

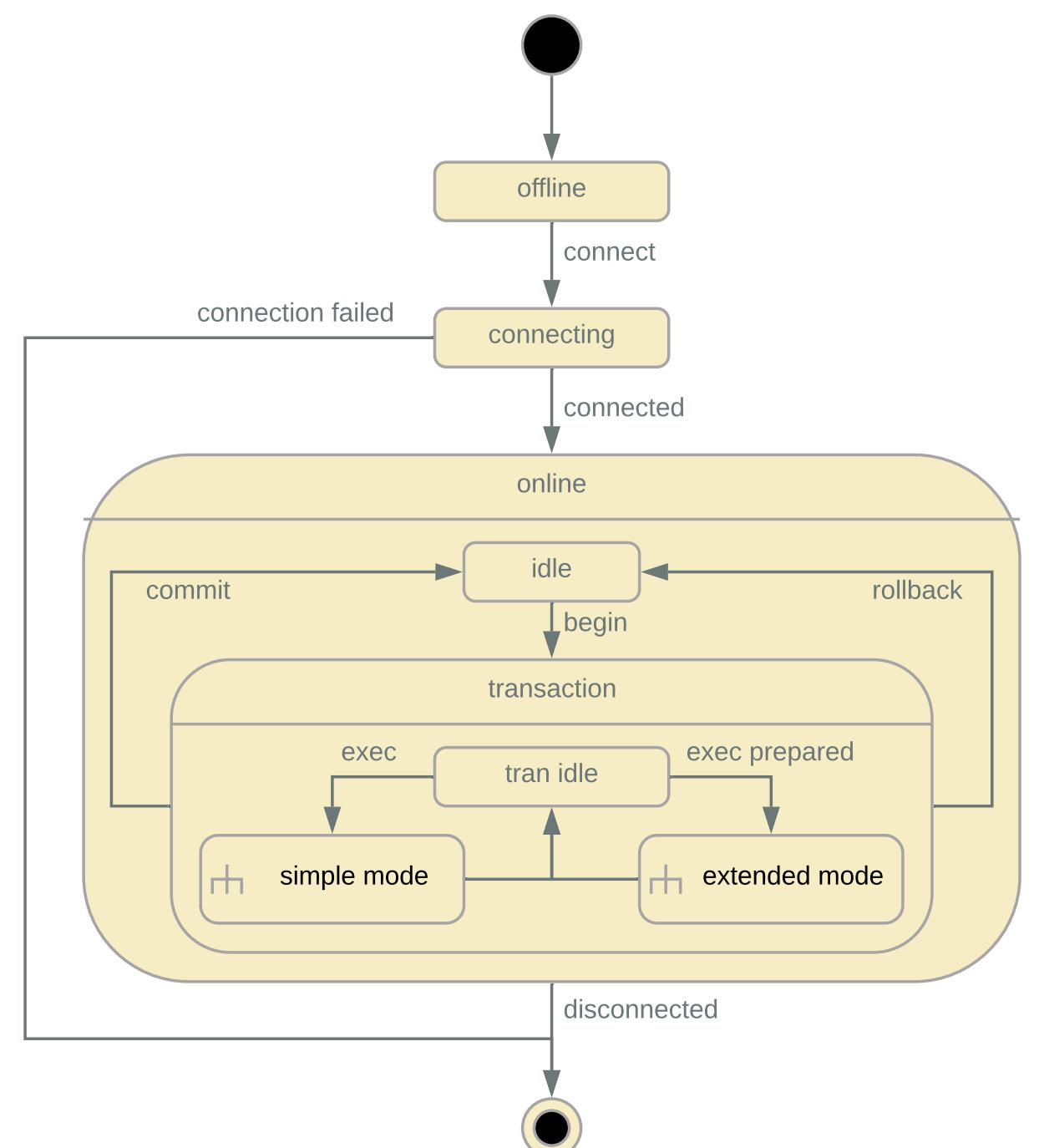
Яндекс Такси

Метапрограммирование: Строим конечный автомат



Яндекс Такси

Метапрограммирование: Строим конечный автомат



О чём поговорим

Конечный автомат - что это?

