**Algorithm 1.** Two-phase simplification of invariants

**Input**: $R = \{\psi_1, \ldots, \psi_n\}$: invariant set to be reduced,
$P$: input property set, $T$: Transition relation

```
Function trim(R, P)
  if R(0..k−1) ∧ P(0..k−1) ∧ T(0..k) ⊨_L P(k)
  then
    // P is k-inductive wrt R
    U = get-unsat-core();
    R′ = {ψ ∈ R | ψ occurs in U};
    if R′(0..k − 1) ∧ P(0..k − 1) ∧ T(0..k) ⊨_L
    R′(k) ∧ P(k) then
      // R′ ∧ P is k-inductive
      return R′ ∪ P
    else // R′ is not strong enough
      trim(R \ R′, R′ ∪ P)
  else error "Not k-inductive";
```

```
Function cherry-pick(R, P)
  if P(0..k − 1) ∧ T(0..k) ⊨_L P(k) then
    // P is k-inductive
    return P
  else
    // Find cex to induction
    M = get-cex();
    // ...and a blocking invariant
    ψ = choose({ψ ∈ R | M ⊭ ψ});
    cherry-pick(R\{ψ}, P ∪ {ψ})
```

```
cherry-pick( trim({ψ_1, ..., ψ_n}, P), P );
```