The Power of Syntax: Type Soundness

1 A Syntactic Approach to Type Soundness

```
@article{wf:type-soundness,
    author = {Wright, A. K. and Felleisen, M.},
    title = {A Syntactic Approach to Type Soundness},
    journal = {Information and Computation},
    volume = {115},
    pages = {38--94},
    year = {1992},
}
```

Summary: Wright and Felleisen propose a new proof technique for establishing type soundness. First, though, they analyze previous ways of proving the property. They conclude that the previous approaches are brittle as they are tuned for particular linguistic features in a manner that is neither extensible nor composable. In particular, the previous techniques depend on a denotational semantics for the language under investigation and(or) a denotational semantics for the types of the language. Wright and Felleisen avoid this complication by (a) viewing computation as a sequence of syntax re-writing steps on program terms, and (b) establishing that computation preserves type-checking. The latter, a translation of subject reduction from combinatorial logic to operation semantics, together with a proof that program terms that raise type errors do not type check is sufficient to establish type soundness. Wright and Felleisen show that the structure of their syntactic proof of type soundness, what nowadays we call progress and preservation, remains unchanged as they apply it first to a pure subset of ML and then extensions of this subset with first-class references and first-class control.

Evaluation: This is a seminal paper. It introduces a highly flexible technique that can be adjusted to diverse linguistic settings. In hindsight, this paper has provided the theoretical tool that enabled and validated the last twenty years of research on the design of type systems.