**Static Stack Depth Property**

Claim: if we apply translation above then the stack size at each point in the bytecode can be computed during compilation.

* we can check for stack underflow and overflow

Why does it hold?

* for each assignment statement
* for if, while

Can JVM rely on this property?

* what if bytecode is not compiled from java / tool ?
* bytecode verifier checks the stack usage (including stack depth and initialization)

Consider depth computation for [Compiled Factorial Example](http://lara.epfl.ch/w/cc09:compiled_factorial_example)

How does this property relate to [Translation Correctness for Expressions](http://lara.epfl.ch/w/cc09:translation_correctness_for_expressions)?

* the property implies that compilation of expression only adds one value to the stack in the end
* assignment will consume this value, so each assignment leaves stack unchanged!