Amdahl's Law

Let the function $T(n)$ represent the time a program takes to execute with n processors. The speedup computed by Amdahl's law is a comparison between $T(1)$, the time on a uniprocessor, and $T(n)$, the time on a multiprocessor with n processors.

$$S(n) = \frac{T(1)}{T(n)}$$

If we know what fraction of $T(1)$ is spent computing parallelizable code, we can determine $T(n)$ in terms of $T(1)$. If p is the parallel fraction of $T(1)$, then it takes $p \cdot T(1)$ units of time to run the parallel part and $(1-p) \cdot T(1)$ units of time to run the sequential part. Dividing the parallel time across all n processors, we form $T(n) = T(1)(1-p) + \frac{T(1)p}{n}$. Substituting into the equation for the speedup, we get

$$S(n) = \frac{T(1)}{T(n)} = \frac{T(1)}{T(1)(1-p) + \frac{T(1)p}{n}}$$

Normally, Amdahl's Law is presented with all the $T(1)$ terms cancelled out, but leaving them in helps with the intuition. In particular, it captures the idea that p and $1-p$ are both fractions of $T(1)$, the time the program takes to execute on 1 processor.