DSL для конечного автомата

Реализация DSL

Где можно применить?

Конечный автомат - что это?

Определение из Википедии

Конечный автомат – абстрактный автомат, число внутренних состояний которого конечно.



Как выглядит?

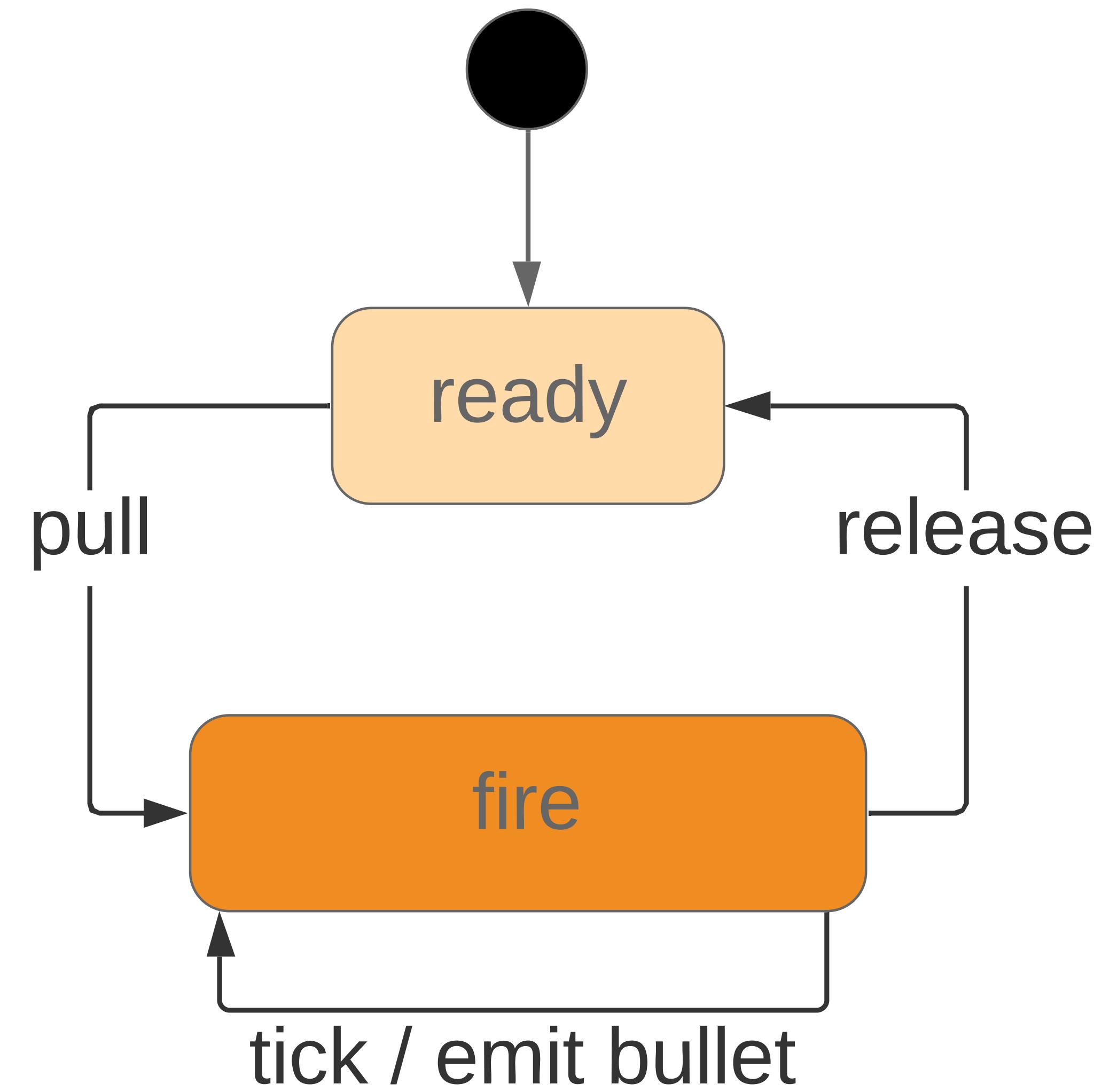
$$M = (V, Q, q_0, F, \delta)$$

```
/\\/* [\\s\\S]*?\\*/|([^\\"\\:]|^)\\/\\.*/$/gm
```

