

Cell Model 与 Logic Programming

Zhiwei Ning



Agenda

Cell Model简介

Cell Model设计Dapp的困难

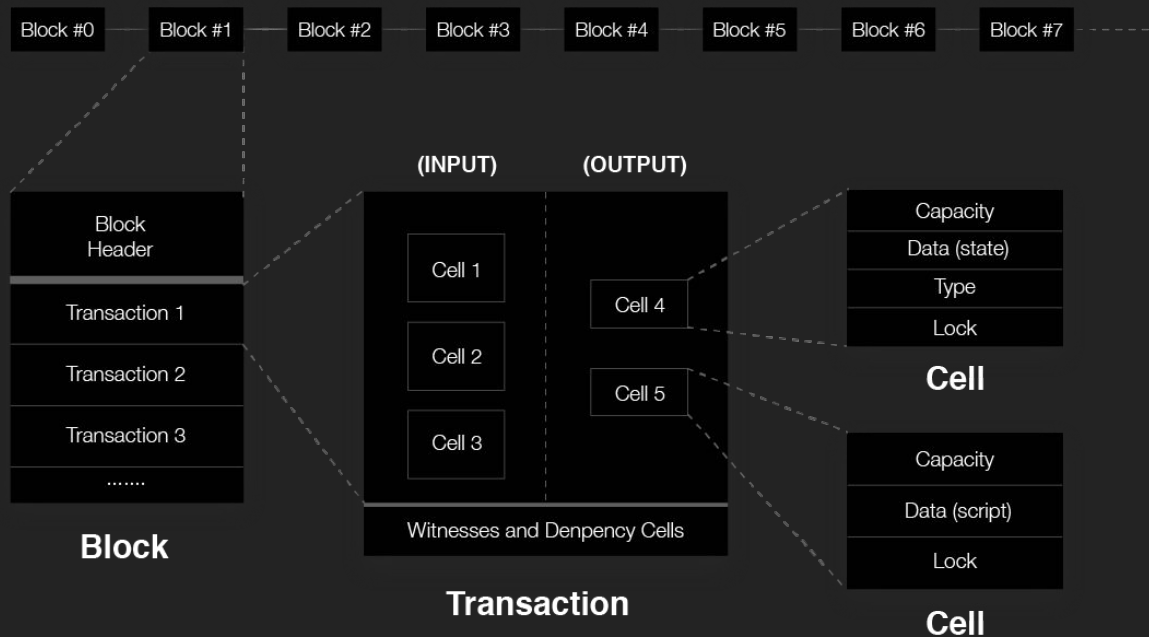
Logic Programming简介

Logic Programming的作用

Rust实现MicroKanren

01 Cell Model简介

Nervos CKB Structure



Cell model from generalizing the UTXO model.

02 Cell Model设计Dapp的困难







03 Logic Programming简介

1. There are five houses.
2. The Englishman lives in the red house.
3. The Spaniard owns the dog.
4. Coffee is drunk in the green house.
5. The Ukrainian drinks tea.
6. The green house is immediately to the right of the ivory house.
7. The Old Gold smoker owns snails.
8. Kools are smoked in the yellow house.
9. Milk is drunk in the middle house.
10. The Norwegian lives in the first house.
11. The man who smokes Chesterfields lives in the house next to the man with the fox.
12. Kools are smoked in the house next to the house where the horse is kept.
13. The Lucky Strike smoker drinks orange juice.
14. The Japanese smokes Parliaments.
15. The Norwegian lives next to the blue house.

```
:- use_module(library(clpfd)).  
change(X, [P5, P10, P20, P50]):-  
    P5 #>= 0, P10 #>= 0, P20 #>= 0, P50 #>= 0,  
    X #= 5 * P5 + 10 * P10 + 20 * P20 + 50 * P50,  
    label([P5, P10, P20, P50]).
```

```
?- change(X, [1,2,3,4]).  
X = 285.
```

```
?- change(100,X).  
X = [0, 0, 0, 2] ;  
X = [0, 0, 5, 0] ;  
X = [0, 1, 2, 1] ;  
X = [0, 2, 4, 0] ;  
X = [0, 3, 1, 1] ;  
X = [0, 4, 3, 0] ;  
X = [0, 5, 0, 1] ;  
X = [0, 6, 2, 0] ;  
X = [0, 8, 1, 0] ;  
X = [0, 10, 0, 0] ;  
X = [2, 0, 2, 1] ;  
X = [4, 0, 4, 0]  
Action (h for help) ? abort
```

