 6$ avec $\lambda \in \mathbb{N}$ peuvent s'écrire sous cette même forme (avec un λ différent) :

Initialisation

Soit p la proposition telle que $p(n) =\ll d^n = 10 * \lambda + 6$ avec $\lambda \in \mathbb{N} \gg$

Soit $d = d^1 = 10 * \lambda + 6$ et $\lambda \in \mathbb{N}$.

$$d^2 = (10 * \lambda + 6)^2 = (10 * \lambda)^2 + 2 * 10 * \lambda + 6^2 = 10 * (10 * \lambda^2 + 2 * \lambda + 3) + 6$$

Or $10 * \lambda^2 + 2 * \lambda + 3 \in \mathbb{N}$, donc la proposition est vraie pour $n = 2$ ($p(2)$ est vraie).

Hérédité

Supposons la propriété $p(n)$ vraie pour un $n \in \mathbb{N}$. Donc $d = 10 * \gamma + 6$ et $d^n = 10 * \lambda + 6$ avec $(\gamma, \lambda) \in \mathbb{N}^2$.

$$d^{n+1} = d^n * d = (10 * \lambda + 6) * (10 * \gamma + 6)$$
$$\Leftrightarrow d^{n+1} = 10 * (10 * \lambda * \gamma + 6 * (\lambda + \gamma) + 3) + 6$$

Or $10 * \lambda * \gamma + 6 * (\lambda + \gamma) + 3 \in \mathbb{N}$, donc la propriété est vraie pour $p(n + 1)$.

Conclusion

La propriété est vraie $\forall n \in \mathbb{N}^\star$.

Propriété sur les puissances impaires sur 4 (PP4)
Montrons par récurrence que toutes les puissances entières strictement positives impaires d'un nombre d qui s'écrit $d = 10 * \lambda + 4$ avec $\lambda \in \mathbb{N}$ peuvent s'écrire sous cette même forme (avec un λ différent) :

Initialisation

Soit p la proposition telle que $p(n) =\ll d^n = 10 * \lambda + 4$ avec $\lambda \in \mathbb{N} \gg$
Soit $d = d^1 = 10 * \lambda + 4$ et $\lambda \in \mathbb{N}$.

$$\begin{aligned} d^3 = (10 * \lambda + 4)^3 &= (10\lambda)^3 + 4^3 + 12(10\lambda)^2 + 3(10\lambda)4^2 \\ &= 10(100\lambda^3 + 12(10\lambda^2) + 48\lambda) + 64 \\ &= 10(100\lambda^3 + 120\lambda^2 + 48\lambda + 6) + 4 \end{aligned}$$

Or $100\lambda^3 + 120\lambda^2 + 48\lambda + 6 \in \mathbb{N}$, donc la proposition est vraie pour $n = 3$ ($p(3)$ est vraie).

Hérédité

Supposons la propriété $p(n)$ vraie pour un $n \in \mathbb{N}_{n \text{ impair}}$. Donc $d = 10 * \gamma + 4$ et $d^n = 10 * \lambda + 4$ avec $(\gamma, \lambda) \in \mathbb{N}^2$.

$$\begin{aligned} d^{n+2} = d^n * d = (10\lambda + 4) * (10\gamma + 4)^2 \\ \Leftrightarrow d^{n+2} = (10\lambda + 4) * ((10\gamma)^2 + 2(40\gamma) + 16) \\ \Leftrightarrow d^{n+2} = (10\lambda + 4) * (10 * (10\gamma^2) + 10 * 8\gamma + 10 * 1 + 6) \\ \Leftrightarrow d^{n+2} = (10\lambda + 4) * (10(10\gamma^2 + 8\gamma + 1) + 6) \\ \Leftrightarrow d^{n+2} = (10\lambda + 4) * (10\omega + 6) \text{ avec } \omega = 10\gamma^2 + 8\gamma + 1 \\ \Leftrightarrow d^{n+2} = 10^2\lambda\omega + 40\omega + 60\lambda + 24 \\ \Leftrightarrow d^{n+2} = 10(10\lambda\omega + 4\omega + 6\lambda + 2) + 4 \end{aligned}$$

Or $10\lambda\omega + 4\omega + 6\lambda + 2 \in \mathbb{N}$, donc la propriété est vraie pour $p(n + 2)$.

