Benchmarking multiple vector theories

This notebook compares the following vector theories (sources in ./boogie-backend/prelude):

- BoogieArray: this is currently the default vector theory used in the Move Prover. It is based on Boogie Arrays (in contrast to native SMT arrays) and does not support extensional equality.
- SmtArray: this is a vector theory using SMT native arrays, without support for extensional equality.
- SmtArrayExt: this is a vector theory using SMT native arrays, with added axioms to ensure extensional equality.

Preparation

Load the prover-lab crate. This may take *long* (minutes) the first time the Jupyter server is started because it compiles a lot Rust sources.

In [14]:
```
:dep prover-lab = { path = "../.." }
```

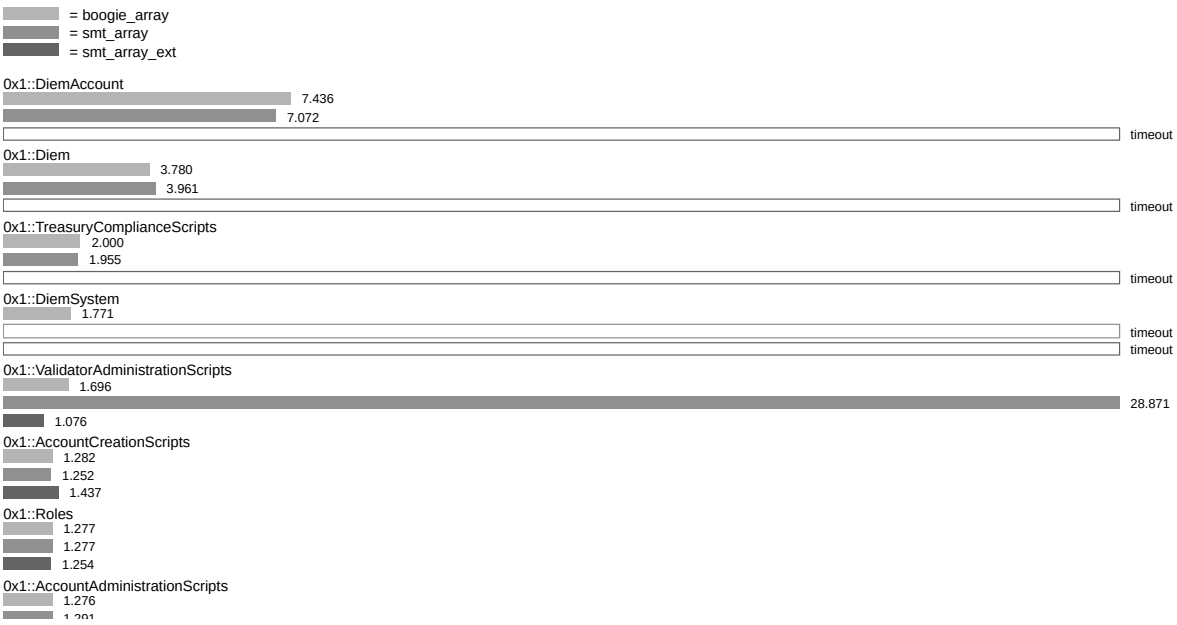
Make functions from the benchmark module available:

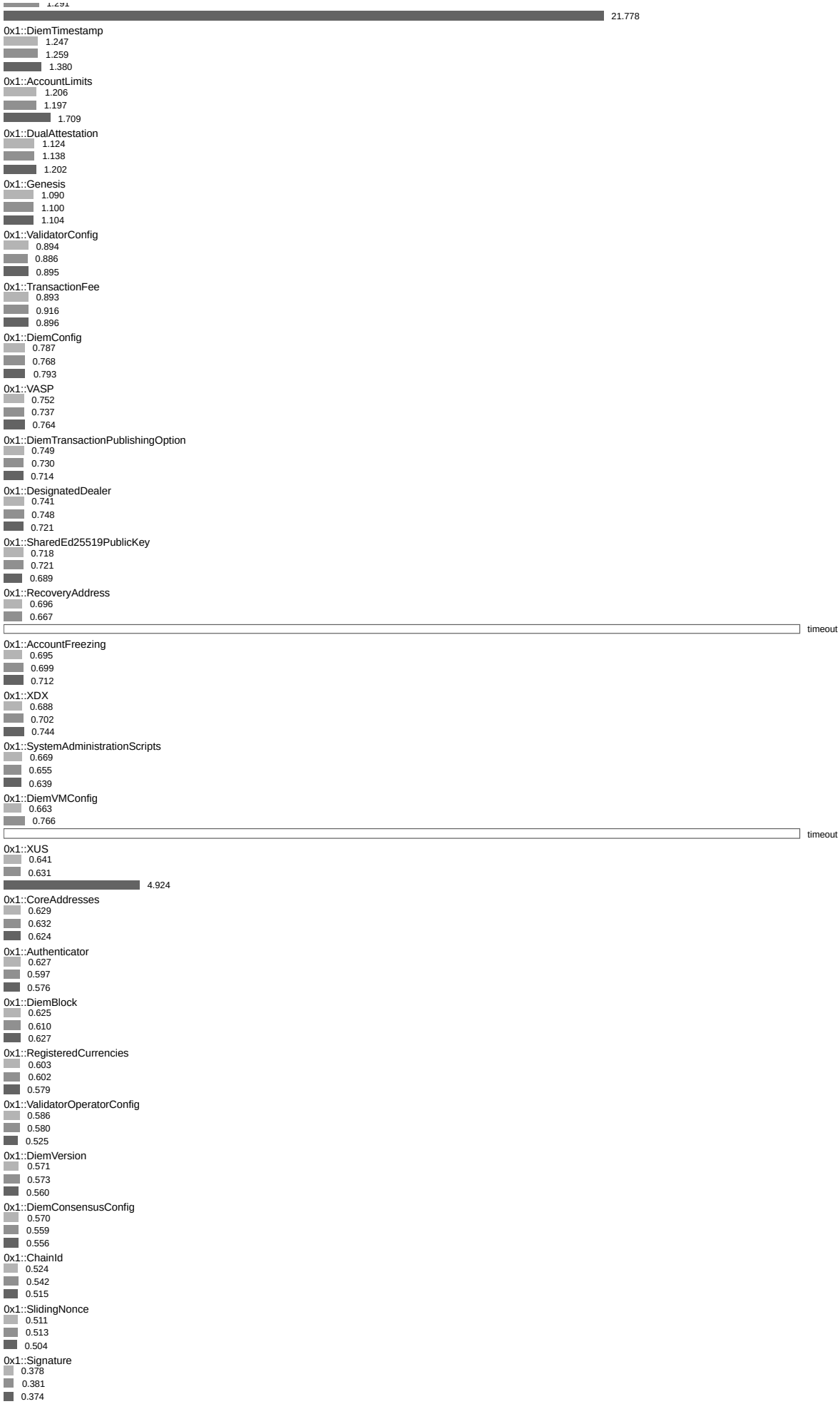
In [15]:
```
use prover_lab::benchmark::*;
```

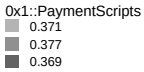
Module Verification Time

In [16]:
```
let mut boogie_array_mod_benchmark = read_benchmark("boogie_array.mod_data")?;
let mut smt_array_mod_benchmark = read_benchmark("smt_array.mod_data")?;
let mut smt_array_ext_mod_benchmark = read_benchmark("smt_array_ext.mod_data")?;
boogie_array_mod_benchmark.sort(); // Will also determine order of other samples
plot_benchmarks(&[&boogie_array_mod_benchmark, &smt_array_mod_benchmark, &smt_ar
```

Out[16]:




1.291
21.778
0x1::DiemTimestamp
1.247
1.259
1.380
0x1::AccountLimits
1.206
1.197
1.709
0x1::DualAttestation
1.124
1.138
1.202
0x1::Genesis
1.090
1.100
1.104
0x1::ValidatorConfig
0.894
0.886
0.895
0x1::TransactionFee
0.893
0.916
0.896
0x1::DiemConfig
0.787
0.768
0.793
0x1::VASP
0.752
0.737
0.764
0x1::DiemTransactionPublishingOption
0.749
0.730
0.714
0x1::DesignatedDealer
0.741
0.748
0.721
0x1::SharedEd25519PublicKey
0.718
0.721
0.689
0x1::RecoveryAddress
0.696
0.667
timeout
0x1::AccountFreezing
0.695
0.699
0.712
0x1::XDX
0.688
0.702
0.744
0x1::SystemAdministrationScripts
0.669
0.655
0.639
0x1::DiemVMConfig
0.663
0.766
timeout
0x1::XUS
0.641
0.631
4.924
0x1::CoreAddresses
0.629
0.632
0.624
0x1::Authenticator
0.627
0.597
0.576
0x1::DiemBlock
0.625
0.610
0.627
0x1::RegisteredCurrencies
0.603
0.602
0.579
0x1::ValidatorOperatorConfig
0.586
0.580
0.525
0x1::DiemVersion
0.571
0.573
0.560
0x1::DiemConsensusConfig
0.570
0.559
0.556
0x1::ChainId
0.524
0.542
0.515
0x1::SlidingNonce
0.511
0.513
0.504
0x1::Signature
0.378
0.381
0.374