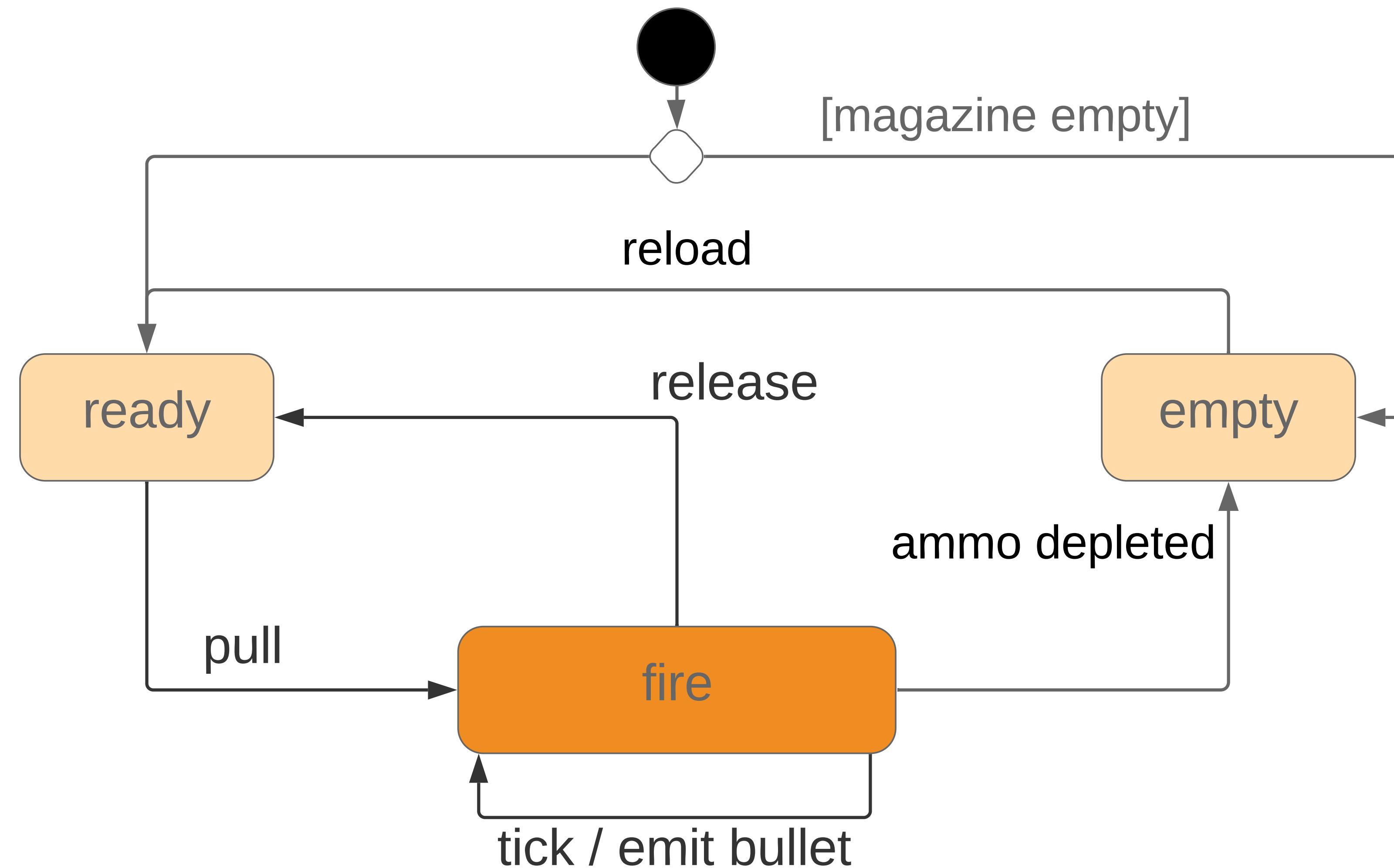
statement

```
: 'if' paren_expr statement
| 'if' paren_expr statement 'else' statement
| 'while' paren_expr statement
| 'do' statement 'while' paren_expr ';'
| '{' statement* '}'
| expr ';'
| ';'
;
```