Conclusion

La propriété est vraie $\forall n \in \mathbb{N}_{n \text{ impair}}$.

On peut écrire $2^{n-1} = (2^{10})^\lambda * 2^k$ avec $\lambda \in \mathbb{N}$ et $k \in \{2, 6\}$.

Si $k = 2$, alors :
$(2^{10})^\lambda * 2^k = (2^{10})^\lambda * 2^2 = (2^{10})^\lambda * 4$ et $(2^{10})^\lambda$ est soit 1 (car $\lambda = 0$), soit de la forme $10 * \gamma + k'$ avec $\gamma \in \mathbb{N}$ et $k' \in \{4, 6\}$ car $(2^{10})^\lambda = 1024^\lambda$.
Si $\lambda = 1$, $1024^\lambda = 1024 = 10 * 102 + 4$ ($\gamma = 102$ et $k' = 4$). On a vu précédemment (PP4) que toutes les puissances impaires de 1024 peuvent s'écrire avec un 4 en dernière place en représentation décimale. Ainsi, $k' = 4$ pour tous les *lambdas* impairs.
Si $\lambda = 2$, $1024^\lambda = 1024^2 = 1048576 = 10 * 104857 + 6$ ($\gamma = 104857$ et $k' = 6$). On a vu préalablement (PP6) que toutes les puissances de 1048576 peuvent s'écrire avec un 6 en dernière place en représentation décimale. Donc, $k' = 6$ pour tous les λ pairs (sauf zéro).

On en conclut que $k' = \begin{cases} 4 & \text{si } \lambda \text{ est impair} \\ 6 & \text{si } \lambda \text{ est pair} \end{cases}$

Si $k = 6$, alors :
$(2^{10})^\lambda * 2^k = (2^{10})^\lambda * 2^6 = (2^{10})^\lambda * 6 * 10 + 4$ et $(2^{10})^\lambda$ peut s'écrire comme dans le cas où $k = 2$.

Par conséquent, 2^{n-1} peut s'écrire $(10 * \lambda + s) * 2^k$ avec $\lambda \in \mathbb{N}$, $s \in \{2, 6\}$ et $k = \{4, 6\}$. Or,

$$(10 * \lambda + s) * 2^k = 10 * (\lambda * 2^k) + s * 2^k$$

Or $\lambda * 2^k \in \mathbb{N}$ et $s * 2^k = \begin{cases} 4 * 2^2 = 16 = 10 * 1 + 6 \\ 4 * 2^6 = 256 = 10 * 25 + 6 \\ 6 * 2^2 = 24 = 10 * 2 + 4 \\ 6 * 2^6 = 384 = 10 * 38 + 4 \end{cases}$ avec $(\lambda, s, k) \in \mathbb{N} \times \{4, 6\} \times \{2, 6\}$

Enfin,

$$\forall n \in I, 2^{n-1} = 10 * \vartheta + t,\ \vartheta \in \mathbb{N},\ t \in \{4, 6\}$$

Ce qu'on a montré jusqu'ici

On a montré que $\forall n \in I, 2^{n-1} = 10 * \vartheta + t$, $\vartheta \in \mathbb{N}$, $t \in \{4, 6\}$.
On sait aussi que $n = 10 * \gamma + r$ avec $\gamma \in \mathbb{N}$ et $r \in \{3, 7\}$.

Il faut trouver un $n \in I$ tel que $2^{n-1} \equiv 1\,[n]$. Peut-être ces résultats sont-ils utiles pour réduire l'ensemble de recherche ?

3 Affirmation

Références

[1] J. Dana. Odd fibonacci pseudoprimes : odd composite numbers k such that either (1) k divides fibonacci(k-1) if k == +-1 (mod 5) or (2) k divides fibonacci(k+1) if k == +-2 (mod 5). https://oeis.org/A081264, 2015. Accessed : 2022-12-21.

[2] A. Joerg. Fermat pseudoprimes to base 2, also called sarrus numbers or poulet numbers. https://oeis.org/A001567, 2014. Accessed : 2022-12-21.

[3] C. Richard and P. Carl. *Prime Numbers, A Computational Perspective (Second Edition)*. Springer, 2005.