We observe that “symbolic computation” and “computer algebra” are really
two different things and that neither one sufficiently addresses the prob-
lems that arise in applications. Symbolic computation may be seen as work-
ing with expression trees representing mathematical formulae and applying
various rules to transform them. Computer algebra may be seen as devel-
oping constructive algorithms to compute quantities in various arithmetic
domains, possibly involving indeterminates. Symbolic computation allows
a wider range of expression, but lacks efficient algorithms. It is often un-
clear what is the algebraic structure of a domain defined by rewrite rules.
Computer algebra admits greater algorithmic precision, but is limited in the
problems that it can model. We argue that considerable work is still required
to make symbolic computation more effective and computer algebra more ex-
pressive. We use polynomials with symbolic exponents, e.g. $x^{n^2+n} - y^{2m}$, as
an example that lies in the middle ground and we present algorithms for their
factorization and gcd.