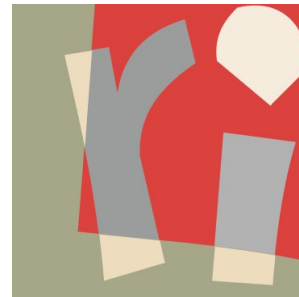


TER INDIVIDUEL

DISTRIBUTION D'UN CALCUL

COMBINATOIRE

EDGAR FOURNIVAL



PARTIE I

CONTEXTE



LRI

- Laboratoire de Recherche en Informatique
- Campus d'Orsay en face de l'IUT et du PUIO
- UMR impliquant l'Université Paris-Sud et le CNRS
- Bâtiment 650 Ada Lovelace

ÉQUIPE GALAC

- Graphes, Algorithmes et Combinatoire
- Dirigé par Johanne Cohen
- Calcul algébrique, études d'algorithmes, théorie des graphes, spécification théorique de systèmes en réseau
- Projet européen Horizon 2020 : OpenDreamKit, Sage



— OPEN —
DREAMKIT

MISSION

Distribuer un problème combinatoire

Exploration de l'arbre des semigroupes numériques

$$S_E = \{0, 3, 6, 7, 9, 10\} \cup \{x \in \mathbb{N}, x \geq 12\}$$

Objectif : lancer le calcul à une profondeur jamais
calculée auparavant

Problème : l'arbre est fortement déséquilibré

PARTIE II

PROBLÈME POSÉ ET SOLUTION EXISTANTE

$$S_8 = \{0, \mathbf{6}, \mathbf{7}, \mathbf{10}, 12, 13, 14, \mathbf{15}, 16\} \cup [17; +\infty[$$

Ensemble d'entiers $\in \mathbb{N}$ stable par l'addition,
complémentaire fini

Trou : élément du complémentaire

Genre : nombre de trous

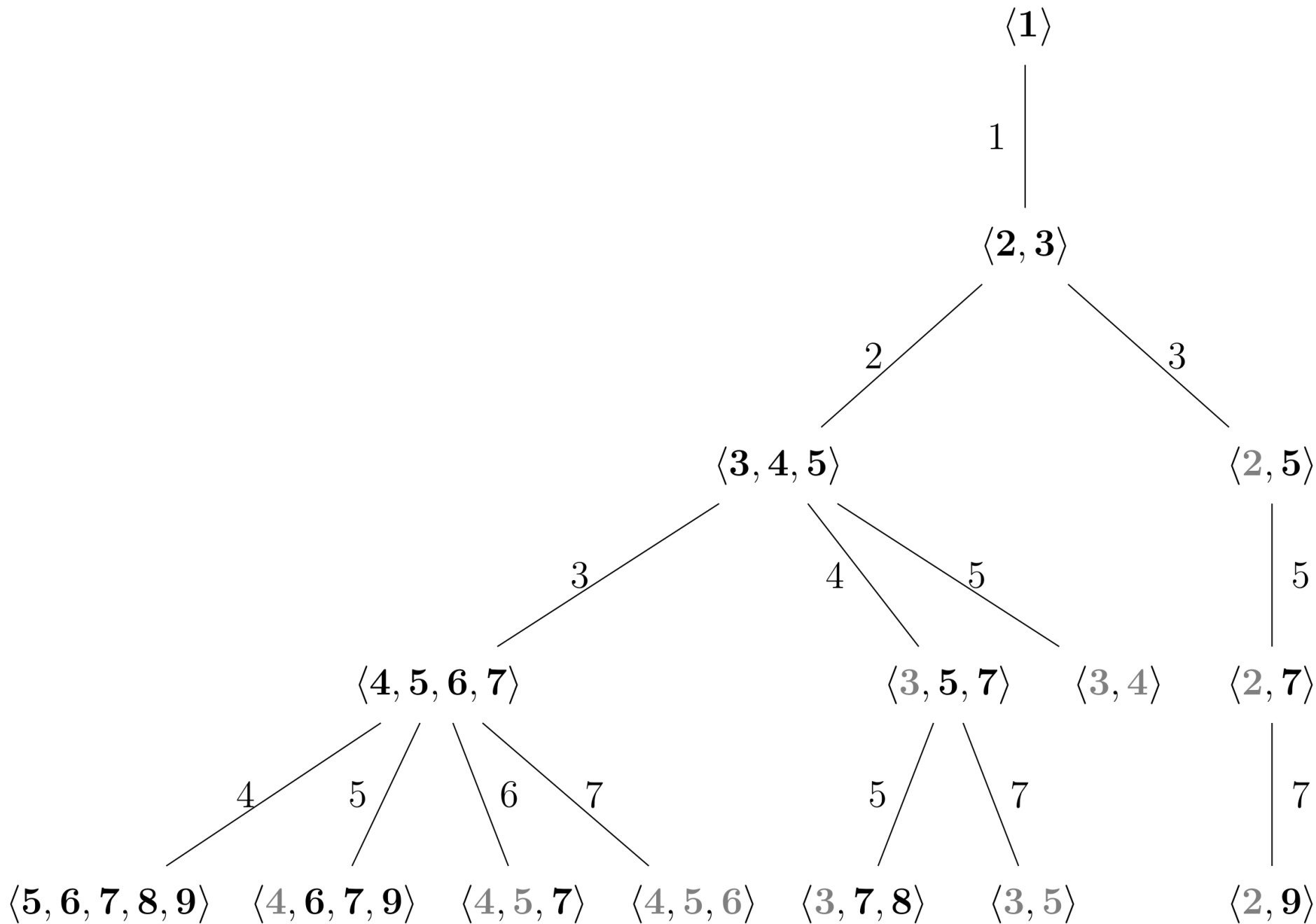
Générateur : nombre qu'on ne peut pas obtenir
comme somme d'autres nombres