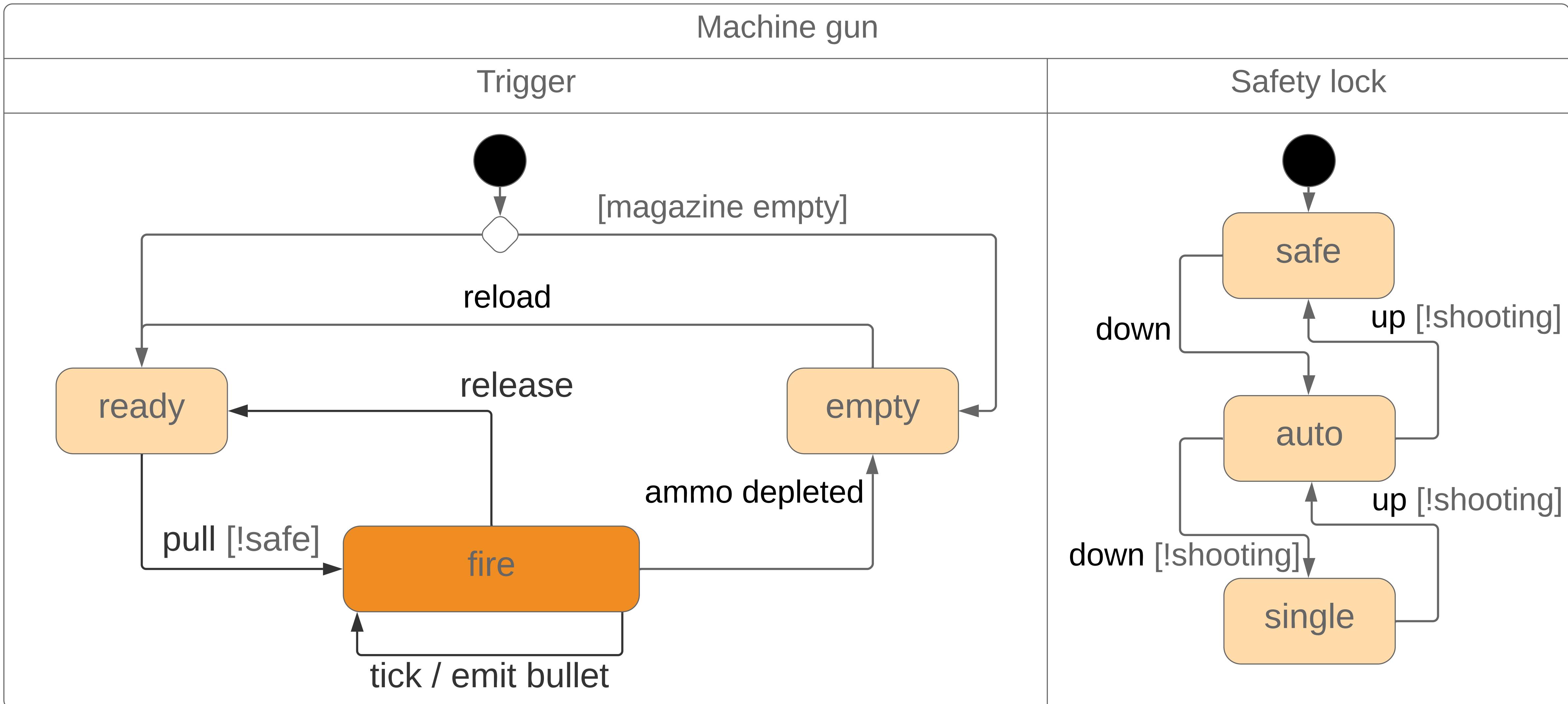
Простейший КА на примере АК



Условные переходы



Ортогональные состояния



Табличное представление

State	Event	Next State	Action	Condition (guard)
trigger				
<i>initial</i>	-	ready		!depleted
<i>initial</i>	-	empty		depleted
empty	reload	ready	change magazine	
ready	reload	fire	change magazine	
ready	pull the trigger	fire	emit bullet	!safe
fire	release the trigger	ready		
fire	tick	fire	emit bullet	automatic && !depleted
