Symbolic Representation of Multivalued Logical Models in Rust

Lukáš Chudíček

Background

- Biological Systems analysis
- Exploring exponential state space
- Existing implementation for Boolean Systems
- Existing theory for Multivalued Systems

Goals

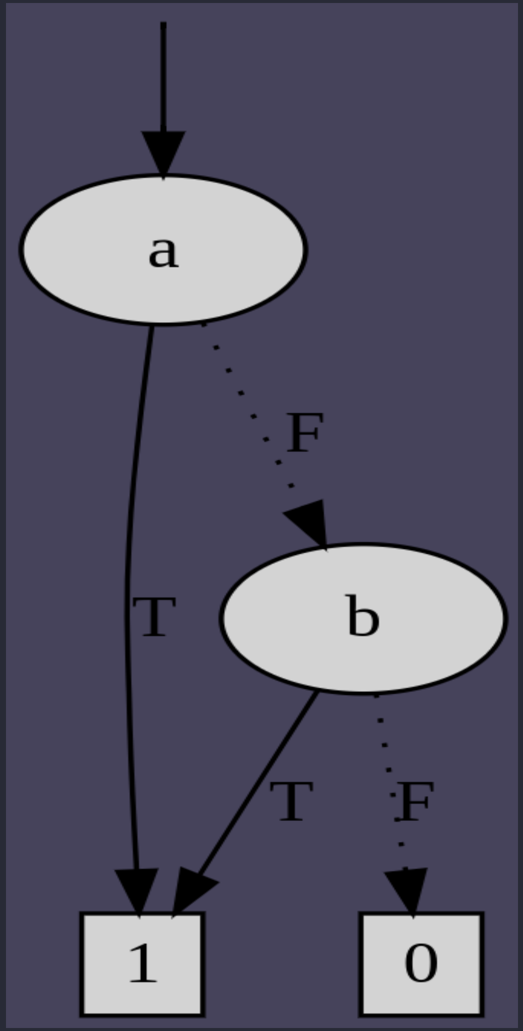
- Symbolic exploration of *multivalued* state space
- Compare encodings

```
fn load_model(entity_relationship_description: &XML) -> Model { /* ... */ }

struct Model<E: Encoding> {
    fn successors(&self, initial_states: &StateSet) -> StateSet { /* ... */ }
}
```

Binary Decision Diagram

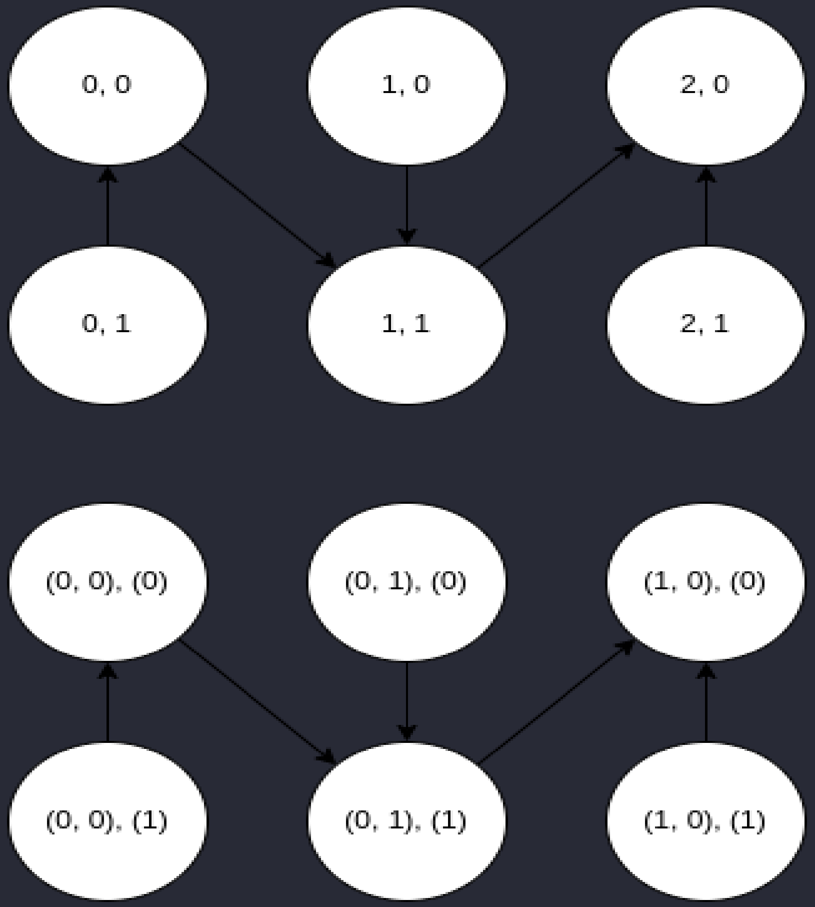
- Concise Boolean function
- Set's characteristic function
- Efficient set operations