04 Logic Programming的作用

逻辑编程有两个特点：

1. 不区分输入和输出。

可以正向计算 $1 + 2 = ?$

也可以反向计算 $1 + x = 3$ 那么 $x = ?$

比如转账，是先确定输出里转账的金额，然后才去查找live cell作为交易的输入。

2. 可以处理不确定的情况。就像解代数方程一样，如果只有一个解，会输出这个解；如果有多个解，会把所有的解都找出来，当然用户也可以控制，找到任何一个解的时候就停下来。

查找live cell的时候肯定有很多种组合可以满足要求。我们可以把所有可能的结果都列出来，然后再附加其他限制；也可以简单一点，找到任意一个结果就可以了。



```
transfer
```

```
  : locked K C * control U K * control U1 K1 -o  
locked K1 C * control U K * control U1 K1.
```

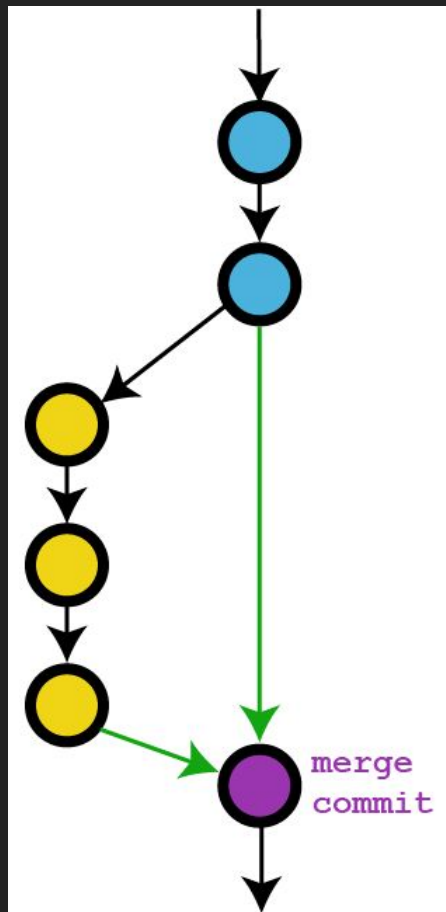
```
context init { (control alice alice_key),  
(control bob bob_key), (locked alice_key c) }
```

```
1: (transfer alice_key c alice bob bob_key)  
?- 1
```

```
1: (transfer bob_key c bob alice alice_key)  
?- 1
```

Final state:

```
{(control alice alice_key), (control bob  
bob_key), (locked alice_key c), (stage transfer)}
```



Cell Model的特点是输入输出都是确定的。
比如像HTLC的例子, Cell Model可以做到交易没上链之前就得到其交易Hash, 并据此构造后续的交易。

一开始就把Dapp需要的资源单独隔离出来。
将这些资源作为初始配置输入逻辑编程环境, 那么用户就可以像玩单机游戏一样进行任意的操作。

只有不得不跟链交互的时候, 再把相关交易提交上链。

POW的作用是解决冲突。

05 使用Rust实现MicroKanren

miniKanren是Dan Friedman在他的书《The Reasoned Schemer》中用来介绍逻辑编程的一个小巧的教学语言。其[官网](#)有相关资料和各种语言的实现。

microKanren是2013年由Dan Friedman的学生在一篇论文<[μKanren: A Minimal Functional Core for Relational Programming](#)>中提出的。对miniKanren进行了进一步的精炼, 整个语言的scheme实现只有40多行。

```
> ((call/fresh (lambda (q) (disj (== q 5) (== q 6))))  
'( () . 0))  
((( (#(0) . 5)) . 1) (((#(0) . 6)) . 1))
```

方程组

$$x_0 = 5 \text{ or } x_0 = 6$$

两组解

$$[\{x_0 = 5\}, \{x_0 = 6\}]$$

Code



代码

Thank You!

www.nervos.org

github.com/nervosnetwork