fire	-	empty		depleted
selector				
safe	lever down	automatic		
automatic	lever up	safe		!firing
automatic	lever down	single		!firing
single	lever up	automatic		!firing

DSL для конечного автомата

Основные сущности

Состояние (state)

Машина состояний (state machine)

Событие (event)

Переход (transition)

Условие перехода (transition guard)

Действие (action)

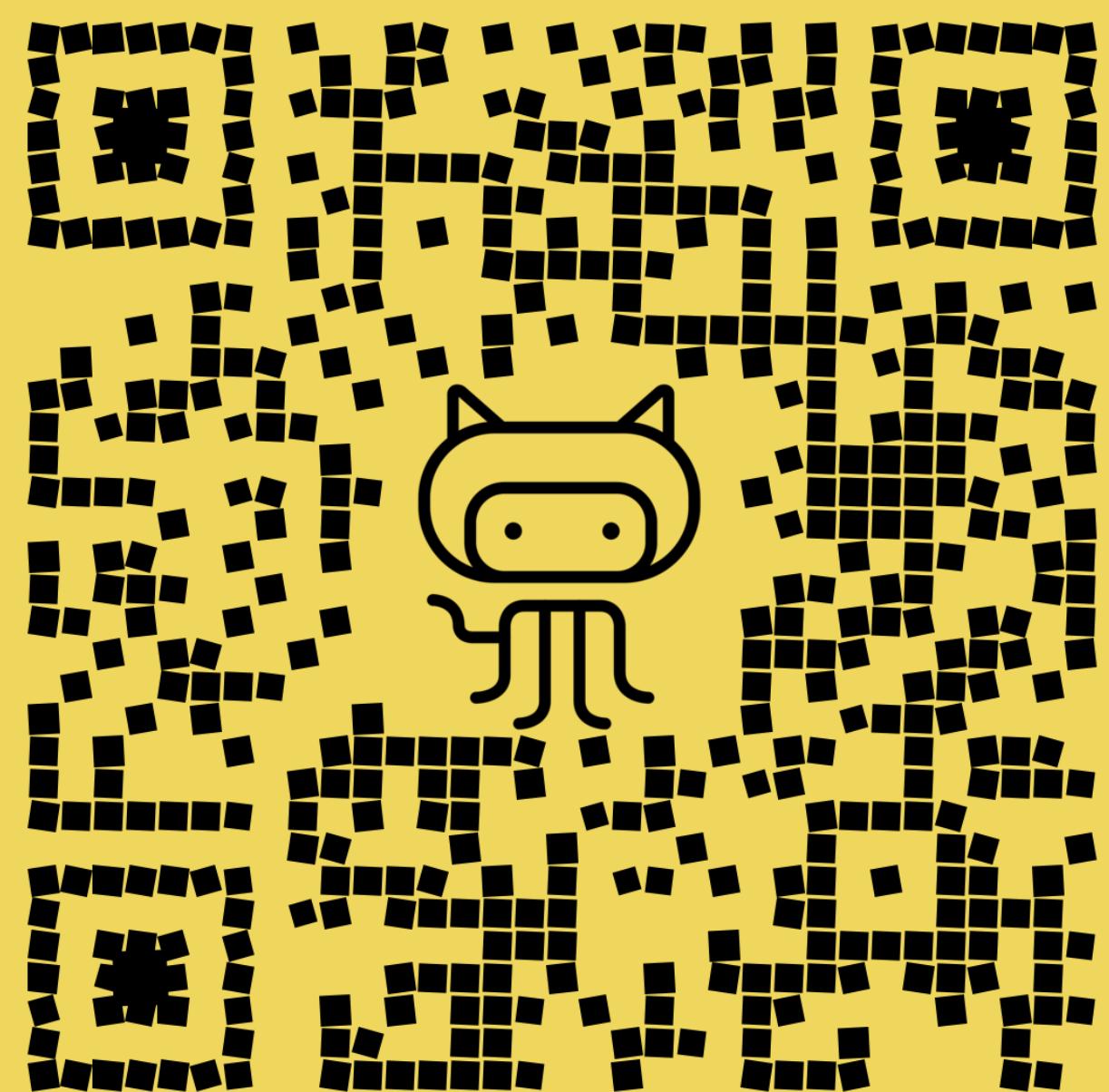
→ при переходе (transition action)