Model

```
state_space = (
    (fun_0_0, fun_0_1),
    (fun_1_0),
);
```

State Space

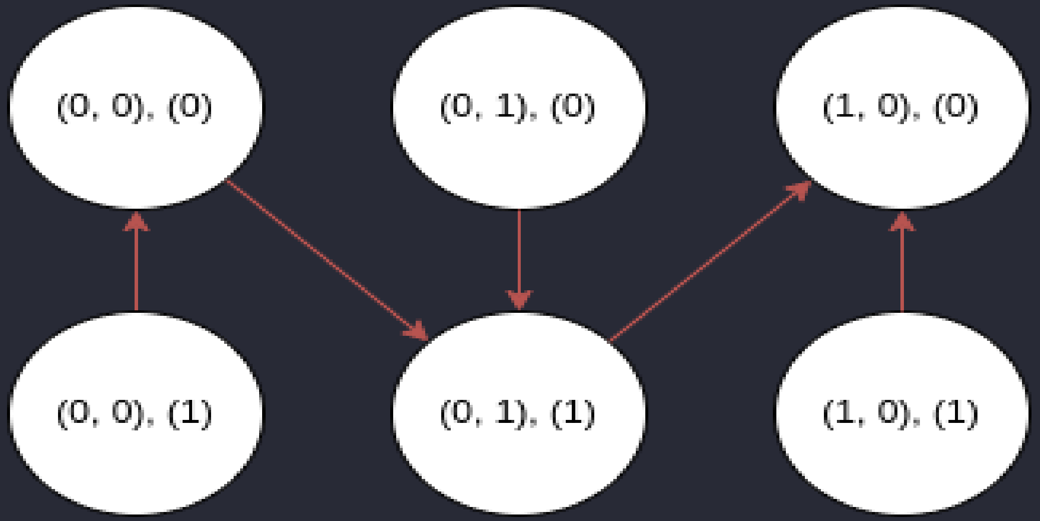


Successors

```
transition_relation_0 =
    (fun_0_0 <=> primed_bit_0_0) &&
    (fun_0_1 <=> primed_bit_0_1);

result_states_set = perform_transition(
    source_states_set,
    transition_relation_0
);
```

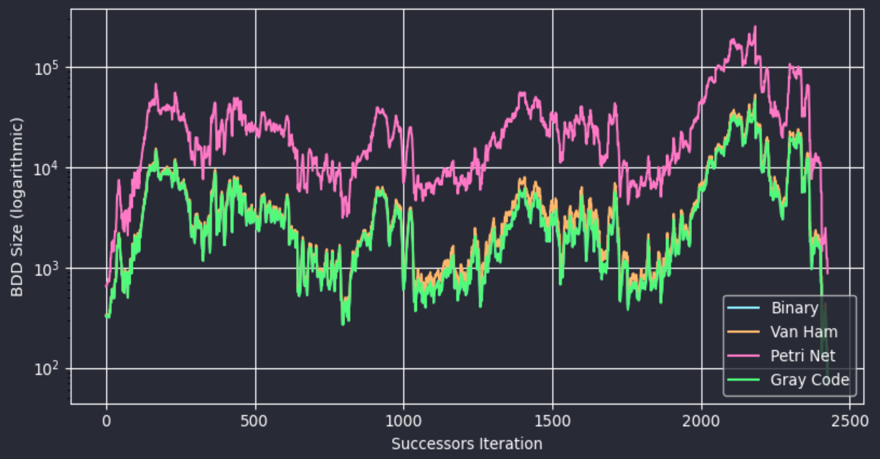
`transition_relation` ~ edges



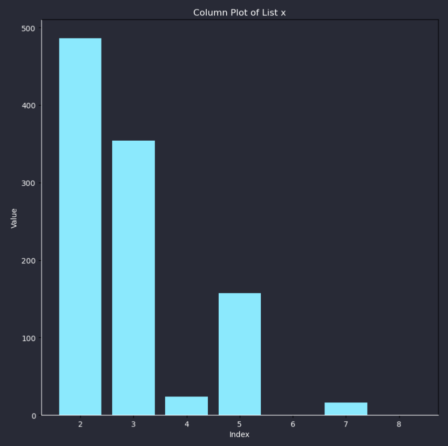
Encodings

Dec	Binary	Van Ham	Petri Net	Gray Code
0	`000`	`0000000`	`10000000`	`000`
1	`001`	`1000000`	`01000000`	`001`
2	`010`	`1100000`	`00100000`	`011`
3	`011`	`1110000`	`00010000`	`010`
4	`100`	`1111000`	`00001000`	`110`
5	`101`	`1111100`	`00000100`	`111`
6	`110`	`1111110`	`00000010`	`101`
7	`111`	`1111111`	`00000001`	`100`

