

The screenshot shows the Truffle UI interface with the following components:

- Left Sidebar:** Icons for Deploy & Run Transactions, Transactions recorded, Deployed Contracts, and a deployed StandardToken contract.
- Middle Panel:** A list of deployed contracts: SAFEMATH AT 0XAE0...96B8B, BASICTOKEN AT 0X9D8...A569, and STANDARDTOKEN AT 0XD4F.. (selected). It also shows the balance as 0 ETH.
- Bottom Panel:** A transaction builder for the STANDARDTOKEN contract. It lists methods: approve, decreaseAppr..., increaseAppr..., transfer, and TRANSFERFROM. The TRANSFERFROM method is highlighted with a red box and has its parameters (_from, _to, _value) filled in. The "Parameters" section is also highlighted with a red box. A red arrow points from the "Parameters" section to the "_from" field in the TRANSFERFROM builder. Another red arrow points from the "Parameters" section to the "transact" button.
- Right Panel:** The code editor window titled "FinToken.sol 1" showing the Solidity code for the StandardToken contract. A red box highlights the "transferFrom" function, which is labeled as the "Target Function".

Red Annotations:

- Enter Transaction Parameters:** A large red box surrounds the "Parameters" section of the transaction builder.
- Execute Transaction:** A red arrow points from the "Parameters" section to the "transact" button in the transaction builder.
- Target Function:** A red box highlights the "transferFrom" function in the code editor.

```
125 * @dev Implementation of the basic standard token.
126 * @dev https://github.com/ethereum/EIPs/issues/20
127 * @dev Based on code by FirstBlood: https://github.com/Firstbloodio/token/blob/master/smart_contract/FirstBloodToken.sol
128 */
contract StandardToken is ERC20, BasicToken {

    mapping (address => mapping (address => uint256)) internal allowed;

    /**
     * @dev Transfer tokens from one address to another
     * @param _from address The address which you want to send tokens from
     * @param _to address The address which you want to transfer to
     * @param _value uint256 the amount of tokens to be transferred
     */

    function transferFrom(address _from, address _to, uint256 _value) public returns (bool) {
        require(_to != address(0));
        require(_value <= balances[_from]);
        require(_value <= allowed[_from][msg.sender]);

        balances[_from] = balances[_from].sub(_value);
        balances[_to] = balances[_to].add(_value);
        allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
        emit Transfer(_from, _to, _value);
        return true;
    }

    /**
     * @dev Approve the passed address to spend the specified amount of tokens on behalf of msg.sender.
     *
     * Beware that changing an allowance with this method brings the risk that someone may use both the old
     * and the new allowance by unfortunate transaction ordering. One possible solution to mitigate this
     */
}
```

[vm] from: 0x5B3...eddC4 to: StandardToken.(constructor) value: 0 wei data: 0x606...70029 logs: 0 hash: 0x6d7...4dc7a