→ при входе в состояние (on entry)

→ при выходе из состояния (on exit)

Таблица переходов (transition table)

Реализация DSL



<https://github.com/zmij/afsm>

Табличное представление

State	Event	Next State	Action	Condition (guard)
trigger				
<i>initial</i>	-	ready		!depleted
<i>initial</i>	-	empty		depleted
empty	reload	ready	change magazine	
ready	reload	fire	change magazine	
ready	pull the trigger	fire	emit bullet	!safe
fire	release the trigger	ready		
fire	tick	fire	emit bullet	automatic && !depleted
fire	-	empty		depleted
selector				
safe	lever down	automatic		
automatic	lever up	safe		!firing
automatic	lever down	single		!firing
single	lever up	automatic		!firing

Таблица переходов на C++

```
// trigger transition table
using transitions = transition_table<
/* State | Event | Next | Action | Guard */
tr< init , none , ready , none , not_depleted>,
tr< init , none , empty , none , depleted>,
tr< empty , reload , ready , change_magazine , none>,
tr< ready , pull , fire , emit_bullet , not_in_state<selector::safe>>,
tr< ready , reload , ready , change_magazine , none>,
tr< fire , release , ready , none , none>,
tr< fire , tick , fire , emit_bullet , and_in_state<selector::automatic>,
                                         not_depleted>,
tr< fire , none , empty , none , depleted>
>;
```



```
// selector transition table
using transitions = transition_table<
/* State | Event | Next | Action | Guard */
tr< safe , down , automatic , update_safety , none >,
tr< automatic , up , safe , update_safety , not_firing>,
tr< automatic , down , single , update_safety , not_firing>,
tr< single , up , automatic , update_safety , not_firing>
>;
```

Где можно применить?

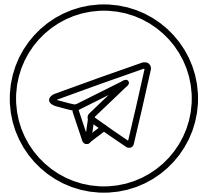
Где применить?

- Сетевые протоколы
- Парсеры
- Игры

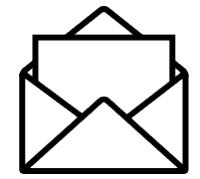
Спасибо

Сергей Федоров

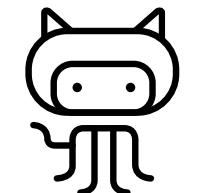
Ведущий разработчик



@zmij_r



ser-fedorov@yandex-team.ru



<https://github.com/zmij>

