These two codes are example contract sources(below example is Solidity, but Vyper has same issue):

contract A { event Transfer(address indexed a, address b, uint256 indexed c);

```
function transfer() public {
    emit Transfer(address(0x776), address(0x777), 256);
}
```

}

contract B { event Transfer(address a, address indexed b, uint256 indexed c);

```
function transfer() public {
    emit Transfer(address(0x777), address(0x776), 256);
}
```

}

Above two contracts are different at the place of the indexed parameter. However, they're compiled bytecodes(with v0.5.11) are same as below(except code hash obviously):

6080604052348015600f57600080fd5b5060a98061001e6000396000f3fe6080604052348015600f57600080fd5b506004361060285760003560e01c80638a4068dd14602d575b600080fd5b60336035565b

So, it means those two different events in each contracts emit the same events and we cannot distinguish the orders of parameters without EXACT

ABI.

The problem is at the ERCs like ERC20 and ERC721.

At ERC20, the official event format for Transfer

is:

event Transfer(address indexed _from, address indexed _to, uint256 _value)

Guess if someone deployed a contract has Transfer

as below:

event Transfer(address indexed _from, address _to, uint256 indexed _value) // or event Transfer(address _from, address indexed _to, uint256 indexed _value)

Is that code ERC20?

- If so, is there any way to distinguish the parameters from event logs without

the code's ABI?

If not, let's take a look withERC721 and CryptoKitties source code.

The official Transfer

event at EIP721 is:

event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);

However, the CryptoKitties implementation is:

event Transfer(address from, address to, uint256 tokenId);

Can we call the Cryptokitties ERC721, then?

So the main question is:

"Is there any ways to distinguish the place of the indexed parameters just with bytecode?"

The question about ERCs is incidental curiosity.

Please let me know if I am wrong.