Problème : compter le nombre de semigroupes
numériques de genre 70

$$S_9 = \{0, \mathbf{6}, \mathbf{7}, \mathbf{10}, 12, 13, 14, 16\} \cup [17; +\infty[$$

noté

$\langle \mathbf{6}, \mathbf{7}, \mathbf{10} \rangle$



ALGORITHME UTILISÉ

DFS en premier lieu par Bras-Amorós en $O(g^2)$ pour calculer les fils à partir d'un noeud

Amélioré par Florent Hivert et Jean Fromentin

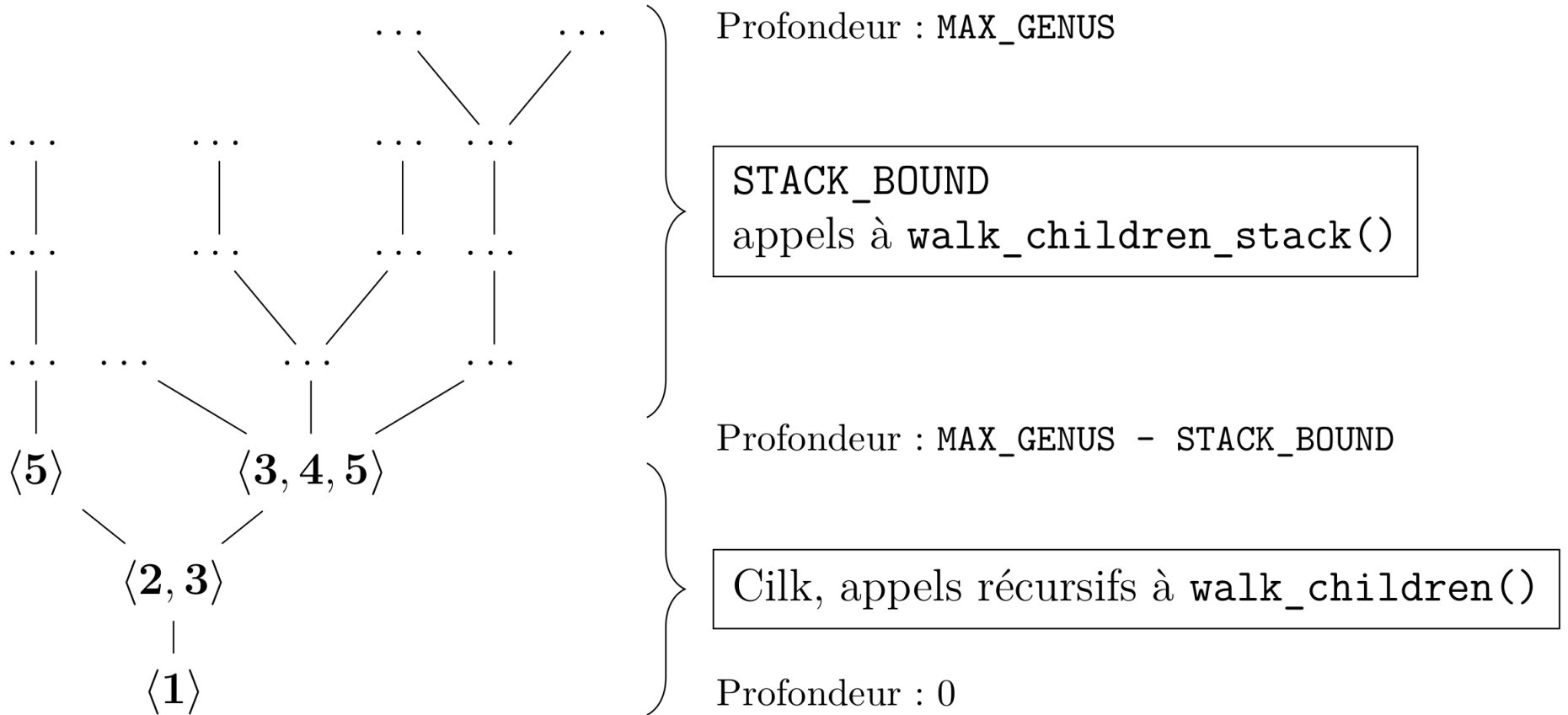
BFS, complexité $O(g)$ mais surtout petite constante car utilisation des instructions SIMD

Vol de tâches pour la parallélisation

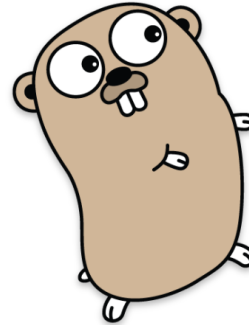
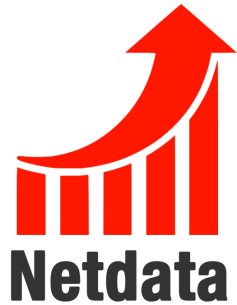
PARTIE III

DISTRIBUTION

IDÉE GÉNÉRALE



TECHNOLOGIES UTILISÉES



Travis CI

BUG GCC N°80038

```
#include <vector>
#include <cilk/cilk.h>

void walk(std::vector v, unsigned size) {
    if (v.size() < size)
        for (int i=0; i<8; i++) {
            std::vector<int> vnew(v);
            vnew.push_back(i);
            cilk_spawn walk(vnew, size);
        }
}

int main(int argc, char **argv) {
    std::vector<int> v{};
    walk(v, 5);
}
```

RÉSULTATS ?

**MERCI POUR VOTRE
ATTENTION**