Alternative Abstract Syntax and Semantic Equations for Calculator

Program ::= ExprSeq
ExprSeq ::= Expr | Expr ExprSeq
Expr ::= Term | Expr Op Term | Expr Ans | Expr Ans +/-
Term ::= Num | $\mathbf{M^R}$ | **Clr** | Term +/-
Op ::= **+** | **−** | ***
Ans ::= $\mathbf{M^+}$ | **=**
Num ::= Fig. 9.1

Alternative (unambiguous) Fig. 9.5

meaning: Program → Integer
perform: ExprSeq → (State → State)
evaluate: Expr → (State → State)
compute: Op → (State → State)
calculate: Ans → {+/-} → (State → State)
value: Num → Integer

Alternative semantic function types

meaning ⟦P⟧ = d, where perform ⟦P⟧(0, nop, 0, 0) = (a, op, d, m)
perform ⟦E⟧ = evaluate ⟦E⟧
perform ⟦E S⟧ = perform ⟦S⟧ ∘ evaluate ⟦E⟧
evaluate ⟦N⟧ (a, op, d, m) = (a, op, value ⟦N⟧, m)
evaluate ⟦$\mathbf{M^R}$⟧ (a, op, d, m) = (a, op, m, m)
evaluate ⟦**Clr**⟧ (a, op, d, m) = (0, nop, 0, 0)
evaluate ⟦T +/-⟧ = calculate ⟦+/-⟧ ∘ evaluate ⟦T⟧
evaluate ⟦E O T⟧ = evaluate ⟦T⟧ ∘ compute ⟦O⟧ ∘ evaluate ⟦E⟧
evaluate ⟦E A⟧ = calculate ⟦A⟧ ∘ evaluate ⟦E⟧
evaluate ⟦E A +/-⟧ = calculate ⟦+/-⟧ ∘ calculate ⟦A⟧ ∘ evaluate ⟦E⟧
compute ⟦**+**⟧ (a, op, d, m) = (op(a, d), plus, op(a, d), m)
compute ⟦**−**⟧ (a, op, d, m) = (op(a, d), minus, op(a, d), m)
compute ⟦***⟧ (a, op, d, m) = (op(a, d), times, op(a, d), m)
calculate ⟦**=**⟧ (a, op, d, m) = (a, nop, op(a, d), m)
calculate ⟦$\mathbf{M^+}$⟧ (a, op, d, m) = (a, nop, op(a, d), plus(m, op(a, d)))
calculate ⟦+/-⟧ (a, op, d, m) = (a, op, minus(0, d), m)
value ⟦N⟧ = Fig. 9.1

Alternative Fig. 9.8