Encodings Evaluation



Domain Sizes

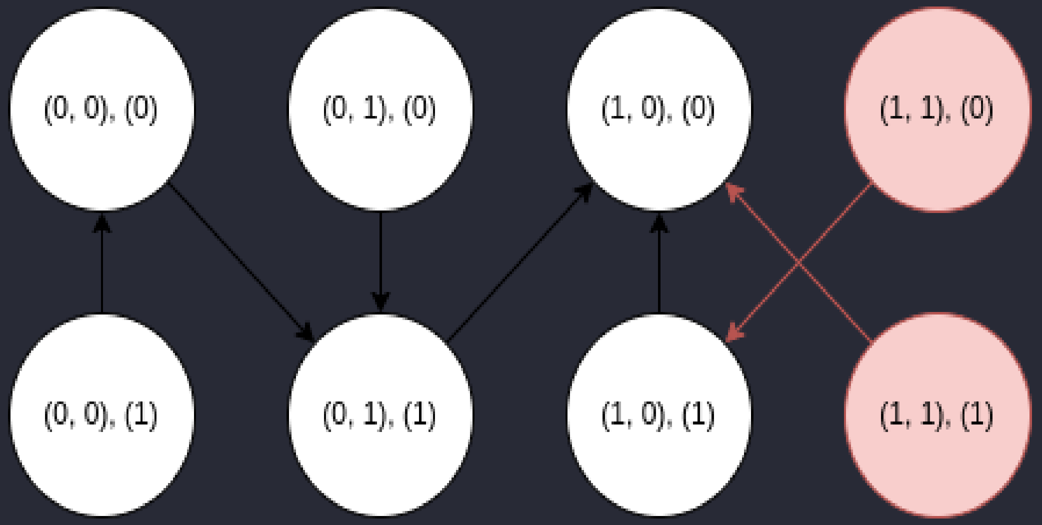


Result

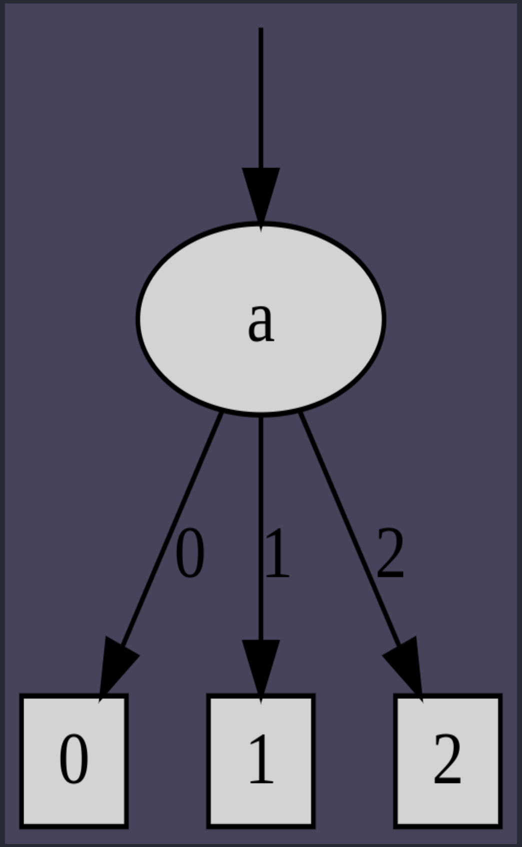
https://github.com/sybila/biodivine-lib-logical-models

Questions

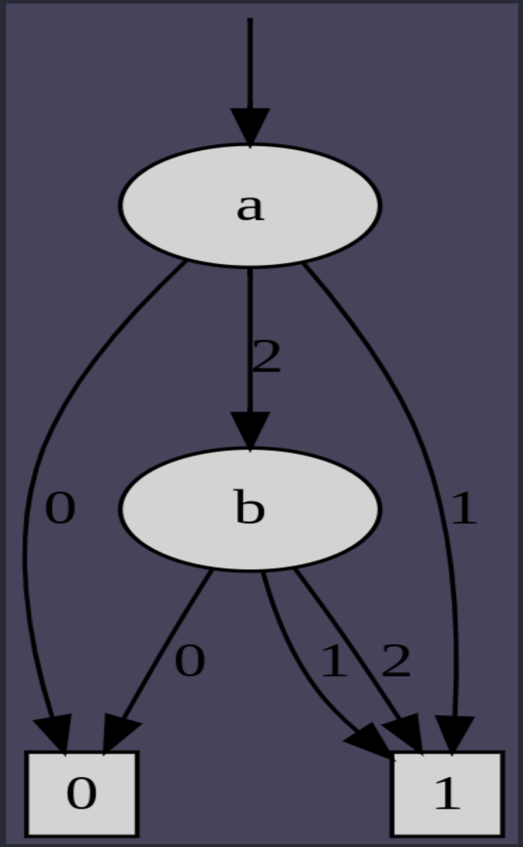
Invalid States



MDD with `n` terminals



MDD with `2` terminals


a
2
b
0
1
0
1
2
0
1