

¿Dónde hay grupos de trabajo enfocados en el estudio de la complejidad en México?

Oscar Andrés Rosas Hernández *Instituto Politécnico Nacional, Escuela Superior de Cómputo, CDMX*

I. CELLULAR AUTOMATAS

```
#include <bitset>
#include <cstdint>
#include <iostream>
#include <vector>

using namespace std;
using space = vector<int>;
using alphabelt = unsigned int;

auto get_rules(const int rules_id) {
    return [=](const space &s, int i) -> alphabelt {
        const int limit = s.size() - 1;
        const auto n1 = i == 0 ? limit : i - 1;
        const auto n2 = i == limit ? 0 : i + 1;

        const alphabelt id = (s[n1] << 2) + (s[i] << 1) + (s[n2] << 0);
        return (rules_id >> id) bitand 1;
    };
}

auto print_space(const space &s) -> void {
    for (auto x : s) {
        cout << (x == 0 ? "_" : "1") << " ";
    }
    cout << endl;
}

auto print_evolution(const int steps, const space s, const int rules_id)
    -> void {
    const auto rules = get_rules(rules_id);
    const int limit = s.size();
    auto current = s, temporal = s;

    for (auto step = 0; step < steps; ++step) {
        print_space(current);

        for (auto i = 0; i < limit; ++i)
            temporal[i] = rules(current, i);

        current = temporal;
    }
}

auto print_rules(const int rules_id) {
    for (auto i = uint64_t{}; i < 8; i++) {
        const auto n = 7 - i;
        const auto bit = (rules_id >> n) bitand 1;
        cout << bitset<3>{n} << ": " << bit << endl;
    }
}

auto main() -> int {
    auto s = vector<int>(61, 0);
    s[30] = 1;

    const auto rule = 110;
    const auto iterations = 60;

    print_rules(rule);

    print_evolution(iterations, s, rule);
    return 0;
}
```

2

VE AL ÍNDICE

[illegible]

[illegible]

[illegible]