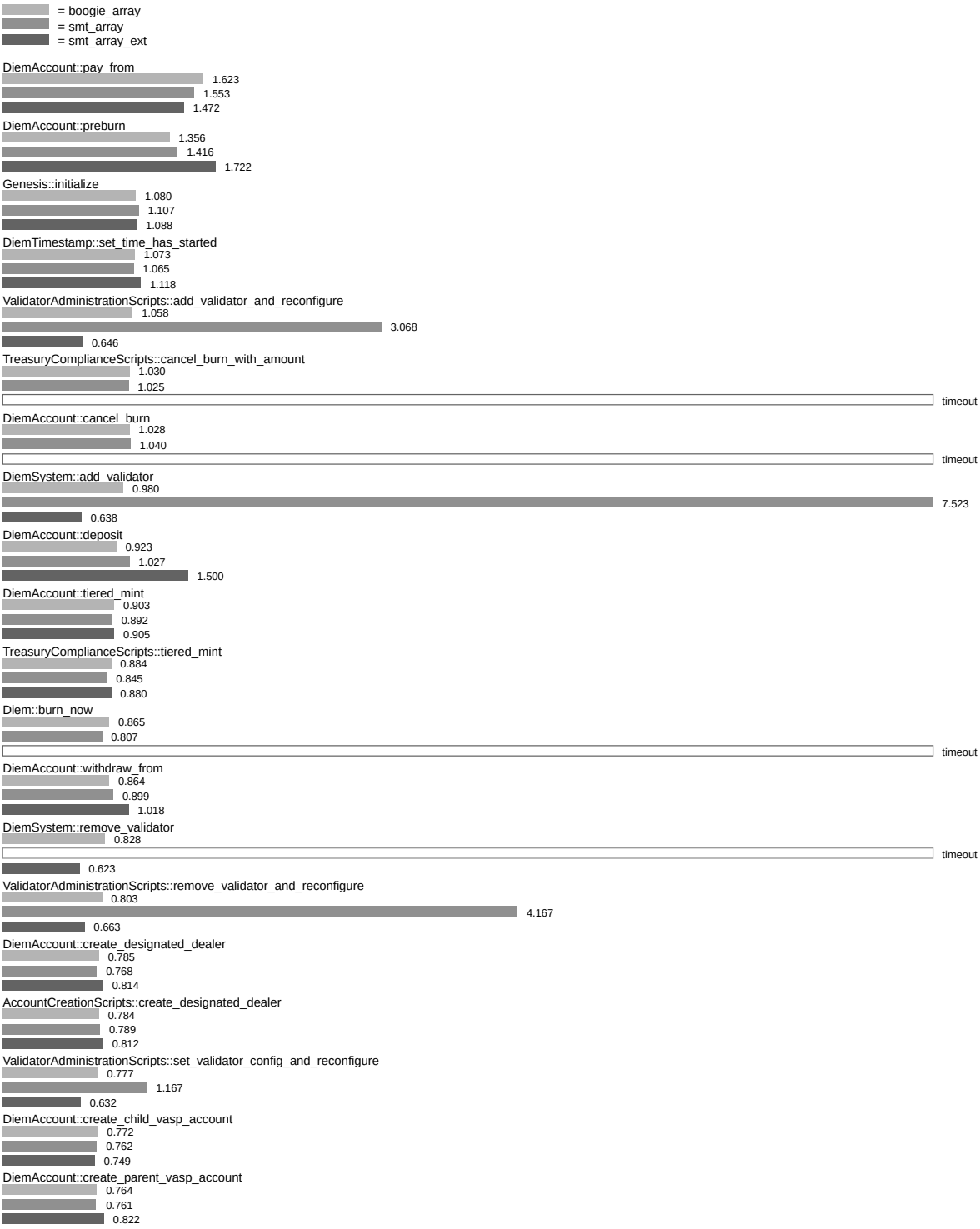


Top 20 by Function

In [17]:

```
let mut boogie_array_fun_benchmark = read_benchmark("boogie_array.fun_data")?;
let mut smt_array_fun_benchmark = read_benchmark("smt_array.fun_data")?;
let mut smt_array_ext_fun_benchmark = read_benchmark("smt_array_ext.fun_data")?;
boogie_array_fun_benchmark.sort();
boogie_array_fun_benchmark.take(20);
plot_benchmarks(&[&boogie_array_fun_benchmark, &smt_array_fun_benchmark, &smt_ar
```

Out[17]:


= boogie_array
= smt_array
= smt_array_ext
DiemAccount::pay_from
1.623
1.553
1.472
DiemAccount::preburn
1.356
1.416
1.722
Genesis::initialize
1.080
1.107
1.088
DiemTimestamp::set_time_has_started
1.073
1.065
1.118
ValidatorAdministrationScripts::add_validator_and_reconfigure
1.058
3.068
0.646
TreasuryComplianceScripts::cancel_burn_with_amount
1.030
1.025
timeout
DiemAccount::cancel_burn
1.028
1.040
timeout
DiemSystem::add_validator
0.980
7.523
0.638
DiemAccount::deposit
0.923
1.027
1.500
DiemAccount::tiered_mint
0.903
0.892
0.905
TreasuryComplianceScripts::tiered_mint
0.884
0.845
0.880
Diem::burn_now
0.865
0.807
timeout
DiemAccount::withdraw_from
0.864
0.899
1.018
DiemSystem::remove_validator
0.828
timeout
0.623
ValidatorAdministrationScripts::remove_validator_and_reconfigure
0.803
4.167
0.663
DiemAccount::create_designated_dealer
0.785
0.768
0.814
AccountCreationScripts::create_designated_dealer
0.784
0.789
0.812
ValidatorAdministrationScripts::set_validator_config_and_reconfigure
0.777
1.167
0.632
DiemAccount::create_child_vasp_account
0.772
0.762
0.749
DiemAccount::create_parent_vasp_account
0.764
0.761
0.822

In [ ]: