

System Specification, Verification and Synthesis (SSVS) – CS 4830/7485, Fall 2019

7: Formal System Modeling: State-Space Explosion

Stavros Tripakis



Northeastern University
**Khoury College of
Computer Sciences**

STATE-SPACE EXPLOSION

State-space explosion

- If I have an asynchronous system with 10 processes and each process has 10 states, how many states in total?

State-space explosion

- If I have an asynchronous system with 10 processes and each process has 10 states, how many states in total?

Answer: 10^{10} = ten billion.

- What if each process has 20 states instead?

State-space explosion

- If I have an asynchronous system with 10 processes and each process has 10 states, how many states in total?
Answer: 10^{10} = ten billion.
- What if each process has 20 states instead?
Answer: 20^{10} > ten trillion.
- What if I had 20 processes instead? (with 10 states each)

State-space explosion

- If I have an asynchronous system with 10 processes and each process has 10 states, how many states in total?

Answer: 10^{10} = ten billion.

- What if each process has 20 states instead?

Answer: 20^{10} > ten trillion.

- What if I had 20 processes instead? (with 10 states each)

Answer: 10^{20} = 100 quintillion $\gg 20^{10}$!

⇒ **The number of processes is more important than the number of states per process!**

- **Exponential increase in the number of processes!**

State-space explosion

- If I have an asynchronous system with 10 processes and each process has 10 states, how many states in total?

Answer: 10^{10} = ten billion.

- What if each process has 20 states instead?

Answer: 20^{10} > ten trillion.

- What if I had 20 processes instead? (with 10 states each)

Answer: 10^{20} = 100 quintillion $\gg 20^{10}$!

⇒ **The number of processes is more important than the number of states per process!**

- **Exponential increase in the number of processes!**
- How many processes are running in your laptop right now?

State-space explosion

- What if I have a single process (just a single sequential program)?
- How many variables does your program have?
- Let's say 3 variables.
- What type are those variables?
- Let's say they are all `int`.
- 32 bit or 64 bit?
- Let's say 32 bit.
- How many states are there then?

State-space explosion

- What if I have a single process (just a single sequential program)?
- How many variables does your program have?
- Let's say 3 variables.
- What type are those variables?
- Let's say they are all int.
- 32 bit or 64 bit?
- Let's say 32 bit.
- How many states are there then?
Answer: $(2^{32})^3 = 2^{32 \cdot 3} = 2^{96} > 10^{28}$
- **Exponential increase in the number of variables and number of bits per variable!**

State-space explosion

- What if I have a single process (just a single sequential program)?
- How many variables does your program have?
- Let's say 3 variables.
- What type are those variables?
- Let's say they are all int.
- 32 bit or 64 bit?
- Let's say 32 bit.
- How many states are there then?
Answer: $(2^{32})^3 = 2^{32 \cdot 3} = 2^{96} > 10^{28}$
- **Exponential increase in the number of variables and number of bits per variable!**
- Imagine what happens with real programs: arrays, lists, other complex data structures, ... E.g., a single array of 10 ints has $(2^{32})^{10}$ states.

State-space explosion

- What if I have a piece of hardware (digital circuit)?
- How many flip-flops (memory elements) does your circuit have?
- Let's say 10 flip-flops.
- Then $2^{10} = 1024$ states: easy for modern model-checkers.
- How many flip-flops does modern hardware have?
- Intel Core i7: just the L1 cache¹ is 64 KiB per core $\Rightarrow 2^{64000 \cdot 8}$ states !

¹There's also L2 and L3 caches and these are much larger.

State-space explosion

- The number 1 problem in formal verification.
- Still an unsolved problem.
- But: situation much better than 40 years ago!
- Thanks to many clever ideas.
- We will examine the most important of them in this course.

Bibliography



Baier, C. and Katoen, J.-P. (2008).

Principles of Model Checking.

MIT Press.



Clarke, E., Grumberg, O., and Peled, D. (2000).

Model Checking.

MIT Press.



Clarke, E. M., Henzinger, T. A., Veith, H., and Bloem, R. (2018).

Handbook of Model Checking.

Springer.



Holzmann, G. (1991).

Design and Validation of Computer Protocols.

Prentice Hall.



Holzmann, G. (2003).

The Spin Model Checker.

Addison-Wesley.